

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Саратовский государственный технический университет  
имени Гагарина Ю.А.»

Филиал федерального государственного бюджетного образовательного  
учреждения высшего образования  
«Саратовский государственный технический университет  
имени Гагарина Ю.А.» в г. Петровске

СОГЛАСОВАНО

Генеральный директор АО  
«Петровский электромеханический  
завод «Молот»

 А.Е.Резник  
«06» 20 24 г.



УТВЕРЖДАЮ

Директор филиала СГТУ  
имени Гагарина Ю.А. в г.Петровске

 Е.А.Беснашникова  
«06» 20 24 г.



**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ОЦЕНКЕ КАЧЕСТВА  
ОСВОЕНИЯ СТУДЕНТАМИ  
ПРОГРАММ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА**


**по профессиональному модулю**

ПМ.01 «Разработка модулей программного обеспечения  
для компьютерных систем»

по специальности

09.02.07 «Информационные системы и программирование»

Фонд оценочных средств рассмотрен  
на заседании предметной (цикловой) комиссии  
обще профессиональных дисциплин,  
профессиональных модулей специальностей  
технического профиля  
«14» июня 2024 года, протокол №12

Председатель ПЦК  /Ю.А.Табарова/

Петровск 2024

# **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ОЦЕНКЕ КАЧЕСТВА ОСВОЕНИЯ СТУДЕНТАМИ ПРОГРАММ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА**

по междисциплинарному курсу  
МДК.01.01. «Разработка программных модулей»

## **Пояснительная записка**

Фонд оценочных средств разработан в соответствии с рабочей программой профессионального модуля, требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 «Информационные системы и программирование», утверждённого приказом Министерства образования и науки РФ от 09.12.2016 г., № 1547 (ред. от 01.09.2022), ФГОС среднего общего образования и примерной основной образовательной программой.

### **1. Паспорт фонда оценочных средств для проведения текущего контроля успеваемости**

#### **1.1. Цели и задачи контроля**

Целью текущего контроля успеваемости обучающихся является обеспечение систематического контроля и оценки уровня освоения предметных результатов, уровня сформированности общих компетенций МДК. 01.01 «Разработка программных модулей».

Главной задачей текущего контроля успеваемости является повышение мотивации обучающихся к регулярной учебной и самостоятельной работе, закрепление, углубление знаний, закрепление и совершенствование умений, обеспечение соответствия результатов обучения задачам будущей профессиональной деятельности посредством внедрения эффективной системы оценки в образовательный процесс.

#### **Общие компетенции, включающие в себя способность:**

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях.

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

ОК 10. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере

### **Профессиональные компетенции, включающие в себя способность:**

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

### **Знания и умения:**

#### **знать:**

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- способы оптимизации и приемы рефакторинга;
- основные принципы отладки и тестирования программных продуктов

#### **уметь:**

- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;
- уметь выполнять оптимизацию и рефакторинг программного кода;
- оформлять документацию на программные средства

## **1.2. Структура фонда оценочных средств**

Фонд оценочных средств для проведения текущего контроля включает в себя комплекты контрольно-оценочных средств, предназначенные для проведения текущего контроля в виде:

- оперативного контроля;
- рубежного контроля.

Оперативный контроль проводится в форме:

- опрос (устный, письменный);
- тестирование;
- выполнение практической работы;

Рубежный контроль проводится в форме:

- опрос (устный, письменный);
- тестирование;
- выполнение практической работы;

Фонд оценочных средств также, включает в себя комплект контрольно-оценочных средств для проведения межсессионной аттестации. Межсессионная аттестация проводится в форме тестирования.

### **1.3. Система оценивания результатов выполнения заданий**

Результаты выполнения заданий оцениваются в соответствии с разработанными критериями оценки.

Используются следующие шкалы для оценивания результатов обучения:

- пяти бальная шкала оценки;
- сто бальная шкала оценки.

Перевод сто бальной шкалы учета результатов в пяти бальную оценочную шкалу:

<b>Оценка</b>	<b>Количество баллов, набранных за выполнение теоретического и практического задания</b>
Оценка 5 «отлично»	90-100
Оценка 4 «хорошо»	76-89
Оценка 3 «удовлетворительно»	50-75
Оценка 2 «неудовлетворительно»	≤ 49

Перевод пяти бальной шкалы учета результатов в пяти бальную оценочную шкалу:

<b>Оценка</b>	<b>Количество баллов, набранных за выполнение теоретического и практического задания, средний балл по итогам аттестации</b>
Оценка 5 «отлично»	4,6-5
Оценка 4 «хорошо»	3,6-4,5
Оценка 3 «удовлетворительно»	3-3,5
Оценка 2 «неудовлетворительно»	≤ 2,9

### **1.4. Учебно-методическое и информационное обеспечение для проведения контроля**

**Основные учебные издания:**

1.Зыков, С. В. Введение в теорию программирования. Объектно-ориентированный подход: учебное пособие для СПО / С. В. Зыков. — Саратов: Профобразование, 2021. — 187 с. — ISBN 978-5-4488-0995-8. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/102188>

2.Кариев, Ч. А. Разработка Windows-приложений на основе Visual C#: учебное пособие / Ч. А. Кариев. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 978 с. — ISBN 978-5-4497-0909-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/102057.html>

3. Лебедева, Т. Н. Технология программирования: учебное пособие для СПО/ Т. Н. Лебедева, С. С. Юнусова. — Саратов: Профобразование, 2019. — 140 с. — ISBN 978-5-4488-0351-2. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/86081>

#### **Дополнительные учебные издания:**

4. Биллиг, В. А. Основы программирования на C#: учебное пособие / В. А. Биллиг. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 573 с. — ISBN 978-5-4497-0893-9. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/102033>

5.Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие для СПО / Т. М. Зубкова. — Саратов: Профобразование, 2019. — 468 с. — ISBN 978-5-4488-0354-3. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/86208>

#### **Электронные издания (электронные ресурсы)**

6.Учебники по программированию <http://programm.ws/index.php>

#### **Электронно-библиотечная система:**

7.ЭБС «IPRbooks», ООО «Ай Пи Ар Медиа»

8. ЭБС «Znanium»

9.ЭБС «PROФобразование»

10.ЭБС «Book.ru»

Таблица 1

**Распределение знаний и умений в соответствии с профессиональными компетенциями**

<b>Профессиональные компетенции</b>	<b>Знания</b>	<b>Умения</b>
ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Актуальная нормативно-правовая база в области документирования алгоритмов.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. Оформлять документацию на программные средства. Оценка сложности алгоритма.
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.	Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Знание API современных мобильных операционных систем	Создавать программу по разработанному алгоритму как отдельный модуль. Оформлять документацию на программные средства. Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.

Таблица 2

**Распределение знаний и умений в соответствии с общими компетенциями**

<b>Общие компетенции</b>	<b>Знания</b>	<b>Умения</b>
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности	распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы; владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или

		с помощью наставника)
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.	номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации	определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях	содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования	определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования
ОК 04. Эффективно взаимодействовать и работать в коллективе и команде.	психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности	организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	особенности социального и культурного контекста; правила оформления документов и построения устных сообщений	грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности	описывать значимость своей специальности
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства,	правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной	соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по

эффективно действовать в чрезвычайных ситуациях.	деятельности; пути обеспечения ресурсосбережения	специальности
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения	использовать физкультурно-оздоровительную деятельность для укрепления здоровья, достижения жизненных и профессиональных целей; применять рациональные приемы двигательных функций в профессиональной деятельности; пользоваться средствами профилактики перенапряжения характерными для данной специальности
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках	правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности	понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы
ОК 10. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере	основы предпринимательской деятельности; основы финансовой грамотности; правила разработки бизнес-планов; порядок выстраивания презентации; кредитные банковские продукты	выявлять достоинства и недостатки коммерческой идеи; презентовать идеи открытия собственного дела в профессиональной деятельности; оформлять бизнес-план; рассчитывать размеры выплат по процентным ставкам кредитования; определять инвестиционную привлекательность коммерческих идей в рамках профессиональной деятельности; презентовать бизнес-идею; определять источники финансирования



## **2. Контрольно-оценочные средства**

### Теоретическое занятие 1. Жизненный цикл ПО

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

#### **Вопросы:**

1. Что такое жизненный цикл ПО?
2. С какого момента появляется жизненный цикл?
3. Назовите модели жизненного цикла.
4. В чем суть каскадной модели жизненного цикла?
5. Что такое спиральная модель жизненного цикла?

### Теоретическое занятие 2. Технология структурного программирования

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

#### **Вопросы:**

1. Что называется структурным программированием?
2. Какие структуры программирования вам известны?
3. Назовите принципы структурного программирования.

### Теоретическое занятие 3. Инструментальные средства оформления и документирования алгоритмов программ

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

#### **Вопросы:**

1. На основании какого стандарта производится оформление алгоритмов программ?
2. Какие инструментальные средства используются для оформления алгоритмов программ?
3. Какие инструментальные средства используются для документирования алгоритмов программ?
4. Поясните правила оформления и документирования алгоритмов программ.

Теоретическое занятие 4. Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое сложность алгоритма?
2. Какими параметрами измеряется сложность алгоритма?
3. Какие бывают разновидности сложности алгоритмов?
4. Что такое асимптотическая сложность алгоритма?
5. Как определить временную сложность алгоритма при известной рабочей функции?

Теоретическое занятие 5. Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Как можно классифицировать алгоритмы в соответствии с их временной сложности?
2. Как по описанию алгоритма определить его временную сложность?
3. Назовите классы алгоритмов.
4. Что такое неразрешимые задачи? Приведите примеры.
5. Опишите проблему определения ключа путем перебора всех возможных вариантов.

Практическая работа 1. Изучение методов оценки алгоритмов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Название и цель работы.
2. Формулы асимптотической и верхней оценки сложности заданного алгоритма.
3. Исходный код программы экспериментальной оценки временной сложности заданного алгоритма для массива большой размерности.
4. Значения временной сложности алгоритма, полученные экспериментальным способом, а также количество повторений алгоритма и объем исходных данных, при котором были получены эти значения.

5. График зависимости времени выполнения алгоритма от объема исходных данных.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 2. Оценка сложности алгоритмов сортировки

**Форма контроля:** Рубежный контроль

**Задание:**

1. Название и цель работы.
2. Словесное описание заданных алгоритмов сортировки.
3. Тексты программ.
4. Формулы верхней оценки временной и емкостной сложности заданных алгоритмов.
5. Результаты экспериментальной оценки временной и емкостной сложности заданных алгоритмов

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 3. Оценка сложности алгоритмов поиска

**Форма контроля:** Рубежный контроль

**Задание:**

1. Название и цель работы.
2. Словесное описание заданного алгоритма поиска.
3. Текст программы.
4. Формулы верхней оценки временной и емкостной сложности заданного алгоритма.
5. Результаты экспериментальной оценки временной и емкостной сложности заданного алгоритма

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 4. Оценка сложности рекурсивных алгоритмов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Название и цель работы.
2. Словесное описание заданных рекурсивных алгоритмов.
3. Тексты программ.
4. Формулы верхней оценки временной и емкостной сложности заданных алгоритмов.
5. Результаты экспериментальной оценки временной и емкостной сложности заданных алгоритмов

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 5. Оценка сложности эвристических алгоритмов

**Форма контроля:** Рубежный контроль

**Задание:**

Составить алгоритм и определить его сложность для следующих задач:

1. Задан массив  $A(6,5)$ . Найти номер строки и номер столбца с минимальным значением.
2. Найти количество отрицательных элементов в массиве  $C1, C2, \dots, C20$ , используя оператор цикла *do until loop*.
3. Найти максимальное значение из значений элементов последовательности  $x1, x2, \dots, x20$  (одномерного массива), используя оператор *do while...loop*
4. Определите произведение трех переменных.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Самостоятельная работа 1. Инструментальные средства разработки программных модулей

**Форма контроля:** Рубежный контроль

**Задание:**

1. Подготовить презентацию по одной из следующих тем:

1. Языки программирования
2. Среды программирования
3. Инструментальные средства разработки программных модулей

Порядок выполнения работы согласно методическим рекомендациям по самостоятельным работам по МДК. 01.01 «Разработка программных модулей».

Теоретическое занятие 6. Основные принципы объектно-ориентированного программирования. Классы: основные понятия

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое объектно-ориентированное программирование?
2. Назовите принципы ООП.
3. Что такое инкапсуляция?
4. Что такое наследование?
5. Что такое полиморфизм?

Практическая работа 6. Работа с классами

**Форма контроля:** Рубежный контроль

**Задание:**

1. Написать программу для определения класса, описывающего некоторого студента вуза

2. Описать класс «Домашняя библиотека».

3. Описать класс «Записная книжка».

4. Описать класс «Студенческая группа».

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

#### Практическая работа 7. Работа с классами

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать программу, которая будет заниматься учетом успеваемости студентов в группе. Создайте заголовочный файл `students.h`, в котором будет находиться класс `Students`.

2. Разработать класс, инкапсулирующий двумерный массив. Класс должен содержать поля и методы, необходимые для реализации приведенного задания

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

#### Теоретическое занятие 7. Перегрузка методов

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое перегрузка методов?
2. Что такое переопределение метода?
3. Что является уникальной характеристикой перегруженных методов

#### Лабораторная работа 1. Перегрузка методов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Использование перегрузки методов
2. Создать перегрузку встроенного метода `IndexOf` класса `String` пространства имен `System`

Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК. 01.01 «Разработка программных модулей»

#### Теоретическое занятие 8. Операции класса

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;

- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что называется оператором?
2. Что такое операции класса?
3. Перечислите операции, которые производят с классами
4. Что такое перегрузка операторов?

Лабораторная работа 2. Определение операций в классе

**Форма контроля:** Рубежный контроль

**Задание:**

1. Написать программу бинарной операции
2. Создать класс-массив, элементы которого должны находиться в диапазоне [0,100]. Кроме того, при доступе к элементу проверяется, не вышел ли индекс за допустимые границы

3. Определение операции сложения для класса SafeArray из задания 2.

Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК. 01.01 «Разработка программных модулей»

Теоретическое занятие 9. Иерархия классов

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Для каких целей используется наследование классов?
2. Для чего может создаваться класс, от которого нельзя наследовать?
3. Можно ли использовать сочетание спецификаторов sealed и abstract при описании класса, и почему?
4. Что могут иметь классы в иерархии классов?

Лабораторная работа 3. Создание наследованных классов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать базовый класс Student, который будет содержать информацию о студенте (фамилия, курс обучения, номер зачетной книги). С помощью механизма наследования реализовать класс Aspirant (аспирант – студент, который готовится к защите кандидатской диссертации). Класс Aspirant есть производным от класса Student.

2. Задан класс Book, который описывает книгу. Класс содержит следующие элементы: название книги; фамилия и имя автора; стоимость книги. В классе Book нужно реализовать следующие методы: конструктор с 3 параметрами; свойства get/set для доступа к полям класса; метод Print(), который выводит информацию о книге. Разработать класс BookGenre, который наследует возможности класса Book и добавляет поле жанра (genre).

Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК. 01.01 «Разработка программных модулей»

#### Теоретическое занятие 10. Синтаксис интерфейсов

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

##### **Вопросы:**

1. Что такое интерфейс?
2. Какие объявления могут входить в декларацию интерфейса?
3. В чем отличия интерфейса от абстрактного класса?
4. Назовите правила реализации классом интерфейса.

#### Практическая работа 8. Работа с объектами через интерфейсы

**Форма контроля:** Рубежный контроль

**Задание:**

1. Применение интерфейса в программе
2. Интерфейс для класса «Матрица»
3. Раскрыть интерфейс для класса «Полином»

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

#### Теоретическое занятие 11. Интерфейсы и наследование

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

##### **Вопросы:**

1. Что такое наследование интерфейсов?
2. Что такое наследование реализации?
3. Что такое наследование специфицированной функциональности?
4. Какие механизмы C# обеспечивают реализацию наследования специфицированной функциональности?

### Практическая работа 9. Использование стандартных интерфейсов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Написать программу, реализующую интерфейс

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

### Теоретическое занятие 12. Структуры

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое структура в программировании?
2. Что является особенностями структуры?
3. Каким образом происходит преобразование структуры в класс?

### Практическая работа 10. Работа с типом данных структура

**Форма контроля:** Рубежный контроль

**Задание:**

1. Программа применения структуры

2. Известно, что точка на экране задается парой целых чисел  $(X, Y)$  в пикселях. Объединим эти два числа в структуру *точка*. Установим максимальную защиту (*private* – по умолчанию) для этих полей. Добавим конструктор с параметрами *точка(x,y)* для инициализации объектов структуры *точка*, а также метод *Get()* для извлечения полей в строковом формате.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

### Практическая работа 11. Работа с типом данных структура

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать структуру студент
2. Создать структуру данных, которая хранит информацию о банковском счете – его номер, тип и баланс. Создать переменную такого типа, заполнить структуру значениями и напечатать результат
3. Создать структуру работник с двумя полями: имя, ВУЗ. Заполнить структуру данными.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

### Теоретическое занятие 13. Делегаты.



**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что называется делегатом?
2. С какой целью создаются и применяются делегаты?
3. Назовите статические методы делегатов.
4. Что задает делегат в языке C#?
5. Что позволяет реализация экземпляра делегата в виде метода-свойства?

#### Теоретическое занятие 14. Коллекции. Параметризованные классы

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое коллекции?
2. Какое используется пространство имен для работы с коллекциями?
3. Что такое параметризованные классы? С какой целью они используются?

#### Практическая работа 12. Коллекции. Параметризованные классы

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создание и применение двух коллекций

2. Для заданной предметной области создать программу, состоящую из трех-пяти классов. Каждый из создаваемых классов должен иметь не менее трёх методов, свойств, конструкторов. Предусмотреть использование типа данных – перечисление, коллекций `List<T>` (варианты заданий с нечётными номерами), `Dictionary<TKey, TValue>` (варианты заданий с чётными номерами). Ввод/вывод данных должен быть реализован вне классов.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

#### Теоретическое занятие 15. Регулярные выражения

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;

- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое регулярные выражения?
2. Для чего предназначены регулярные выражения?
3. Какие действия можно выполнить над строками с помощью регулярных выражений?
4. Что является главным преимуществом регулярных выражений?

Практическая работа 13. Использование регулярных выражений

**Форма контроля:** Рубежный контроль

**Задание:**

1. Поиск первого соответствия шаблону
2. Поиск всех соответствий шаблону
3. Индивидуальные задания

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Теоретическое занятие 16. Указатели

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое указатель?
2. Как объявляются указатели?
3. Какие методы и операции можно применять к указателям?

Теоретическое занятие 17. Операции со списками

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое списки?
2. Какой класс используется для работы со списками?
3. Как можно объявить списки?
4. Какие операции можно производить со списками?

Практическая работа 14. Операции со списками.

**Форма контроля:** Рубежный контроль

**Задание:**

1. Реализовать список
2. Демонстрационная программа, реализующая операции создания, обработки, просмотра содержимого списка произвольного вида

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 15. Операции со списками.

**Форма контроля:** Рубежный контроль

**Задание:**

1. Использование списковой структуры стека для решения задачи проверки баланса скобок различного вида в тексте.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Самостоятельная работа 2. Объектно-ориентированное программирование

**Форма контроля:** Рубежный контроль

**Задание:**

1. Подготовить конспект по теме «Объектно-ориентированное программирование». Рассмотреть что относится к данному виду программирования, изучить принципы ООП.

Порядок выполнения работы согласно методическим рекомендациям по самостоятельным работам по МДК. 01.01 «Разработка программных модулей».

Теоретическое занятие 18. Назначение и виды паттернов

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое паттерн проектирования?
2. Каково назначение паттернов проектирования?
3. Назовите виды паттернов проектирования.
4. Как выбрать нужный паттерн?

Теоретическое занятие 19-20. Основные шаблоны

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;

- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. В чем заключается общий принцип описания паттернов проектирования?
2. Что такое основной шаблон?
3. Назовите основные отношения.

Практическая работа 16. Использование основных шаблонов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Отношение наследования
2. Отношение реализации
3. Отношение ассоциации
4. Отношение композиции
5. Отношение агрегации

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Теоретическое занятие 21. Порождающие шаблоны

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое порождающий паттерн?
2. Назовите порождающие паттерны.
3. В каком случае применяются порождающие паттерны?

Теоретическое занятие 22. Порождающие шаблоны

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Охарактеризуйте паттерн Абстрактная фабрика.
2. Охарактеризуйте паттерн Строитель.
3. Охарактеризуйте паттерн Прототип.
4. Охарактеризуйте паттерн Одиночка.

Практическая работа 17. Использование порождающих шаблонов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Реализация паттерна Builder
  2. Используя средства языка C# и возможности паттерна Builder разработать программу для генерирования объектов, являющихся массивами случайных чисел
  3. Написать программу на C#, реализующую паттерн Singleton
- Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Теоретическое занятие 23. Структурные шаблоны

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое структурный паттерн?
2. Какова цель структурных шаблонов?
3. Что относится к структурным шаблонам?
4. Охарактеризуйте каждый вид структурного шаблона.

Теоретическое занятие 24. Структурные шаблоны

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 15 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Каково назначение структурного шаблона Адаптер?
2. Каким образом осуществляется взаимодействие клиента с адаптируемым классом в шаблоне Адаптер?
3. В чём заключается различие между Адаптером объектов и Адаптером классов?
4. Какая проблема решается с помощью шаблона Фасад?
5. В чём заключается решение, предлагаемое в шаблоне Фасад?
6. К каким последствиям приводит использование шаблона Фасад?

Практическая работа 18. Использование структурных шаблонов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Реализация паттерна Адаптер для класса.
2. Реализацию паттерна Адаптер для объекта

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Теоретическое занятие 25. Поведенческие шаблоны

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое поведенческий паттерн?
2. Какова цель поведенческих шаблонов?
3. Что относится к поведенческим шаблонам?
4. Охарактеризуйте каждый вид поведенческого шаблона.

Теоретическое занятие 26. Поведенческие шаблоны

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Для чего предназначен поведенческий шаблон Состояние?
2. Какое решение предлагается в шаблоне Состояние?
3. Что понимают под стратегией в шаблоне Стратегия?
4. Какое решение предлагается в шаблоне Стратегия?
5. Какую проблему позволяет решить шаблон Шаблонный метод?

Практическая работа 19. Использование поведенческих шаблонов

**Форма контроля:** Рубежный контроль

**Задание:**

1. В классе реализовать интерфейс IAggregate. В этом интерфейсе объявляется единственный метод CreateIterator(). Этот метод возвращает итератор для контейнера

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Теоретическое занятие 27. Событийно-управляемое программирование

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;

- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что называется событием?
2. В чем состоит суть событийного программирования?
3. Назовите типовые события для формы, кнопок и строки редактирования?
4. Поясните порядок разработки метода для обработки события.
5. Поясните порядок разработки метода для обработки группы идентичных событий.

Теоретическое занятие 28. Событийно-управляемое программирование

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое событийно-управляемое программирование?
2. Что называется объектом?
3. Какими параметрами характеризуется каждый объект?
4. Что называется свойством объекта?
5. Что называется методом объекта?

Теоретическое занятие 29. Элементы управления. Диалоговые окна.

Обработчики событий.

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое элементы управления?
2. С какой целью применяются элементы управления при программировании?
3. Назовите элементы управления, которые используются при разработке программы?
4. Для чего применяется элемент Button?

Теоретическое занятие 30. Элементы управления. Диалоговые окна.

Обработчики событий.

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое диалоговое окно?
2. Каким методом возвращаются результаты диалогового окна в виде значений?
3. Назовите статический метод, использующийся для отображения стандартных окон сообщений Windows.

Теоретическое занятие 31. Элементы управления. Диалоговые окна. Обработчики событий

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое обработчик событий?
2. Для чего разрабатываются обработчики событий?
3. Назовите операции и методы, допустимые при работе с обработчиками событий.
4. К каким элементам управления можно установить обработчики событий?

Практическая работа 20. Разработка приложения с использованием текстовых компонентов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать программу, которая при нажатии на кнопку «Output» выводит сообщение «Моя первая программа на языке C#», а затем при повторном нажатии на эту же кнопку сообщение исчезает. При повторном выводе цвет надписи должен быть красным.

2. Разработать программу, которая при вводе текста в первый компонент textBox1, во втором компоненте textBox 2 отображает реальную длину вводимой строки. Кроме этого, при выходе из компонента textBox 1 его содержимое копируется в буфер обмена и удаляется, а при возвращении в программу появляется снова.

3. Разработать программу, которая запрещает ввод в компонент textBox1 подряд двух одинаковых символов.

4. Разработать программу, которая при нажатии на кнопку заполняет текстовое поле фразой «Специальность Информационные системы и программирование».

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»



Практическая работа 21. Разработка приложения с использованием текстовых компонентов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать программу, которая считает количество нажатий на кнопку и выдает это значение в компоненте textBox.
2. Разработать программу, которая считывает строку под определенным номером и помещает её в текстовое поле.
3. Разработать программу, которая сохраняет текст, набранный в поле textBox1 в файл, имя которого задано в текстовом поле textBox2.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 22. Разработка приложения с использованием текстовых компонентов

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать приложение, с помощью которого можно вычислить время падения тела с некоторой высоты при условии, что высота может задаваться в метрах, сантиметрах и дюймах
2. Разработайте приложение, которое при выборе определенного цвета в выпадающем списке, изменяет цвет формы
3. Разработать приложение «Конвертер валют» - перевод из одной валюты в другую.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 23. Разработка приложения с несколькими формами

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать проект с возможностью группировки элементов на вкладках.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 24. Разработка приложения с несколькими формами

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать демонстрационное приложение, осуществляющее вызов из главной формы второстепенной формы по схеме. Приложение реализует взаимодействие между различными формами, которыми могут быть диалоговые окна любой сложности

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 25. Разработка приложения с несколькими формами

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать приложение в соответствии с приведенной формой, каждая кнопка открывает дополнительную форму

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 26. Разработка приложения с не визуальными компонентами

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать приложение «Таймер»

2. Разработать приложение «Часы»

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 27. Разработка приложения с не визуальными компонентами

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать проект демонстрации работы RGB – функций (установок цвета по трем составляющим) с помощью полос прокрутки. Каждый бегунок полос прокрутки должен будет менять вклад RGB – компонента, отображающийся на панели как цвет, а на метке как число. Результирующий цвет должен отображаться на панели

2. Разработать проект, который позволяет пользователю вычислить факториал числа. Число, для которого рассчитывается факториал, выбирается с помощью элемента управления TrackBar. При щелчке по кнопке «Расчет», меняется надпись «Число n» на «N!» и в строке ввода выводится значение факториала числа

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 28. Разработка приложения с не визуальными компонентами

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создание невизуального компонента Planets

2. Создайте приложение, содержащее визуальные и невизуальные компоненты

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Теоретическое занятие 32. Введение в графику

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Какое пространство имен используется для добавления графики в код программы?
2. Для чего предназначены все методы класса Graphics?
3. Какая панель используется для добавления графики?

Теоретическое занятие 33-34. Введение в графику

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. В каком обработчике событий будут создаваться фигуры?
2. На каком компоненте основана реализация движения объекта?
3. С помощью какого компонента регулируется скорость движения объекта?

Практическая работа 29. Разработка приложения с графикой и анимацией

**Форма контроля:** Рубежный контроль

**Задание:**

1. Реализовать эффект движения простейшего геометрического объекта (круга) внутри области рисования с «отскакиванием от стенок». Предусмотреть возможность изменения его цвета и скорости движения. Изменение цвета круга происходит при щелчке в нем левой кнопкой мыши

2. В каждом варианте необходимо реализовать движение требуемого изображения по заданной траектории, не допуская при этом выход фигуры за пределы области рисования. Каждое изображение состоит из нескольких (двух-трех) элементарных фигур. Предусмотреть возможность изменения скорости движения с помощью компонента TrackBar.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 30. Разработка приложения с графикой и анимацией

**Форма контроля:** Рубежный контроль

**Задание:**

1. С помощью графических операторов в с# сделать часы
2. Разработайте приложение по индивидуальным заданиям

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 31. Разработка приложения с графикой и анимацией

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать приложение, реализующее простой графический редактор. Функциями этого редактора должны быть: открытие рисунка, рисование поверх него простой кистью, сохранение рисунка в другой файл.

2. Добавьте в приведенный графический редактор свои функции в соответствии с вариантом

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 32. Разработка игрового приложения

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать простой шутер «Вторжение НЛО» («UFO Invasion»), в котором придется отражать нашествие на Землю армады НЛО.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 33. Разработка игрового приложения

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать простой шутер «Вторжение НЛО» («UFO Invasion»), в котором придется отражать нашествие на Землю армады НЛО.

2. Развить продолжение игры, созданной в Практических работах № 32 и № 33, используя любое направление

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 34. Разработка игрового приложения

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать игру «Мозаика». Игра Мозаика похожа на пятнашки, только вместо цифр картинка. Загружается желаемое изображение. Изображение делится на сегменты, сегменты перемешиваются, один сегмент исчезает. Необходимо передвигая сегменты на пустое место собрать картинку полностью.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Теоретическое занятие 35. Методы оптимизации программного кода

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое оптимизация программного кода?
2. Назовите цели оптимизации программного кода.
3. Какое требование предъявляют к методу оптимизации?
4. На чем в основном основывается оптимизация программного кода?

Теоретическое занятие 36. Методы оптимизации программного кода

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Назовите виды оптимизации программного кода.
2. Какими способами может проводиться оптимизация кода?
3. В чем состоят недостатки ручной оптимизации кода?
4. В чем состоят недостатки автоматизированной оптимизации кода?

Теоретическое занятие 37. Методы оптимизации программного кода

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Опишите алгоритм проведения оптимизации программного кода
2. Перечислите уровни оптимизации программного кода
3. Что такое сверстка констант?
4. Что такое устранение общих подвыражений?

Теоретическое занятие 38. Цели и методы рефакторинга

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое рефакторинг кода?

2. Охарактеризуйте концепцию рефакторинга.
3. В чем заключается проблема применения рефакторинга?
4. Назовите правила рефакторинга.

#### Теоретическое занятие 39. Цели и методы рефакторинга

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Перечислите принципы рефакторинга.
2. Зачем нужно проводить рефакторинг?
3. В каком случае следует проводить рефакторинг?
4. Когда рефакторинг не нужен?

#### Теоретическое занятие 40. Цели и методы рефакторинга

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Почему рефакторинг приносит результаты?
2. Каким образом рефакторинг влияет на производительность программы?
3. Что является важным предварительным условием перед проведением рефакторинга?
4. По каким признакам можно определить, что необходимо проводить рефакторинг?
5. Назовите методы рефакторинга.

#### Практическая работа 35. Оптимизация и рефакторинг кода

**Форма контроля:** Рубежный контроль

**Задание:**

1. Модифицировать метод Main(), добавив в него код
2. Выберите любую созданную ранее программу и оптимизируйте.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

#### Практическая работа 36. Оптимизация и рефакторинг кода

**Форма контроля:** Рубежный контроль

**Задание:**

1. Изучить код программы, исправить при необходимости ошибки и оптимизировать данный код

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

### Практическая работа 37. Оптимизация и рефакторинг кода

**Форма контроля:** Рубежный контроль

**Задание:**

1. Оптимизируйте код игрового приложения «Вторжение НЛО»
2. Оптимизируйте код программы, написанной в Практической работе № 6 по созданию класса, чтобы он был создан в Windows Form.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

### Теоретическое занятие 41. Разработка пользовательского интерфейса

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое пользовательский интерфейс?
2. Какие виды пользовательского интерфейса вы знаете?
3. Какими требованиями должен обладать пользовательский интерфейс?
4. Что такое интерфейс командной строки?
5. Что такое графический пользовательский интерфейс?

### Теоретическое занятие 42. Разработка пользовательского интерфейса

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Назовите элементы графического пользовательского интерфейса.
2. Что такое меню?
3. Назовите компоненты графического пользовательского интерфейса.

### Теоретическое занятие 43. Разработка пользовательского интерфейса

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;

- 2) время, отводимое на опрос – 15 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Как создать верхнее меню?
2. Как добавить выпадающее меню в верхнее меню?
3. Как установить определенному пункту меню сочетание клавиш?
4. Как создать панель инструментов?
5. Как добавить несколько кнопок на панель инструментов?
6. Как разместить на кнопке изображение?

Теоретическое занятие 44. Разработка пользовательского интерфейса

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Каким должен быть хороший интерфейс?
2. Что является основными элементами графического интерфейса?
3. Для чего предназначены диалоговые окна?
4. На основе какого интерфейса строится взаимодействие человека и компьютера?

Теоретическое занятие 45. Разработка пользовательского интерфейса

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. В виде чего представляются объекты объектно-ориентированного интерфейса?
2. Какие типы окон можно выделить в пользовательском интерфейсе?
3. Укажите основное свойство GUI-интерфейса?
4. С чего начинается разработка пользовательского интерфейса?
5. На что должен быть ориентирован пользовательский интерфейс?

Практическая работа 38. Разработка интерфейса пользователя

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать приложение SimpleNotepad, представляющее собой простейший текстовый редактор. Использовать в приложении компонент для работы с меню и стандартные окна диалога



Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 39. Разработка интерфейса пользователя

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать приложение SimpleNotepad, представляющее собой простейший текстовый редактор. Использовать в приложении компонент для работы с меню и стандартные окна диалога

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 40. Разработка интерфейса пользователя

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создание панели инструментов приложения из Практической работе № 39.

2. Создание строки состояния

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 41. Разработка интерфейса пользователя

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создания многооконного приложения с фреймами

2. Создание методов для элемента управления TreeView

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Практическая работа 42. Разработка интерфейса пользователя

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создайте приложение, демонстрирующее особенности работы с несколькими окнами

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

Самостоятельная работа 3. Интерфейс пользователя

**Форма контроля:** Рубежный контроль

**Задание:**

1. Подготовить презентацию по одной из следующих тем:

1. Командный интерфейс

2. Графический интерфейс

Порядок выполнения работы согласно методическим рекомендациям по самостоятельным работам по МДК. 01.01 «Разработка программных модулей».

#### Теоретическое занятие 46. Работа с базами данных

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что собой представляет ADO.Net?
2. Для чего необходим провайдер данных?
3. Какие встроенные провайдеры имеются в ADO.Net?

#### Теоретическое занятие 47. Работа с базами данных

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что собой представляет работа с базами данных на уровне приложения .Net?
2. Что обеспечивают функциональные особенности этой системы взаимодействующих классов?
3. По какой схеме можно получить и отобразить данные?

#### Теоретическое занятие 48. Доступ к данным

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Для чего предназначена объектная модель ADO.Net?
2. На чем основана концепция доступа к данным в ADO .NET?
3. Что предполагает объектная модель ADO .NET?

#### Теоретическое занятие 49. Доступ к данным

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

- 1.Что необходимо знать, чтобы подключиться правильно к базе данных?
- 2.Назовите встроенные объекты ADO.Net для подключения к БД.
- 3.Охарактеризуйте эти объекты.

#### Практическая работа 43. Создание приложения с БД

**Форма контроля:** Рубежный контроль

**Задание:**

1. База данных из двух связанных таблиц «Телефонная книжка» создана в Microsoft Access. Реализовать с помощью управляемого провайдера OLE DB доступ к этой базе данных

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

#### Практическая работа 44. Создание приложения с БД

**Форма контроля:** Рубежный контроль

**Задание:**

1. Написать приложение для работы с БД MS Access, которое сможет обновлять, удалять и вставлять данные

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

#### Практическая работа 45. Создание приложения с БД

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создадим простое приложение баз данных, которое выводит на экранную форму информацию из таблицы «Туристы» и связанную с текущей записью таблицы «Туристы» запись таблицы «Информация о туристах» из базы данных Microsoft Access.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

#### Практическая работа 46. Создание приложения с БД

**Форма контроля:** Рубежный контроль

**Задание:**

1. В каждом варианте необходимо разработать базу данных минимум из двух связанных между собой таблиц. В каждой таблице – не менее трех полей. Реализовать доступ к созданной базе с помощью управляемого провайдера OLE DB.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

#### Теоретическое занятие 50-51. Создание таблицы, работа с записями.

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Как создать таблицы в технологии ADO.Net?
2. Какие команды используются для работы с записями?
3. Как создать базу данных в Visual Studio?

Теоретическое занятие 52. Способы создания команд

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 20 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Пул соединений – это элемент архитектуры ADO.NET, или поддержка пула зависит от поставщика данных? *(Зависит от поставщика данных)*
2. Можно ли при помощи класса Connection выполнить команду, создающую в базе данных хранимую процедуру? *(да)*
3. Какое условие должно выполняться, чтобы в таблице работал метод поиска строк Find()? *(должен быть задан первичный ключ)*
4. Предположим, что был создан объект-команда. После того, как команда выполнялась хотя бы один раз, можно ли изменить у объекта-команды то соединение, с которым он работает? *(да)*
5. Зависит ли от поставщика данных тип для описания параметров команды?
6. Можно ли при помощи одного вызова метода Fill() адаптера данных заполнить в рассоединенном наборе несколько таблиц? *(да)*
7. Ридер данных зависит от поставщика данных или он универсален? *(зависит от поставщика)*
8. В DataTable программно добавлена новая строка. Верно ли, что у такой строки будут совпадать версии Current и Original? *(нет)*
9. Может ли ридер использоваться для чтения данных, если они получены одной командой из нескольких таблиц БД? *(да)*
10. При заполнении DataSet использовалась команда SELECT с выборкой из нескольких таблиц. Удается ли в этом случае построить команды обновления базы при помощи класса CommandBuilder? *(нет)*

Теоретическое занятие 53. Способы создания команд

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 20 мин;
- 3) максимальный балл за задание – 5 баллов

### **Вопросы:**

1. Можно ли при помощи метода ExecuteScalar() выполнить команду с текстом Select \* from tbl? *(да)*
2. Класс SqlCommand различает параметры команды по номеру позиции или по имени? *(по имени)*
3. Может ли объект DataColumn являться частью первичного ключа таблицы, если его тип данных – string? *(да)*
4. Верно ли, что вызов метода адаптера FillSchema() кроме прочего заполнит в DataSet коллекцию Relations, то есть опишет связи между таблицами? *(нет)*
5. Может ли ссылка на один объект DataColumn быть помещена в несколько разных коллекций Columns одновременно? *(нет)*
6. Имеет ли соединение информацию о том, с какой командой оно связано? *(нет)*
7. Если выполняется метод команды ExecuteReader(), то соединение с базой устанавливается автоматически? *(нет)*
8. У строки вызывается метод GetChildRows(). Что принимает этот метод в качестве параметра? *(Имя связи (строка) или объект, описывающий связь между таблицами)*
9. Если тип команды – StoredProcedure, нужно ли указывать в тексте команды CommandText имена параметров процедуры? *(нет)*
10. Можно ли у DataView задать свойство Sort, не задавая свойство RowFilter? *(да)*

### Практическая работа 47. Создание запросов к БД

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать запросы к БД «Туристы» из Практической работы № 45.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

### Практическая работа 48. Создание запросов к БД

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать запросы к БД по индивидуальным заданиям из Практической работы № 46.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.01 «Разработка программных модулей»

### Лабораторная работа 4. Создание хранимых процедур

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создание хранимых процедур к базе данных из Практической работы № 43
2. Создать хранимые процедуры к БД «Туристы», используя следующие примеры.

Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК. 01.01 «Разработка программных модулей»

Лабораторная работа 5. Создание хранимых процедур

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать хранимые процедуры к БД по индивидуальным заданиям из Практической работы № 44.

Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК. 01.01 «Разработка программных модулей»

### **3.Критерии оценки**

#### **3.1. Инвариантные критерии оценки**

**Критерии оценки устных (письменных) ответов на теоретические вопросы**

<b>Критерии оценки</b>		<b>Оценка</b>
<b>1</b>	Демонстрирует глубокое, полное знание и понимание программного материала. Последовательно, самостоятельно раскрывает основное содержание вопроса. Выводы аргументированы, основаны на самостоятельно выполненном анализе, обобщении данных. Четко и верно даны определения понятий и научных терминов. Дает верные, самостоятельные ответы на вопросы.	5 (отлично)
<b>2</b>	Демонстрирует недостаточно глубокое, полное знание и понимание программного материала. Недостаточно последовательно, но самостоятельно раскрывает основное содержание вопроса. Выводы основаны на самостоятельно выполненном анализе, обобщении данных, но в отдельных случаях недостаточно аргументированы. Недостаточно четко и верно даны определения понятий и научных терминов. При ответе на вопросы допускает несущественные ошибки, которые может исправить самостоятельно.	4 (хорошо)
<b>3</b>	Демонстрирует в отдельных вопросах, неглубокое владение знаниями программного материала. Излагает программный материал фрагментарно, не всегда последовательно. Допущены ошибки и неточности в использовании научной терминологии. При ответе на вопросы допускает неточности.	3 (удовлетворительно)
<b>4</b>	Студент демонстрирует незнание и непонимание программного материала.	2 (неудовлетворительно)

Основное содержание учебного материала не раскрыто; допущены грубые ошибки в определении понятий, при использовании терминологии. Затрудняется отвечать на вопросы, при ответе допускает серьезные ошибки.	тельно)
---	---------

### Критерии оценки результатов выполнения тестового задания

Оценка	Количество правильных ответов на вопросы в % соотношении от общего числа вопросов
Оценка 5 «отлично»	90-100%
Оценка 4 «хорошо»	76-89%
Оценка 3 «удовлетворительно»	50-75%
Оценка 2 «неудовлетворительно»	≤ 49%

### Критерии оценки результатов промежуточной аттестации:

Критерии оценки		Оценка
1	обучающийся показал полный объем знаний по вопросу, владеет культурой общения, навыками научного изложения материала, устанавливает связь между теоретическими знаниями и способами практической деятельности; ясно, точно и логично отвечает на заданные вопросы.	5 (отлично)
2	обучающийся логично и научно изложил материал, но недостаточно полно определяет практическую значимость теоретических знаний; не высказывает своей точки зрения по данному вопросу, не смог дать достаточно полного ответа на поставленные вопросы	4 (хорошо)
3	обучающийся при раскрытии вопроса допустил содержательные ошибки, испытывает затруднения при ответе на большинство вопросов.	3 (удовлетворительно)
4	обучающийся показал слабые теоретические и практические знания, допустил грубые ошибки при раскрытии вопроса, не смог ответить на заданные вопросы	2 (неудовлетворительно)

## РУБЕЖНЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Что такое модель жизненного цикла программного обеспечения

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.
- б) модель содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.
- в) действия содержащие процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.
- г) структура, содержащая процессы задачи, которые осуществляются в ходе использования и сопровождения программного продукта.
- д) структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки.

2. Какая это модель жизненного цикла программного обеспечения?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2



- а) Каскадная модель (водопад)
- б) Каскадная модель с промежуточным контролем
- в) V модель
- г) Модель кодирования и устранения ошибок
- д) Спиральная модель жизненного цикла ПО

3. Дана модель:

- 1-Постановка задачи
  - 2-Выполнение
  - 3-Проверка результата
  - 4-При необходимости переход к первому пункту
- Выберите название данной модели

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) Каскадная модель
- б) Модель кодирования и устранения ошибок
- в) V-модель
- г) Каскадная модель с промежуточным контролем
- д) Спиральная модель

4. Количество стадий, разработки программного обеспечения



**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) 7
- б) 6
- в) 3
- г) 5
- д) 4

5. С какого момента появляется жизненный цикл?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) с момента осознания появления данного ПО;
- б) с момента его полного выхода из употребления.
- в) с момента начала разработки программного обеспечения

6. Как расшифровывается ООП?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) основы объектного программирования
- б) объектно-ориентированное программирование
- в) основное ориентированное программирование

7. Механизм программирования, объединяющий вместе код и данные, которыми он манипулирует, исключая как вмешательство извне, так и неправильное использование данных.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) инкапсуляция
- б) полиморфизм
- в) наследование

8. Позволяет расширять поведение базового (или родительского) класса, наследуя основную функциональность в производном подклассе (также именуемом дочерним классом)

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) инкапсуляция
- б) полиморфизм
- в) наследование

9. Свойство, которое позволяет одно и тоже имя использовать для решения нескольких технически разных задач.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) инкапсуляция

- б) полиморфизм  
в) наследование

10. Разновидность абстрактного типа данных в ООП, характеризующийся способом своего построения.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) класс  
б) объект  
в) метод

**Критерии оценивания:**

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	Менее 50

**Ключ ответов к тестовым заданиям**

№ вопроса	Правильный ответ	Проверяемые компетенции
1	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
2	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
3	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
4	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
5	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
6	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
7	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
8	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
9	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
10	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

## МЕЖСЕССИОННЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Что такое модель жизненного цикла программного обеспечения

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) структура, содержащая процессы задачи, которые осуществляются в ходе использования и сопровождения программного продукта.
- б) модель содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.
- в) действия содержащие процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.
- г) структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.
- д) структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки.

2. Количество стадий, разработки программного обеспечения

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) 7
- б) 6
- в) 5
- г) 4

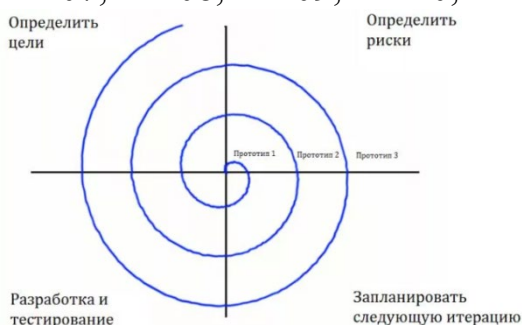
3. С какого момента появляется жизненный цикл?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) с момента осознания появления данного ПО;
- б) с момента его полного выхода из употребления.
- в) с момента начала разработки программного обеспечения

4. Какая это модель жизненного цикла программного обеспечения?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2



- а) Каскадная модель (водопад)
- б) Каскадная модель с промежуточным контролем
- в) V модель
- г) Модель кодирования и устранения ошибок
- д) Спиральная модель жизненного цикла ПО

5. Дана модель:

1-Постановка задачи

2-Выполнение

3-Проверка результата

4-При необходимости переход к первому пункту

Выберите название данной модели

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

а) Каскадная модель

б) V-модель

в) Модель кодирования и устранения ошибок

г) Каскадная модель с промежуточным контролем

д) Спиральная модель

6.Как расшифровывается ООП?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

а) основы объектного программирования

б) основное ориентированное программирование

в) объектно-ориентированное программирование

7. Механизм программирования, объединяющий вместе код и данные, которыми он манипулирует, исключая как вмешательство извне, так и неправильное использование данных.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

а) инкапсуляция

б) полиморфизм

в) наследование

8. Свойство, которое позволяет одно и тоже имя использовать для решения нескольких технически разных задач.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

а) инкапсуляция

б) полиморфизм

в) наследование

9. Позволяет расширять поведение базового (или родительского) класса, наследуя основную функциональность в производном подклассе (также именуемом дочерним классом)

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

а) инкапсуляция

б) полиморфизм

в) наследование

10. Отметьте корректные объявления делегата – функционального класса:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) `public class D = delegate void(string s)`
- б) `Delegate D1 = new Delegate( void(string s)`
- в) `public delegate void D2 (string s)`
- г) `delegate int T(int x1, ref int x2)`
- д) `public delegate T1(int x)`

11. Особенности структур являются

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) структуры не могут наследовать интерфейсы
- б) структуры могут наследовать интерфейсы
- в) поля структуры не могут быть инициализированы в момент объявления
- г) для структур нельзя создать собственный конструктор без аргументов

12. Реализация экземпляра делегата в виде метода-свойства позволяет:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) скрыть его от клиента
- б) иметь один экземпляр вместо нескольких
- в) динамически связывать экземпляр с методом и экономить память

13. Какие методы делегатов являются статическими?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) `Combine` и `Remove`
- б) `Clone`
- в) `GetInvocationList` и `GetHashCode`

14. Делегат в языке C# задает:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) класс функций с заданной сигнатурой
- б) функцию с заданной сигнатурой и с заданной реализацией
- в) класс, у которого все методы представляют собой функции

15. Экземпляр делегата следует рассматривать как:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) единственную нетипизированную ссылку на метод
- б) единственную ссылку на метод с жестко фиксированной сигнатурой, заданной делегатом

в) совокупность ссылок - на метод с жестко фиксированной сигнатурой, заданной делегатом, на объект, вызвавший метод, на список присоединенных делегатов

#### Критерии оценивания:

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	Менее 50

#### Ключ ответов к тестовым заданиям

№ вопроса	Правильный ответ	Проверяемые компетенции
1	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
2	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
3	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
4	Д	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
5	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
6	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
7	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
8	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
9	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
10	А, В, Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
11	Б, В, Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
12	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
13	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
14	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2
15	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

## ИТОГОВЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Что такое модель жизненного цикла программного обеспечения

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.
- б) модель содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.
- в) действия содержащие процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.
- г) структура, содержащая процессы задачи, которые осуществляются в ходе использования и сопровождения программного продукта.
- д) структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки.

2. С какого момента появляется жизненный цикл?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) с момента его полного выхода из употребления
- б) с момента осознания появления данного ПО
- в) с момента начала разработки программного обеспечения

3. Механизм программирования, объединяющий вместе код и данные, которыми он манипулирует, исключая как вмешательство извне, так и неправильное использование данных.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) инкапсуляция
- б) полиморфизм
- в) наследование

4. Позволяет расширять поведение базового (или родительского) класса, наследуя основную функциональность в производном подклассе (также именуемом дочерним классом)

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.1, ПК 1.2

- а) инкапсуляция
- б) полиморфизм
- в) наследование

5. Свойство, которое позволяет одно и тоже имя использовать для решения нескольких технически разных задач.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, , ПК 1.1, ПК 1.2

- а) инкапсуляция
- б) полиморфизм
- в) наследование

6. Особенности структур не являются

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) структуры не могут наследовать интерфейсы
- б) структуры могут наследовать интерфейсы
- в) поля структуры не могут быть инициализированы в момент объявления
- г) для структур нельзя создать собственный конструктор без аргументов

7. Делегат в языке C# задает:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) класс функций с заданной сигнатурой
- б) функцию с заданной сигнатурой и с заданной реализацией
- в) класс, у которого все методы представляют собой функции

8. Экземпляр делегата следует рассматривать как:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) единственную нетипизированную ссылку на метод
- б) единственную ссылку на метод с жестко фиксированной сигнатурой, заданной делегатом
- в) совокупность ссылок - на метод с жестко фиксированной сигнатурой, заданной делегатом, на объект, вызвавший метод, на список присоединенных делегатов

9. Какими параметрами характеризуется каждый объект

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) класс, событие, метод
- б) свойство, событие, окно
- в) свойство, метод, событие

10. Это программирование заключается в описании действий, которые должны быть выполнены при возникновении того или иного события для каждого объекта приложения

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) объектно-ориентированное
- б) событийно-управляемое



- в) визуальное
- г) событийное

11. Является основным элементом управления пользовательского интерфейса, предназначена для того, чтобы принимать команды пользователя и выполнять код, когда элемент щелкают мышью

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) *Button*
- б) *CheckBox*
- в) *TextBox*

12. Используется для отображения текста только для чтения

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) *GroupBox*
- б) *TextBox*
- в) *Label*

13. Выпадающий список

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) *ComboBox*
- б) *CheckBox*
- в) *ListBox*

14. Обработчик события:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) всегда принадлежит классу, зажигающему событие
- б) никогда не принадлежит классу, зажигающему событие
- в) может принадлежать классу, зажигающему событие
- г) принадлежит только одному классу, слушающему событие

15. Укажите операции и методы, допустимые при работе с обработчиками событий:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) операция "+", присоединяющая обработчик к списку вызовов события
- б) операция "-", удаляющая обработчик из списка вызовов события
- в) операция присваивания "=", присоединяющая обработчик к списку вызовов события, предварительно очищая список вызовов, если он был не пуст
- г) метод *Combine*, являющийся аналогом операции "+"

16. Оптимизация – это...

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) Получение оптимальных результатов в определенных пределах;
- б) Целенаправленная деятельность, заключающаяся в получении наилучших результатов при соответствующих условиях;
- в) Ответы а и б – правильные;
- г) Правильного ответа нет.

17. Выберите уровни оптимизации программного кода

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) системный уровень, уровень приложения, микроархитектурный уровень
- б) производительный уровень, уровень улучшения кода
- в) системный уровень, архитектурный уровень, аналитический уровень

18. Процесс, включающий в себя поиск выражений (или подвыражений), включающих только константы.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) исключение недоступного кода
- б) распространения констант
- в) свертки констант

19. Оптимизация, которая заключается в следующем: если вы используете расчёт какого-либо выражения два или более раз, то его можно рассчитать один раз, а затем подставить во все использующие его выражения

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) устранение общих подвыражений
- б) распространение копий
- в) исключение недоступного кода

20. Совокупность средств и правил взаимодействия пользователя с компьютером называют:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) аппаратным интерфейсом;
- б) программным интерфейсом;
- в) объектом управления;
- г) пользовательским интерфейсом

21. Основными элементами графического интерфейса являются:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) окна и меню;

- б) папки и файлы;
- в) программы;
- г) рабочий стол и кнопка «Пуск»
- д) окна приложений;

22. Диалоговые окна предназначены для:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) для одностороннего взаимодействия человека и компьютера;
- б) для диалога человека и компьютера;
- в) для одностороннего взаимодействия компьютера и человека

23. Укажите основное свойство GUI-интерфейса?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) использование полосы прокрутки
- б) отображение пространства
- в) возможность непосредственного манипулирования

24. С чего начинается разработка пользовательского интерфейса?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) с обследования предметной области
- б) с ранних набросков и схем на этапе анализа требований
- в) с согласования сметы

25. Каким должен быть хороший интерфейс?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.1, ПК 1.2

- а) интуитивно понятным, предсказуемым, привлекательным
- б) уметь общаться с пользователем, кликабельные и некликабельные элементы должны одинаковыми
- в) иметь одинаковые стили для кнопок с разными типами действий

#### Критерии оценивания:

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	Менее 50

#### Ключ ответов к тестовым заданиям

№ вопроса	Правильный ответ	Проверяемые компетенции
-----------	------------------	-------------------------

[illegible]

23	B	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.1, ПК 1.2
24	A	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.1, ПК 1.2
25	A	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.1, ПК 1.2

**Вопросы для экзамена по МДК.01.01 «Разработка программных модулей»**

**Теоретические вопросы:**

1. Понятие ЖЦ ПО. Этапы ЖЦ ПО.
2. Технология структурного программирования
3. Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи
4. Основные принципы объектно-ориентированного программирования. Классы: основные понятия
5. Перегрузка методов
6. Операции класса. Иерархия классов
7. Синтаксис интерфейсов. Интерфейсы и наследование
8. Структуры. Делегаты.
9. Коллекции. Параметризованные классы
10. Регулярные выражения. Указатели. Операции со списками
11. Паттерны проектирования
12. Событийно-управляемое программирование
13. Элементы управления. Диалоговые окна. Обработчики событий
14. Введение в графику
15. Методы оптимизации программного кода
16. Цели и методы рефакторинга
17. Правила разработки интерфейсов пользователя
18. Основы ADO.Net

**Практические задания:**

1. Работа с классами
2. Определение операций в классе
3. Работа с объектами через интерфейсы
4. Работа с типом данных структура
5. Использование регулярных выражений
6. Операции со списками
7. Перегрузка методов
8. Использование основных шаблонов
9. Разработка приложения с использованием текстовых компонентов
10. Разработка приложения с несколькими формами
11. Разработка приложения с не визуальными компонентами
12. Разработка приложения с графикой и анимацией
13. Разработка игрового приложения
14. Оптимизация и рефакторинг кода
15. Разработка интерфейса пользователя
16. Создание приложения с БД
17. Создание запросов к БД
18. Создание хранимых процедур

# **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ОЦЕНКЕ КАЧЕСТВА ОСВОЕНИЯ СТУДЕНТАМИ ПРОГРАММ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА**

по междисциплинарному курсу  
МДК.01.02 «Поддержка и тестирование программных модулей»  
**Пояснительная записка**

Фонд оценочных средств разработан на основе рабочей программы МДК.01.02 «Поддержка и тестирование программных модулей» в соответствии с требованиями ФГОС СПО по специальности СПО 09.02.07 «Информационные системы и программирование» утверждённого приказом Министерства образования и науки РФ от 09.12.2016г., № 1547, ФГОС среднего общего образования и примерной основной образовательной программой.

## **1.Паспорт фонда оценочных средств для проведения контроля успеваемости**

### **1.1 Цели и задачи контроля**

Целью текущего контроля успеваемости обучающихся является обеспечение систематического контроля и оценки уровня освоения знаний и умений, уровня сформированности общих и профессиональных компетенций МДК.01.02 «Поддержка и тестирование программных модулей»

Главной задачей текущего контроля успеваемости является повышение мотивации обучающихся к регулярной учебной и самостоятельной работе, закрепление, углубление знаний, закрепление и совершенствование умений, обеспечение соответствия результатов обучения задачам будущей профессиональной деятельности посредством внедрения эффективной системы оценки в образовательный процесс. Общие компетенции, включающие в себя способность:

#### **Общие компетенции, включающие в себя способность:**

ОК 01 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК 02 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности

ОК 03 Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04.Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05.Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста

ОК 06.Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения

ОК 07.Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях

ОК 08.Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

ОК 09.Использовать информационные технологии в профессиональной деятельности.

ОК 10.Пользоваться профессиональной документацией на государственном и иностранном языках.

ОК 11.Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере.

**Профессиональные компетенции, соответствующие видам деятельности:**

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК1.4. Выполнять тестирование программных модулей

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

**Знания и умения:**

**уметь:**

- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;

- создавать программу по разработанному алгоритму как отдельный модуль;

- выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;

- уметь выполнять оптимизацию и рефакторинг программного кода;

- оформлять документацию на программные средства

**знать:**

- основные этапы разработки программного обеспечения;

- основные принципы технологии структурного и объектно-ориентированного программирования;

- способы оптимизации и приемы рефакторинга;

- основные принципы отладки и тестирования программных продуктов

## **1.2. Структура фонда оценочных средств**

Фонд оценочных средств для проведения текущего контроля включает в себя комплекты контрольно-оценочных средств, предназначенные для проведения текущего контроля в виде:

- оперативного контроля;
- рубежного контроля.



Оперативный контроль проводится в форме:

- опрос (устный);
- тестирование;
- выполнение письменной работы (графическая работа);
- выполнение практической работы.

Рубежный контроль проводится в форме: опрос (устный);  
тестирование;  
выполнение  
практической  
работы; защита портфолио

Фонд оценочных средств также, включает в себя комплект контрольно-оценочных средств для проведения промежуточной аттестации. (Приложение 1).

Фонд оценочных средств также, включает в себя комплект контрольно-оценочных средств для выполнения курсовой работы. (Приложение 2).

### 1.3. Система оценивания результатов выполнения заданий

Результаты выполнения заданий оцениваются в соответствии с разработанными критериями оценки. Используются следующие шкалы для оценивания результатов обучения: пятибальная шкала оценки; десятибалльная шкала оценки.

Перевод десятибалльной шкалы учета результатов в пятибалльную оценочную шкалу:

Оценка	Количество баллов, набранных за выполнение теоретического и практического задания
Оценка 5 «отлично»	90-100
Оценка 4 «хорошо»	76-89
Оценка 3 «удовлетворительно»	50-75
Оценка 2 «неудовлетворительно»	≤ 49

Перевод пятибалльной шкалы учета результатов в десятибалльную оценочную шкалу:

Оценка	Количество баллов, набранных за выполнение теоретического и практического задания, средний балл по итогам аттестации
Оценка 5 «отлично»	4,6-5
Оценка 4 «хорошо»	3,6-4,5

Оценка3«удовлетворительно»	3-3,5
Оценка2«неудовлетворительно»	≤2,9

#### **1.4. Материально-техническое обеспечение для проведения контроля**

Контроль проводится в учебном кабинете «информатики»;

#### **1.5. Учебно-методическое и информационное обеспечение для проведения контроля**

##### **Основные учебные издания**

1. Аблязов, Р.З. Программирование на ассемблере на платформе x86-64/Р.З. Аблязов. — 2-е изд. — Саратов: Профобразование, 2019. — 301 с. — ISBN 978-5-4488-0117-4. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/88005.html>

2. Введение в разработку приложений для ОС Android: учебное пособие / Ю. В. Березовская, О. А. Юфрякова, В. Г. Вологодина [и др.]. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 427 с. — ISBN 978-5-4497-0890-8. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/102000.html>

3. Зыков, С. В. Введение в теорию программирования. Объектно-ориентированный подход: учебное пособие для СПО / С. В. Зыков. — Саратов: Профобразование, 2021. — 187 с. — ISBN 978-5-4488-0995-8. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/102188>

4. Кариев, Ч. А. Разработка Windows-приложений на основе Visual C#: учебное пособие / Ч. А. Кариев. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 978 с. — ISBN 978-5-4497-0909-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/102057.html>

5. Котляров, В.П. Основы тестирования программного обеспечения: курс лекций / Котляров В.П. — Москва: Интуит НОУ, 2016. — 348 с. — ISBN 978-5-9556-0027-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://book.ru/book/917951>

6. Лебедева, Т. Н. Технология программирования: учебное пособие для СПО/ Т. Н. Лебедева, С. С. Юнусова. — Саратов: Профобразование, 2019. — 140 с. — ISBN 978-5-4488-0351-2. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/86081>

##### **Дополнительные учебные издания:**

7. Авдеев, В. А. Периферийные устройства: интерфейсы, схемотехника, программирование / В. А. Авдеев. — 2-е изд. — Саратов: Профобразование, 2019. — 848 с. — ISBN 978-5-4488-0053-5. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/88002.html>

8.Биллиг, В. А. Основы программирования на C#: учебное пособие / В. А. Биллиг. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 573 с. — ISBN 978-5-4497-0893-9. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/102033>

9.Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие для СПО / Т. М. Зубкова. — Саратов: Профобразование, 2019. — 468 с. — ISBN 978-5-4488-0354-3. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/86208>

10.Пирская, Л. В. Разработка мобильных приложений в среде AndroidStudio: учебное пособие / Л. В. Пирская. — Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2019. — 123 с. — ISBN 978-5-9275-3346-6. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/100196.html>

11.Семакова, А. Введение в разработку приложений для смартфонов на ОС Android: учебное пособие для СПО/ А. Семакова. — Саратов: Профобразование, 2021. — 102 с. — ISBN 978-5-4488-0994-1. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/102187>

#### **Электронные издания (электронные ресурсы)**

12.Учебники по программированию <http://programm.ws/index.php>

Таблица 1

**Распределение знаний и умений в соответствии с профессиональными компетенциями**

Профессиональные компетенции	Знания	Умения
ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	- основные этапы разработки программного обеспечения;	- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.	-способы оптимизации и приемы рефакторинга;	- создавать программу по разработанному алгоритму как отдельный модуль;
ПК1.3.Выполнять отладку программных модулей с использованием специализированных программных средств.	-основные принципы отладки и тестирования программных продуктов	- выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;
ПК1.4 Выполнять тестирование программных модулей	-основные принципы технологии структурного и объектно-ориентированного программирования;	- уметь выполнять оптимизацию и рефакторинг программного кода;
ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода.	- основные этапы разработки программного обеспечения;	- оформлять документацию на программные средства
ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ	-способы оптимизации и приемы рефакторинга;	- уметь выполнять оптимизацию и рефакторинг программного кода;

Таблица 2

**Распределение знаний и умений в соответствии с общими компетенциями**

Общие компетенции	Знания	Умения
<b>ОК 01</b> Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	-способы оптимизации и приемы рефакторинга;	- создавать программу по разработанному алгоритму как отдельный модуль;
<b>ОК 02</b> Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	- основные этапы разработки программного обеспечения;	- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
<b>ОК 03</b> Планировать и реализовывать собственное профессиональное и личностное развитие.	- основные принципы отладки и тестирования программных продуктов	-оформлять документацию на программные средства
<b>ОК 04</b> Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.	- основные принципы технологии структурного и объектно-ориентированного программирования;	- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
<b>ОК 05</b> Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	- способы оптимизации и приемы рефакторинга;	- создавать программу по разработанному алгоритму как отдельный модуль;
<b>ОК 06</b> Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих	- основные этапы разработки программного обеспечения;	- выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках

ценностей, применять стандарты антикоррупционного поведения		программирования;
ОК 07 Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях	- способы оптимизации и приемы рефакторинга;	- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
ОК 08 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	- основные принципы технологии структурного и объектно-ориентированного программирования;	- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
ОК 09 Использовать информационные технологии в профессиональной деятельности	- основные принципы отладки и тестирования программных продуктов	- уметь выполнять оптимизацию и рефакторинг программного кода;
ОК 10 Пользоваться профессиональной документацией на государственном и иностранном языках	- основные этапы разработки программного обеспечения;	- оформлять документацию на программные средства
ОК 11 Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере	- основные принципы технологии структурного и объектно-ориентированного программирования;	- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;

## **2. Контрольно-оценочные средства**

### **Теоретическое занятие 1**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: входной контроль;

Задание: Вопросы

1. Что является целью тестирования программ? Почему?
2. Перечислите известные вам виды контроля качества программного обеспечения. На каких этапах применяют каждый из них?
3. Какие подходы к тестированию вы знаете? В чем они заключаются?

### **Теоретическое занятие 2**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Почему функциональное тестирование называют «тестированием по методу черного ящика»?
2. Перечислите методы функционального тестирования и определите, в каких случаях следует использовать каждый из них.
3. Почему структурное тестирование называют «тестированием по методу белого или прозрачного ящика»?

### **Теоретическое занятие 3**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Перечислите методы структурного тестирования и определите возможности каждого из них. Какой метод структурного тестирования обеспечивает наибольшую вероятность обнаружения ошибок?
2. Используя методы обоих подходов, сформируйте пакет тестов для тестирования программы, вычисляющей действительные корни квадратного уравнения. Какие методы вы использовали и почему?

### **Теоретическое занятие 4**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Чем нисходящее тестирование отличается от восходящего?
2. Что понимают под комплексным тестированием и чем оно отличается от тестирования компонент?
3. Когда можно прекращать тестирование компонентов?

### **Теоретическое занятие 5**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Перечислите виды тестирования системы в целом.
2. В каких случаях применяют каждый из них?

### **Теоретическое занятие 6**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Что такое отладка программного средства?
2. Что такое тестирование программного средства?
3. Что такое автономная отладка программного средства?

### **Теоретическое занятие 7**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Что такое комплексная отладка программного средства?
2. Что такое ведущий отладочный модуль?
3. Что такое отладочный имитатор программного модуля?

### **Теоретическое занятие 8**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Какие методы тестирования вы знаете?
2. В чем заключаются методы «черного» и «белого» ящика?
3. На каком этапе проводится ручная отладка?
4. Опишите методы отладки



### **Теоретическое занятие 9**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Какой процесс называют отладкой? В чем его сложность?
2. Назовите основные типы ошибок. Как они проявляются при выполнении программы?

### **Теоретическое занятие 10**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Перечислите основные методы отладки.
2. В чем заключается различие между ними?
3. Возьмите любую программу, содержащую ошибки, и попробуйте найти ошибку, используя каждый из перечисленных методов.
4. Какой метод для вас проще и естественней и почему?

### **Теоретическое занятие 11**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Какие средства получения дополнительной информации об ошибках вы знаете?
2. Вспомните, какие ошибки вы искали дольше всего и почему.
3. В каких случаях дополнительная информация позволяет найти ошибку?

### **Теоретическое занятие 12**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Чем нисходящее тестирование отличается от восходящего?
2. Что понимают под комплексным тестированием и чем оно отличается от тестирования компонент?

### **Теоретическое занятие 13**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Возьмите любую программу, содержащую ошибки, и попробуйте найти ошибку, используя каждый из перечисленных методов.

#### **Теоретическое занятие 14**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Почему структурное тестирование называют «тестированием по методу белого или прозрачного ящика»?

#### **Теоретическое занятие 15**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. В чем заключается различие между ними?

2. Возьмите любую программу, содержащую ошибки, и попробуйте найти ошибку, используя каждый из перечисленных методов.

#### **Теоретическое занятие 16**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Какие методы тестирования вы знаете?

2. В чем заключаются методы «черного» и «белого» ящика?

#### **Теоретическое занятие 17**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Перечислите методы структурного тестирования и определите возможности каждого из них.

2. Какой метод структурного тестирования обеспечивает наибольшую вероятность обнаружения ошибок?

3. Используя методы обоих подходов, сформируйте пакет тестов для тестирования программы, вычисляющей действительные корни квадратного уравнения.

4. Какие методы вы использовали и почему?

### **Теоретическое занятие 18**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Что является целью тестирования программ?
2. Перечислите известные вам виды контроля качества программного обеспечения.
3. На каких этапах применяют каждый из них?

### **Теоретическое занятие 19**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Почему структурное тестирование называют «тестированием по методу белого или прозрачного ящика»?
2. Перечислите методы структурного тестирования и определите возможности каждого из них.
3. Какой метод структурного тестирования обеспечивает наибольшую вероятность обнаружения ошибок?

### **Практическая работа 1**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1. Тестирование «белым ящиком»

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

### **Практическая работа 2**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1. Тестирование «белым ящиком»

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 3**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1. Тестирование «белым ящиком»

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 4**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1. Тестирование «белым ящиком»

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 5**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1. Тестирование «белым ящиком»

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 6**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1. Тестирование «черным ящиком»

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 7**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Тестирование «черным ящиком»

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 8**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Тестирование «черным ящиком»

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 9**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Тестирование «черным ящиком»

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 10**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Тестирование «черным ящиком»

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 11**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Модульное тестирование

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 12**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Модульное тестирование

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 13**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Модульное тестирование

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 14**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Модульное тестирование

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 15**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Модульное тестирование

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 16**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Интеграционное тестирование

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 17**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Интеграционное тестирование

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 18**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: оперативный контроль;

Задание:

1.Интеграционное тестирование

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 19**

**Тема: Отладка и тестирование программного обеспечения**

**Форма контроля: оперативный контроль;**

**Задание:**

**1.Интеграционное тестирование**

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 21**

**Тема: Отладка и тестирование программного обеспечения**

**Форма контроля: оперативный контроль;**

**Задание:**

**1.Интеграционное тестирование**

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Лабораторная работа 1**

**Тема: Отладка и тестирование программного обеспечения**

**Форма контроля: входной контроль;**

**Задание:**

**1.Тестирование и отладка программного обеспечения**

**Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК. 01.02. «Методы создания и корректировки компьютерных моделей»**

**Лабораторная работа 2**

**Тема: Отладка и тестирование программного обеспечения**

**Форма контроля: входной контроль;**

**Задание:**

**1.Тестирование и отладка программного обеспечения**



**Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Лабораторная работа 3**

Тема: **Отладка и тестирование программного обеспечения**

Форма контроля: входной контроль;

Задание:

1. Тестирование и отладка программного обеспечения

**Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Теоретическое занятие 20**

Тема: **Документирование**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Средства разработки технической документации.

**Теоретическое занятие 21**

Тема: **Документирование**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Технологии разработки документов.

**Теоретическое занятие 22**

Тема: **Документирование**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Документирование программного обеспечения в соответствии с единой системой программной документации.

**Теоретическое занятие 23**

Тема: **Документирование**

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Автоматизация разработки технической документации

### **Теоретическое занятие 24**

Тема: Документирование

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Автоматизированные средства оформления документации

### **Теоретическое занятие 25**

Тема: Документирование

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Единая система программной документации

### **Теоретическое занятие 26**

Тема: Документирование

Форма контроля: Оперативный контроль;

Задание: Вопросы

1. Разработка программных документов

### **Практическая работа 22**

Тема: Документирование

Форма контроля: оперативный контроль;

Задание:

1. Построить диаграмму классов предметной области задачи.
2. Построить диаграмму вариантов использования предметной области задачи.
3. Составить отчет по практической работе.

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

### **Практическая работа 23**

Тема: Документирование

Форма контроля: оперативный контроль;

Задание:

1. Описать этапы проектирования модулей программы.

2. Составить в виде блок-схемы алгоритм решения задачи.
3. Составить отчет по практической работе.

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

#### **Практическая работа 24**

Тема: Документирование

Форма контроля: оперативный контроль;

Задание:

1. Создать интерфейс программы решения задачи с использованием любой визуальной среды программирования.
2. Отладить программу.

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

#### **Практическая работа 25**

Тема: Документирование

Форма контроля: оперативный контроль;

Задание:

1. Сформировать диаграмму вариантов использования
2. Сгенерировать набор тестов

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

#### **Практическая работа 26**

Тема: Документирование

Форма контроля: оперативный контроль;

Задание:

1. Составить в виде блок-схемы алгоритм решения задачи.
2. Создать программу решения задачи на любом алгоритмическом языке программирования.

3. Отладить программу.

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Практическая работа 27**

Тема: Документирование

Форма контроля: оперативный контроль;

Задание:

- 1.Оформить внешнюю спецификацию.
- 2.Составить в виде блок-схемы алгоритм решения задачи.
- 3.Создать программу решения задачи на любом алгоритмическом языке программирования.

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

**Самостоятельная работа 1**

Тема: Документирование

Задание:

- 1.Тестирование программного обеспечения.
- 2.Документирование.

**Порядок выполнения работы согласно методическим рекомендациям по самостоятельным работам по МДК.01.02 «Поддержка и тестирование программных модулей»**

## Приложение1

### Варианты заданий для промежуточной аттестации

1. Тестирование как часть процесса верификации программного обеспечения.
2. Виды ошибок. Методы отладки.
3. Методы тестирования.
4. Классификация тестирования по уровням.
5. Тестирование производительности
6. Регрессионное тестирование.
- 7 Тестирование «белым ящиком»
- 8 Тестирование «черным ящиком»
- 9 Модульное тестирование
- 10 Интеграционное тестирование
11. Средства разработки технической документации.
12. Технологии разработки документов.
13. Документирование программного обеспечения в соответствии с единой системой программной документации.
14. Автоматизация разработки технической документации
15. Автоматизированные средства оформления документации
16. Оформление документации на программные средства с использованием инструментальных средств

## ВХОДНОЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

### 1. Управление требованиями:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Задача выявления изначальных проблем заказчика и создание системы, удовлетворяющей этим требованиям
- B. Процесс систематического выявления, организации и документирования требований к сложной системе
- C. Выявление требований заказчика и управление ими
- D. Задача, состоящая в том, чтобы понимать проблемы заказчиков в их предметной области и на их языке и создавать системы, удовлетворяющие их потребности
- E. Процесс создания программного обеспечения и адаптация его под требования заказчика
- F. Разработка требований к программному обеспечению и создание ПО на основе этих требований
- G. Процесс, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и выполняющей проект группой по поводу меняющихся требований к системе
- H. Разработка программного обеспечения и выработка требований к изменению работы системы заказчика

### 2. К методам выявления требований относятся:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Беседы с первыми руководителями предприятия, для которого разрабатывается программное обеспечение
- B. Анализ научной и технической литературы, посвященной вопросам разработки программного обеспечения
- C. Личные встречи и беседы со всеми сотрудниками предприятия
- D. Анализ технической документации и на основе нее разработка требований к системе
- E. На начальном этапе требования не выявляются, а формируются по мере разработки программного обеспечения
- F. Интервьюирование и анкетирование, мозговой штурм и отбор идей
- G. Совещания, посвященные требованиям, создание прототипов
- H. Раскадровки, прецеденты, обыгрывание ролей

### 3. Требования к разрабатываемой системе должны включать:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Разработку программного обеспечения и выработка требований к изменению работы системы заказчика
- B. Совокупность условий, при которых предполагается эксплуатировать будущую систему (аппаратные и программные ресурсы, предоставляемые системе; внешние условия ее функционирования; состав людей и работ, имеющих к ней отношение)
- C. Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
- D. Описание выполняемых системой функций
- E. Технологию создания сложного программного обеспечения, основанную на объектном представлении кода программы
- F. Ограничения в процессе разработки (директивные сроки завершения отдельных этапов, имеющиеся ресурсы, организационные процедуры и мероприятия, обеспечивающие защиту информации)
- G. Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения
- H. Технологию разработки программного обеспечения на базе структурной схемы развития языков программирования

4. Типы средств, иллюстрирующие цели моделирования системы:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Функции, которые система должна выполнять
- B. Отношения между данными
- C. Зависящее от времени поведение системы (аспекты реального времени)
- D. Способы отладки и тестирования программного обеспечения
- E. Создание программного обеспечения на основе структурной схемы исследуемого объекта или процесса
- F. Выявление требований заказчика и управление ими
- G. Технология разработки программного обеспечения на базе структурной схемы развития языков программирования
- H. Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения

5. Преимущества объектноориентированного подхода:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Быстрота написания программного кода
- B. Статичность конфигурации системы
- C. Возможность многократного использования
- D. Низкая стоимость проекта
- E. Восприимчивость к изменениям

- Г. Отсутствие необходимости документирования
- Д. Простота реализуемых моделей
- Е. Реалистичное моделирование

6. Требования – это:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Документ, регулирующий отношения между заказчиком информационной системы и проектировщиком
- Б. Некоторое свойство программного обеспечения, необходимое пользователю для решения проблемы при достижении поставленной цели
- В. Оформленное заказчиком в виде документа задание на проектирование программного обеспечения
- Г. Возможность, которую должна обеспечивать система
- Д. Характеристика проектируемого программного обеспечения с точки зрения разработчика
- Е. Некоторое свойство программного обеспечения, которым должна обладать система или ее компонент, чтобы удовлетворить требования формальной документации
- Ж. Оформленное разработчиком в виде документа задание на проектирование программного обеспечения
- З. Характеристика проектируемого программного обеспечения с точки зрения заказчика

7. Типичная схема процесса анализа Требования включает в себя:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Идентификацию заказчика и проведение интервью с представителями заказчика
- Б. Разработку программного обеспечения в соответствии с требованиями заказчика
- В. Изложение заказчику требований к системе на основе разработанного программного обеспечения
- Г. Написание Требования в форме стандартного документа
- Д. Верификацию разработанного программного обеспечения в соответствии с требованиями заказчика
- Е. Составление плана мероприятий по анализу Требования
- Ж. Проверку Требования и согласование их с заказчиком
- З. Адаптацию разработанного программного обеспечения в соответствии с требованиями заказчика

8. В классификацию требований к программной системе входят:



**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Требования заказчика
- В. Требования, накладываемые условиями эксплуатации
- С. Функциональные требования
- Д. Требования, накладываемые аппаратными средствами
- Е. Нефункциональные требования
- Ф. Требования предметной области
- Г. Экономические требования
- Н. Требования разработчиков

9. Процесс определения и анализа требований включает в себя:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Анализ работы систем с аналогичной предметной областью
- В. Анализ предметной области, сбор и классификацию требований
- С. Проведение совместных совещаний с представителями заказчика
- Д. Разрешение противоречий и определение приоритетов
- Е. Адаптацию требований к разрабатываемому программному обеспечению
- Ф. Декомпозицию общей задачи на подзадачи
- Г. Проверку, специфицирование и документирование требований
- Н. Верификацию требований в соответствии с разработанным программным обеспечением

10. Опорные точки зрения конечных пользователей системы программного обеспечения можно трактовать как:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Источник информации о системных данных
- В. Структуру требований
- С. Источник событий
- Д. Структуру событий
- Е. Структуру представлений
- Ф. Получателей требований
- Г. Источник сценариев
- Н. Получателей системных сервисов

**Критерии оценивания:**

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69

«2» (неудовлетворительно)	Менее 50
---------------------------	----------

### Ключ ответов к тестовым заданиям

№ вопроса	Правильный ответ	Проверяемые компетенции
1	B,D,G	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
2	F,G,H	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
3	B,D,F	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
4	A,B,C	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
5	C,E,H	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
6	B,D,F	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
7	A,D,G	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
8	C,E,F	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
9	B,D,G	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
10	A,E,H	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

## РУБЕЖНЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Уровни организационного управления при планировании разработки системы:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Стратегический
- В. Тактический
- С. Оперативный
- Д. Основной
- Е. Вспомогательный
- Ф. Дополнительный
- Г. Системный
- Н. Аналитический

2. Для различных представлений проектируемой системы используют типы моделей:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Статическая модель
- В. Динамическая модель
- С. Модель классов
- Д. Модель декомпозиции
- Е. Модель размещения
- Ф. Модель состояний
- Г. Модель взаимодействия
- Н. Модель агрегации

3. Классификация бизнеспроцессов включает следующие классы процессов:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Вспомогательные бизнеспроцессы
- В. Основные бизнеспроцессы
- С. Дополнительные бизнеспроцессы
- Д. Обеспечивающие бизнеспроцессы
- Е. Обслуживающие бизнеспроцессы
- Ф. Бизнеспроцессы согласования
- Г. Бизнеспроцессы управления
- Н. Руководящие бизнеспроцессы

4. Типы Требования:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Функциональные требования
- В. Интерфейсные требования
- С. Нефункциональные требования
- Д. Программные требования
- Е. Обратные требования
- Ф. Ограниченные требования
- Г. Производительные требования
- Н. Надежность

5. Возможные способы организации Требования:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. По атрибутам, по компонентам
- В. По взаимоотношениям сущности
- С. По пакетам и по иерархии компонентов
- Д. По свойствам, по классам
- Е. По вариантам использования
- Ф. По узлам и по использованным процессам
- Г. По состояниям и по иерархии функции
- Н. По прецедентам, по кооперациям

6. К моделированию относится:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Система обозначений
- В. Система атрибутов
- С. Синтаксис языка моделирования
- Д. Система свойств
- Е. Совокупность поведения объектов
- Ф. Совокупность графических объектов
- Г. Семантика языка моделирования
- Н. Совокупность текстовых объектов

7. Классификация имитационных моделей:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Статистическая
- В. Адаптивная

- С. Статическая или динамическая
- Д. Структурная
- Е. Сетерминированная или стохастическая
- Ф. Непрерывная или дискретная
- Г. Объединенная
- Н. Декомпозиционная

8. Принципы разработки эффективного пользовательского интерфейса:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Сложность, графика
- В. Структура, простота
- С. Связь, обработка
- Д. Видимость, обратная связь
- Е. Невидимость, сложность
- Ф. Толерантность, повторное использование
- Г. Первое использование, итерация
- Н. Интеграция, повторение

9. Принципы разработки программного обеспечения:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Коллективный процесс разработки
- В. Индивидуальный процесс разработки
- С. Параллельный процесс разработки
- Д. Командный процесс разработки
- Е. Промежуточный процесс разработки
- Ф. Модель зрелости возможностей
- Г. Модель законченности возможностей
- Н. Модель готовности процессов

10. Типы интерфейсных требований:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Пользовательские требования
- В. Аппаратные требования
- С. Административные требования
- Д. Требования к производительности
- Е. Программные и коммуникационные требования
- Ф. Требования к надежности
- Г. Требования к устойчивости
- Н. Атрибуты программной системы и другие требования

11. Технология проектирования определяется как совокупность составляющих:  
Поэтапная процедура

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Пошаговая процедура
- B. Модели и правила
- C. Критерий и правила
- D. Тестирование
- E. Нотаций
- F. Прецеденты
- G. Классы

**Критерии оценивания:**

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	Менее 50

**Ключ ответов к тестовым заданиям**

№ вопроса	Правильный ответ	Проверяемые компетенции
1	A,B,C	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
2	C,F,G	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
3	B,D,G	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
4	C,E,H	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
5	D,E,G	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
6	A,C,F	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
7	C,F,E	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК

		07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
8	B,D,F	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
9	B, D,F	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
10	A,B,E	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
11	B,D,E	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

## МЕЖСЕССИОННЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Типы интерфейсных требований:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Пользовательские требования
- В. Аппаратные требования
- С. Административные требования
- Д. Требования к производительности
- Е. Программные и коммуникационные требования
- Ф. Требования к надежности
- Г. Требования к устойчивости
- Н. Атрибуты программной системы и другие требования

2. Технология проектирования определяется как совокупность составляющих:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Поэтапная процедура
- В. Пошаговая процедура
- С. Модели и правила
- Д. Критерий и правила
- Е. Тестирование
- Ф. Нотаций
- Г. Прецеденты
- Н. Классы

3. Разработка и сопровождение ИС в конкретной организации и конкретном проекте должна поддерживаться стандартами:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Стандарт организации
- В. Стандарт конкретного проекта
- С. Стандарт проектирования
- Д. Стандарт оценки
- Е. Стандарт оформления проектной документации
- Ф. Стандарт аудита
- Г. Стандарт оформления разработки
- Н. Стандарт пользовательского интерфейса

4. Результатами проектирования архитектуры являются:



**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Модель административного интерфейса
- B. Модель процессов
- C. Модель потоков
- D. Модель классов
- E. Модель данных
- F. Модель пользовательского интерфейса
- G. Модель компонентов
- H. Модель узлов

5. Какие работы включает процесс разработки программного обеспечения:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Документирование, управление конфигурацией
- B. Управление, создание инфраструктуры
- C. Структура из процессов, работ, задач
- D. Обеспечение качества, верификация
- E. Анализ требований, проектирование
- F. Программирование, сборка, тестирование
- G. Ввод в действие, приемка
- H. Совместный анализ, аудит

6. Какие технологии разработки программ используются в современном программировании:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Визуальные
- B. Событийные
- C. Структурные
- D. Объектноориентированные
- E. Модульные
- F. Текстуальные
- G. Графические
- H. Машинноориентированное

7. Объектноориентированное проектирование использует инструментальные средства:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Model mart

- B. Rational Rose
- C. Bpwin
- D. ARIS
- E. IdeflX
- F. Erwin
- G. MS Visio
- H. Jam

8. Проектирование функциональных моделей поддерживается инструментальными средствами:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Jam
- B. Model Mart
- C. MS visio
- D. ERwin
- E. Idef0
- F. Aris
- G. Rational rose
- H. BPwin

9. IEEE – это:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Коммерческая организация ученых и исследователей
- B. Просто принятое обозначение, расшифровки не имеет
- C. Обозначение всемирной компьютерной сети
- D. Всемирная некоммерческая техническая профессиональная ассоциация ученых и исследователей
- E. Такая аббревиатура нигде не используется
- F. Institute Of Electrical and Electronic Engineers, Inc
- G. Американская организация ученых экономистов
- H. Институт инженеров радиоэлектроники и электротехники

10. Ядро знаний SWEBOOK – это:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. ГОСТ на разработку программного обеспечения
- B. Нормативный документ, разработанный IEEE
- C. ГОСТ на разработку информационных систем
- D. Документ, устанавливающий правовые отношения между заказчиком и разработчиком программного обеспечения

- Е. Основополагающий научнотехнический документ, который отображает мнение специалистов в области программной инженерии
- Е. Документ, устанавливающий методику тестирования и испытания программного обеспечения
- Г. Документ, который согласуется с современными регламентированными процессами жизненного цикла ПО стандарта ISO/IEC 12207
- Н. ГОСТ на разработку и комплектацию сопровождающей документации

11. Каждая область ядра знаний SWEBOOK представляется:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Структурной схемой
- В. Общей схемой описания
- С. Диаграммой UML
- Д. Описанием и комментариями
- Е. Определением понятийного аппарата, методов и средств инженерной деятельности
- Е. Определением языка программирования
- Г. Определением инструментов поддержки инженерной деятельности
- Н. Иерархической диаграммой

12. К основным областям знаний SWEBOOK относятся:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Инженерия требований, проектирование ПО
- В. Анализ деятельности системы
- С. Управление проектами
- Д. Конструирование ПО
- Е. Управление персоналом
- Е. Тестирование ПО, сопровождение ПО
- Г. Управление конфигурацией
- Н. Инженерия качества программных средств

13. К организационным областям знаний SWEBOOK относятся:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Инженерия требований
- В. Управление конфигурацией, управление проектами
- С. Конструирование ПО
- Д. Процесс инженерии программных средств, методы и средства программной инженерии

Е. Проектирование ПО  
Ф. Сопровождение ПО  
Тестирование ПО  
Инженерия качества программных средств

14. В рамках Rational Unified Process (RUP) набор действий по разработке программ включает этапы:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Создание структурных схем
- В. Определения входных, выходных данных
- С. Согласование стоимости проекта
- Д. Согласования требований с заказчиком
- Е. Создания бизнесмоделей
- Ф. Определение требований
- Г. Проектирование, программирование
- Н. Тестирование, внедрение

15. Этапы разработки консалтинговых проектов включают в себя:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Анализ первичных требований и планирование работ
- В. Снятие программного продукта с эксплуатации
- С. Декомпозицию задачи на подзадачи
- Д. Разработку спецификации и документации
- Е. Проведение обследования деятельности предприятия
- Ф. Тестирование и сопровождение программного обеспечения
- Г. Построение моделей деятельности предприятия (модели AS – IS – “как есть” и модели TO – BE – “как должно быть”)
- Н. Разработку программного обеспечения

16. Концепции, лежащие в основе модульного программирования:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Объем реализации и время исполнения (реакции)
- В. Мера автоматизма в работе реализации и инструментах разработки
- С. Визуальность и тестируемость разработки
- Д. Функциональная декомпозиция, пространственная и временная группировка информации (модульность)
- Е. Упрощение связей
- Ф. Комментируемость функций и данных

Г. Надежность, устойчивость  
Н. Безопасность

17. Инструмент разработки программ выбирается на основе:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Визуальности, набора реализуемых технологий
- В. Мощности множества элементов разработки
- С. Системного подхода к анализу, проектированию и реализации ПО
- Д. Функциональной декомпозиции, пространственной и временной группировка информации (модульность)
- Е. Упрощения связей, комментируемости функций и данных
- Г. Объема реализации и времени исполнения (реакции), надежности, устойчивости, безопасности
- Г. Меры автоматизма в работе реализации и инструментах разработки
- Н. Визуальности и тестируемости разработки

**Критерии оценивания:**

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	Менее 50

**Ключ ответов к тестовым заданиям**

№ вопроса	Правильный ответ	Проверяемые компетенции
1	А,В,Е	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
2	В,Д,Е	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
3	С,Е,Н	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
4	В, Е, F	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
5	Е, G, F	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4.

		ПК 1.5.
6	A,B,D	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
7	B, D, G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
8	B, D, H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
9	D,F,H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
10	B,E,G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
11	A,D,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
12	A,D,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
13	B,D,H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
14	G,F,H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
15	A,E,G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
16	D,E,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
17	G,F,H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.

## ИТОГОВЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Управление требованиями:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Задача выявления изначальных проблем заказчика и создание системы, удовлетворяющей этим требованиям
- B. Процесс систематического выявления, организации и документирования требований к сложной системе
- C. Выявление требований заказчика и управление ими
- D. Задача, состоящая в том, чтобы понимать проблемы заказчиков в их предметной области и на их языке и создавать системы, удовлетворяющие их потребности
- E. Процесс создания программного обеспечения и адаптация его под требования заказчика
- F. Разработка требований к программному обеспечению и создание ПО на основе этих требований
- G. Процесс, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и выполняющей проект группой по поводу меняющихся требований к системе
- H. Разработка программного обеспечения и выработка требований к изменению работы системы заказчика

2. К методам выявления требований относятся:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Беседы с первыми руководителями предприятия, для которого разрабатывается программное обеспечение
- B. Анализ научной и технической литературы, посвященной вопросам разработки программного обеспечения
- C. Личные встречи и беседы со всеми сотрудниками предприятия
- D. Анализ технической документации и на основе нее разработка требований к системе
- E. На начальном этапе требования не выявляются, а формируются по мере разработки программного обеспечения
- F. Интервьюирование и анкетирование, мозговой штурм и отбор идей
- G. Совещания, посвященные требованиям, создание прототипов
- H. Раскадровки, прецеденты, обыгрывание ролей

3. Требования к разрабатываемой системе должны включать:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Разработку программного обеспечения и выработка требований к изменению работы системы заказчика
- B. Совокупность условий, при которых предполагается эксплуатировать будущую систему (аппаратные и программные ресурсы, предоставляемые системе; внешние условия ее функционирования; состав людей и работ, имеющих к ней отношение)
- C. Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
- D. Описание выполняемых системой функций
- E. Технологию создания сложного программного обеспечения, основанную на объектном представлении кода программы
- F. Ограничения в процессе разработки (директивные сроки завершения отдельных этапов, имеющиеся ресурсы, организационные процедуры и мероприятия, обеспечивающие защиту информации)
- G. Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения
- H. Технологию разработки программного обеспечения на базе структурной схемы развития языков программирования

4. Типы средств, иллюстрирующие цели моделирования системы:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Функции, которые система должна выполнять
- B. Отношения между данными
- C. Зависящее от времени поведение системы (аспекты реального времени)
- D. Способы отладки и тестирования программного обеспечения
- E. Создание программного обеспечения на основе структурной схемы исследуемого объекта или процесса
- F. Выявление требований заказчика и управление ими
- G. Технология разработки программного обеспечения на базе структурной схемы развития языков программирования
- H. Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения

5. Преимущества объектноориентированного подхода:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Быстрота написания программного кода
- B. Статичность конфигурации системы
- C. Возможность многократного использования
- D. Низкая стоимость проекта
- E. Восприимчивость к изменениям



- Г. Отсутствие необходимости документирования
- Д. Простота реализуемых моделей
- Е. Реалистичное моделирование

6. Требования – это:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Документ, регулирующий отношения между заказчиком информационной системы и проектировщиком
- Б. Некоторое свойство программного обеспечения, необходимое пользователю для решения проблемы при достижении поставленной цели
- В. Оформленное заказчиком в виде документа задание на проектирование программного обеспечения
- Г. Возможность, которую должна обеспечивать система
- Д. Характеристика проектируемого программного обеспечения с точки зрения разработчика
- Е. Некоторое свойство программного обеспечения, которым должна обладать система или ее компонент, чтобы удовлетворить требования формальной документации
- Ж. Оформленное разработчиком в виде документа задание на проектирование программного обеспечения
- З. Характеристика проектируемого программного обеспечения с точки зрения заказчика

7. Типичная схема процесса анализа Требования включает в себя:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Идентификацию заказчика и проведение интервью с представителями заказчика
- Б. Разработку программного обеспечения в соответствии с требованиями заказчика
- В. Изложение заказчику требований к системе на основе разработанного программного обеспечения
- Г. Написание Требования в форме стандартного документа
- Д. Верификацию разработанного программного обеспечения в соответствии с требованиями заказчика
- Е. Составление плана мероприятий по анализу Требования
- Ж. Проверку Требования и согласование их с заказчиком
- З. Адаптацию разработанного программного обеспечения в соответствии с требованиями заказчика

8. В классификацию требований к программной системе входят:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Требования заказчика
- В. Требования, накладываемые условиями эксплуатации
- С. Функциональные требования
- Д. Требования, накладываемые аппаратными средствами
- Е. Нефункциональные требования
- Ф. Требования предметной области
- Г. Экономические требования
- Н. Требования разработчиков

9. Процесс определения и анализа требований включает в себя:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Анализ работы систем с аналогичной предметной областью
- В. Анализ предметной области, сбор и классификацию требований
- С. Проведение совместных совещаний с представителями заказчика
- Д. Разрешение противоречий и определение приоритетов
- Е. Адаптацию требований к разрабатываемому программному обеспечению
- Ф. Декомпозицию общей задачи на подзадачи
- Г. Проверку, специфицирование и документирование требований
- Н. Верификацию требований в соответствии с разработанным программным обеспечением

10. Опорные точки зрения конечных пользователей системы программного обеспечения можно трактовать как:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Источник информации о системных данных
- В. Структуру требований
- С. Источник событий
- Д. Структуру событий
- Е. Структуру представлений
- Ф. Получателей требований
- Г. Источник сценариев
- Н. Получателей системных сервисов

11. При аттестации требований выполняются следующие типы проверок документации требований:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Проверка на совместимость
- B. Проверка на управляемость
- C. Проверка правильности требований
- D. Проверка на непротиворечивость
- E. Проверка на соответствие
- F. Проверка на обратимость
- G. Проверка на полноту и на выполнимость
- H. Проверка на заменяемость

12. К методам аттестации требований относится:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Тестирование
- B. Обзор требований
- C. Верификация
- D. Сравнительный анализ
- E. Прототипирование
- F. Генерация случайных данных
- G. Генерация тестовых сценариев
- H. Декомпозиция

13. Уровни организационного управления при планировании разработки системы:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Стратегический
- B. Tактический
- C. Оперативный
- D. Основной
- E. Вспомогательный
- F. Дополнительный
- G. Системный
- H. Аналитический

14. Для различных представлений проектируемой системы используют типы моделей:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Статическая модель
- B. Динамическая модель
- C. Модель классов
- D. Модель декомпозиции

- Е. Модель размещения
- Ф. Модель состояний
- Г. Модель взаимодействия
- Н. Модель агрегации

15. Классификация бизнеспроцессов включает следующие классы процессов:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Вспомогательные бизнеспроцессы
- В. Основные бизнеспроцессы
- С. Дополнительные бизнеспроцессы
- Д. Обеспечивающие бизнеспроцессы
- Е. Обслуживающие бизнеспроцессы
- Ф. Бизнеспроцессы согласования
- Г. Бизнеспроцессы управления
- Н. Руководящие бизнеспроцессы

16. Типы Дтребований:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Функциональные требования
- В. Интерфейсные требования
- С. Нефункциональные требования
- Д. Программные требования
- Е. Обратные требования
- Ф. Ограниченные требования
- Г. Производительные требования
- Н. Надежность

17. Возможные способы организации Дтребований:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. По атрибутам, по компонентам
- В. По взаимоотношениям сущности
- С. По пакетам и по иерархии компонентов
- Д. По свойствам, по классам
- Е. По вариантам использования
- Ф. По узлам и по использованным процессам
- Г. По состояниям и по иерархии функции
- Н. По прецедентам, по кооперациям

18. К моделированию относится:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Система обозначений
- В. Система атрибутов
- С. Синтаксис языка моделирования
- Д. Система свойств
- Е. Совокупность поведения объектов
- Ф. Совокупность графических объектов
- Г. Семантика языка моделирования
- Н. Совокупность текстовых объектов

19. Классификация имитационных моделей:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Статистическая
- В. Адаптивная
- С. Статическая или динамическая
- Д. Структурная
- Е. Сетерминированная или стохастическая
- Ф. Непрерывная или дискретная
- Г. Объединенная
- Н. Декомпозиционная

20. Принципы разработки эффективного пользовательского интерфейса:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Сложность, графика
- В. Структура, простота
- С. Связь, обработка
- Д. Видимость, обратная связь
- Е. Невидимость, сложность
- Ф. Толерантность, повторное использование
- Г. Первое использование, итерация
- Н. Интеграция, повторение

21. Принципы разработки программного обеспечения:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Коллективный процесс разработки

- В. Индивидуальный процесс разработки
- С. Параллельный процесс разработки
- Д. Командный процесс разработки
- Е. Промежуточный процесс разработки
- Ф. Модель зрелости возможностей
- Г. Модель законченности возможностей
- Н. Модель готовности процессов

22. Типы интерфейсных требований:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Пользовательские требования
- В. Аппаратные требования
- С. Административные требования
- Д. Требования к производительности
- Е. Программные и коммуникационные требования
- Ф. Требования к надежности
- Г. Требования к устойчивости
- Н. Атрибуты программной системы и другие требования

23. Технология проектирования определяется как совокупность составляющих:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Поэтапная процедура
- В. Пошаговая процедура
- С. Модели и правила
- Д. Критерий и правила
- Е. Тестирование
- Ф. Нотаций
- Г. Прецеденты
- Н. Классы

24. Разработка и сопровождение ИС в конкретной организации и конкретном проекте должна поддерживаться стандартами:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Стандарт организации
- В. Стандарт конкретного проекта
- С. Стандарт проектирования
- Д. Стандарт оценки
- Е. Стандарт оформления проектной документации
- Ф. Стандарт аудита

- G. Стандарт оформления разработки
- H. Стандарт пользовательского интерфейса

25. Результатами проектирования архитектуры являются:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Модель административного интерфейса
- B. Модель процессов
- C. Модель потоков
- D. Модель классов
- E. Модель данных
- F. Модель пользовательского интерфейса
- G. Модель компонентов
- H. Модель узлов

26. Какие работы включает процесс разработки программного обеспечения:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Документирование, управление конфигурацией
- B. Управление, создание инфраструктуры
- C. Структура из процессов, работ, задач
- D. Обеспечение качества, верификация
- E. Анализ требований, проектирование
- F. Программирование, сборка, тестирование
- G. Ввод в действие, приемка
- H. Совместный анализ, аудит

27. Какие технологии разработки программ используются в современном программировании:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Визуальные
- B. Событийные
- C. Структурные
- D. Объектноориентированные
- E. Модульные
- F. Текстуальные
- G. Графические
- H. Машинноориентированное

28. Объектноориентированное проектирование использует инструментальные средства:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Model mart
- B. Rational Rose
- C. Bpwin
- D. ARIS
- E. IdeflX
- F. Erwin
- G. MS Visio
- H. Jam

29. Проектирование функциональных моделей поддерживается инструментальными средствами:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Jam
- B. Model Mart
- C. MS visio
- D. ERwin
- E. Idef0
- F. Aris
- G. Rational rose
- H. BPwin

30. IEEE – это:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Коммерческая организация ученых и исследователей
- B. Просто принятое обозначение, расшифровки не имеет
- C. Обозначение всемирной компьютерной сети
- D. Всемирная некоммерческая техническая профессиональная ассоциация ученых и исследователей
- E. Такая аббревиатура нигде не используется
- F. Institute Of Electrical and Electronic Engineers, Inc
- G. Американская организация ученыхэкономистов
- H. Институт инженеров радиоэлектроники и электротехники

31. Ядро знаний SWEBOK – это:

- A. ГОСТ на разработку программного обеспечения
- B. Нормативный документ, разработанный IEEE
- C. ГОСТ на разработку информационных систем



- D. Документ, устанавливающий правовые отношения между заказчиком и разработчиком программного обеспечения
- E. основополагающий науднотехнический документ, который отображает мнение специалистов в области программной инженерии
- F. Документ, устанавливающий методику тестирования и испытания программного обеспечения
- G. Документ, который согласуется с современными регламентированными процессами жизненного цикла ПО стандарта ISO/IEC 12207
- H. ГОСТ на разработку и комплектацию сопровождающей документации

32. Каждая область ядра знаний SWEBOOK представляется:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Общей схемой описания
- B. Диаграммой UML
- C. Описанием и комментариями
- D. Определением понятийного аппарата, методов и средств инженерной деятельности
- E. Определением языка программирования
- F. Определением инструментов поддержки инженерной деятельности
- G. Иерархической диаграммой

33. К основным областям знаний SWEBOOK относятся:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Инженерия требований, проектирование ПО
- B. Анализ деятельности системы
- C. Управление проектами
- D. Конструирование ПО
- E. Управление персоналом
- F. Тестирование ПО, сопровождение ПО
- G. Управление конфигурацией
- H. Инженерия качества программных средств

34. К организационным областям знаний SWEBOOK относятся:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- A. Инженерия требований
- B. Управление конфигурацией, управление проектами
- C. Конструирование ПО
- D. Процесс инженерии программных средств, методы и средства программной инженерии

- Е. Проектирование ПО
- Ф. Сопровождение ПО
- Г. Тестирование ПО
- Н. Инженерия качества программных средств

35. В рамках Rational Unified Process (RUP) набор действий по разработке программ включает этапы:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Создание структурных схем
- В. Определения входных, выходных данных
- С. Согласование стоимости проекта
- Д. Согласования требований с заказчиком
- Е. Создания бизнесмоделей
- Ф. Определение требований
- Г. Проектирование, программирование
- Н. Тестирование, внедрение

36. Этапы разработки консалтинговых проектов включают в себя:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Анализ первичных требований и планирование работ
- В. Снятие программного продукта с эксплуатации
- С. Декомпозицию задачи на подзадачи
- Д. Разработку спецификации и документации
- Е. Проведение обследования деятельности предприятия
- Ф. Тестирование и сопровождение программного обеспечения
- Г. Построение моделей деятельности предприятия (модели AS – IS – “как есть” и модели TO – BE – “как должно быть”)
- Н. Разработку программного обеспечения

37. Концепции, лежащие в основе модульного программирования:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Объем реализации и время исполнения (реакции)
- В. Мера автоматизма в работе реализации и инструментах разработки
- С. Визуальность и тестируемость разработки
- Д. Функциональная декомпозиция, пространственная и временная группировка информации (модульность)
- Е. Упрощение связей
- Ф. Комментируемость функций и данных
- Г. Надежность, устойчивость

## Н. Безопасность

38. Инструмент разработки программ выбирается на основе:

**Проверяемые компетенции:** ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07  
ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

- А. Визуальности, набора реализуемых технологий
- В. Мощности множества элементов разработки
- С. Системного подхода к анализу, проектированию и реализации ПО
- Д. Функциональной декомпозиции, пространственной и временной группировка информации (модульность)
- Е. Упрощения связей, комментируемости функций и данных
- Г. Объема реализации и времени исполнения (реакции), надежности, устойчивости, безопасности
- Г. Меры автоматизма в работе реализации и инструментах разработки
- Н. Визуальности и тестируемости разработки

### Критерии оценивания:

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	Менее 50

### Ключ ответов к тестовым заданиям

№ вопроса	Правильный ответ	Проверяемые компетенции
1	В,D,G	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
2	F,G,H	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
3	В,D,F	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
4	А,В,С	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
5	С,Е,Н	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

6	B,D,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
7	A,D,G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
8	C,E,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
9	B,D,G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
10	A,E,H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
11	C,D,G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
12	B,E,G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
13	A,B,C	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
14	C,F,G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
15	B,D,G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
16	C,E,H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
17	D,E,G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
18	A,C,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
19	C,F,E	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
20	B,D,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4.

		ПК 1.5.
21	B, D,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
22	A,B,E	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
23	B,D,E	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
24	C,E,H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
25	B, E,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
26	E,G,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
27	A,B,D	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
28	B, D, G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
29	B, D, H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
30	D,F,H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
31	B,E,G	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
32	A,D,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
33	A,D,F	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
34	B,D,H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK 07 OK 08 OK 09. OK 10. OK 11. ПК 1.3. ПК1.4. ПК 1.5.
35	G,F,H	OK 01 OK 02 OK 03 OK 04. OK 05. OK 06. OK

		07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
36	A,E,G	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
37	D,E,F	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.
38	G,F,H	ОК 01 ОК 02 ОК 03 ОК 04. ОК 05. ОК 06. ОК 07 ОК 08 ОК 09. ОК 10. ОК 11. ПК 1.3. ПК1.4. ПК 1.5.

# **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ОЦЕНКЕ КАЧЕСТВА ОСВОЕНИЯ СТУДЕНТАМИ ПРОГРАММ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА**

по междисциплинарному курсу  
МДК.01.03. «Разработка мобильных приложений»

## **Пояснительная записка**

Фонд оценочных средств разработан в соответствии с рабочей программой профессионального модуля, требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 «Информационные системы и программирование», утверждённого приказом Министерства образования и науки РФ от 09.12.2016 г., № 1547 (ред. от 17.12.2020), ФГОС среднего общего образования и примерной основной образовательной программой.

### **1. Паспорт фонда оценочных средств для проведения текущего контроля успеваемости**

#### **1.1. Цели и задачи контроля**

Целью текущего контроля успеваемости обучающихся является обеспечение систематического контроля и оценки уровня освоения предметных результатов, уровня сформированности общих компетенций МДК. 01.03 «Разработка мобильных приложений».

Главной задачей текущего контроля успеваемости является повышение мотивации обучающихся к регулярной учебной и самостоятельной работе, закрепление, углубление знаний, закрепление и совершенствование умений, обеспечение соответствия результатов обучения задачам будущей профессиональной деятельности посредством внедрения эффективной системы оценки в образовательный процесс.

#### **Общие компетенции, включающие в себя способность:**

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

ОК 09. Использовать информационные технологии в профессиональной деятельности

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках

ОК 11. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере

**Профессиональные компетенции, включающие в себя способность:**

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ

**Знать и уметь:**

**знать:**

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- способы оптимизации и приемы рефакторинга;
- основные принципы отладки и тестирования программных продуктов

**уметь:**

- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
- создавать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;
- уметь выполнять оптимизацию и рефакторинг программного кода;
- оформлять документацию на программные средства

## **1.2. Структура фонда оценочных средств**

Фонд оценочных средств для проведения текущего контроля включает в себя комплекты контрольно-оценочных средств, предназначенные для проведения текущего контроля в виде:

- оперативного контроля;
- рубежного контроля.

Оперативный контроль проводится в форме:

- опрос (устный, письменный);
- тестирование;
- выполнение практической работы;



Рубежный контроль проводится в форме:

- опрос (устный, письменный);
- тестирование;
- выполнение практической работы;

Фонд оценочных средств также, включает в себя комплект контрольно-оценочных средств для проведения межсессионной аттестации. Межсессионная аттестация проводится в форме тестирования.

### **1.3. Система оценивания результатов выполнения заданий**

Результаты выполнения заданий оцениваются в соответствии с разработанными критериями оценки.

Используются следующие шкалы для оценивания результатов обучения:

- пяти бальная шкала оценки;
- сто бальная шкала оценки.

Перевод сто бальной шкалы учета результатов в пяти бальную оценочную шкалу:

<b>Оценка</b>	<b>Количество баллов, набранных за выполнение теоретического и практического задания</b>
Оценка 5 «отлично»	90-100
Оценка 4 «хорошо»	76-89
Оценка 3 «удовлетворительно»	50-75
Оценка 2 «неудовлетворительно»	≤ 49

Перевод пяти бальной шкалы учета результатов в пяти бальную оценочную шкалу:

<b>Оценка</b>	<b>Количество баллов, набранных за выполнение теоретического и практического задания, средний балл по итогам аттестации</b>
Оценка 5 «отлично»	4,6-5
Оценка 4 «хорошо»	3,6-4,5
Оценка 3 «удовлетворительно»	3-3,5
Оценка 2 «неудовлетворительно»	≤ 2,9

### **1.4. Учебно-методическое и информационное обеспечение для проведения контроля**

**Основные учебные издания:**

1. Введение в разработку приложений для ОС Android: учебное пособие / Ю. В. Березовская, О. А. Юфрякова, В. Г. Вологодина [и др.]. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 427 с. — ISBN 978-5-4497-0890-8. — Текст: электронный //

Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/102000.html>

**Дополнительные учебные издания:**

2. Пирская, Л. В. Разработка мобильных приложений в среде Android Studio: учебное пособие / Л. В. Пирская. — Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2019. — 123 с. — ISBN 978-5-9275-3346-6. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/100196.html>

3. Семакова, А. Введение в разработку приложений для смартфонов на ОС Android: учебное пособие для СПО / А. Семакова. — Саратов: Профобразование, 2021. — 102 с. — ISBN 978-5-4488-0994-1. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/102187>

**Электронные издания (электронные ресурсы)**

4. Учебники по программированию <http://programm.ws/index.php>

**Электронно-библиотечная система:**

5. ЭБС «elibrary», ООО «РУНЭБ»

6. ЭБС «IPRbooks», ООО «Ай Пи Ар Медиа»

7. ЭБС «Лань», ООО «Издательство Лань»

8. ЭБС «PROФобразование»

9. ЭБС «Book.ru»

Таблица 1

**Распределение знаний и умений в соответствии с профессиональными компетенциями**

Профессиональные компетенции	Знания	Умения
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.	Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Знание API современных мобильных операционных систем	Создавать программу по разработанному алгоритму как отдельный модуль. Оформлять документацию на программные средства. Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ
ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ	Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования	Осуществлять разработку кода программного модуля на современных языках программирования. Оформлять документацию на программные средства

## Распределение знаний и умений в соответствии с общими компетенциями

Общие компетенции	Знания	Умения
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности	распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы; владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)
ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.	номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации	определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие	содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования	определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.	психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности	организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	особенности социального и культурного контекста; правила оформления документов и построения устных сообщений	грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения	сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности	описывать значимость своей специальности
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.	правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения	соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения	использовать физкультурно-оздоровительную деятельность для укрепления здоровья, достижения жизненных и профессиональных целей; применять рациональные приемы двигательных функций в профессиональной деятельности; пользоваться средствами профилактики перенапряжения характерными для данной специальности
ОК 09. Использовать информационные технологии в профессиональной деятельности.	современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности	применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение
ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.	правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности	понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и

		планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы
ОК 11. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере	основы предпринимательской деятельности; основы финансовой грамотности; правила разработки бизнес-планов; порядок выстраивания презентации; кредитные банковские продукты	выявлять достоинства и недостатки коммерческой идеи; презентовать идеи открытия собственного дела в профессиональной деятельности; оформлять бизнес-план; рассчитывать размеры выплат по процентным ставкам кредитования; определять инвестиционную привлекательность коммерческих идей в рамках профессиональной деятельности; презентовать бизнес-идею; определять источники финансирования

## 2. Контрольно-оценочные средства

Теоретическое занятие 1. Основные платформы мобильных приложений,  
сравнительная характеристика

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Назовите основные платформы мобильных приложений.
2. Каково устройство платформы Android?
3. Что представляет собой Android SDK?

Теоретическое занятие 2. Основные платформы мобильных приложений,  
сравнительная характеристика

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Назовите основные средства разработки под Android.
2. Перечислите достоинства и недостатки эмуляторов Android.

Теоретическое занятие 3. Основные платформы мобильных приложений, сравнительная характеристика

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Выясните объем продаж мобильных устройств с ОС Android.
2. Какая версия платформы наиболее популярна в настоящее время?

Теоретическое занятие 4. Основные платформы мобильных приложений, сравнительная характеристика

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое Android SDK?
2. С какой целью был создан Open Handset Alliance?
3. Что такое операционная система Android?
4. Какой движок баз данных используется в ОС Android?

Теоретическое занятие 5. Основные платформы мобильных приложений, сравнительная характеристика

**Форма контроля:** оперативный контроль – опрос письменный.

**Задание:** ответить на вопросы.

**Условия выполнения задания:**

- 1) обучающиеся письменно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Платформа и операционная система. Есть различия или сходства между двумя терминами? Ответ объяснить.
2. Напишите основные мобильные платформы. Минимум 4
3. Напишите основные мобильные операционные системы. Минимум 4
4. Какая мобильная платформа лучше всего подходит для первой разработки? Ответ объяснить.
5. Напишите особенности и различия мобильных платформ:  
А) IOS  
Б) Android

Теоретическое занятие 6. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое нативные приложения?
2. Для чего создаются нативные приложения? Приведите примеры.
3. В чем плюсы и минусы нативных приложений?
4. Где применяются нативные приложения?

Теоретическое занятие 7. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое Web-приложения?
2. Для чего создаются Web-приложения? Приведите примеры.
3. В чем плюсы и минусы Web-приложений?
4. Где применяются Web-приложения?

Теоретическое занятие 8. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое гибридные приложения?
2. Для чего создаются нативные приложения? Приведите примеры.
3. В чем плюсы и минусы гибридных приложений?
4. Где применяются гибридные приложения?

Теоретическое занятие 9. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое кроссплатформенные приложения?
2. Для чего создаются кроссплатформенные приложения? Приведите примеры.
3. В чем плюсы и минусы кроссплатформенных приложений?
4. Где применяются кроссплатформенные приложения?

Теоретическое занятие 10. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы письменно.

**Условия выполнения задания:**

- 1) обучающиеся письменно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое нативное приложение? Привести примеры
2. Что такое гибридное приложение? Привести примеры
3. Что такое Web-приложение? Привести примеры
4. Что такое кроссплатформенные приложения? Привести примеры
5. Примером каких приложений являются Вконтакте, Instagram, Twitter?

Теоретическое занятие 11. Основные языки для разработки мобильных приложений (Java, Objective-C и др.)

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Назовите языки для разработки мобильных приложений?
2. Какие языки применяются для разработки мобильных приложений для платформы iOS?
3. Назовите главные языки для разработки приложений под Android.

Теоретическое занятие 12. Основные языки для разработки мобильных приложений (Java, Objective-C и др.)

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**



- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1.Какое количество и какие именно языки программирования можно использовать для разработки простого мобильного приложения?

*Ответ: JS, HTML, CSS*

- 2.Особенности и недостатки ЯП Java (минимум по 2 критерия)
- 3.Особенности и недостатки ЯП JavaScript (минимум по 2 критерия)

Теоретическое занятие 13. Основные языки для разработки мобильных приложений (Java, Objective-C и др.)

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы письменно.

**Условия выполнения задания:**

- 1) обучающиеся письменно отвечают на вопросы;
- 2) время, отводимое на опрос – 15 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

- 1.Напишите языки разработки мобильных приложений для Android. Минимум 3.
- 2.Напишите языки разработки мобильных приложений для IOS. Минимум 3.
- 3.Особенности и недостатки ЯП Swift (минимум по 2 критерия)
- 4.Особенности и недостатки ЯП C# (минимум по 2 критерия)
- 5.Особенности и недостатки ЯП Objective-C (минимум по 2 критерия)

Теоретическое занятие 14-15. Основные языки для разработки мобильных приложений (Java, Objective-C и др.)

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

- 1.Для чего предназначен язык Java?
- 2.Охарактеризуйте язык программирования Kotlin
3. Охарактеризуйте язык программирования Swift

Теоретическое занятие 16. Инструменты разработки мобильных приложений (JDK/ Android Studio/ WebView/ Phonegap и др.)

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;

- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что лучше: использовать конструктор или самим разработать приложение?  
Ответ объяснить.
2. Что такое инструменты разработки мобильных приложений?
3. Что такое фреймворк?
4. Для чего нужны инструменты разработки мобильных приложений?

Теоретическое занятие 17. Инструменты разработки мобильных приложений (JDK/ Android Studio/ WebView/ Phonegap и др.)

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы письменно.

**Условия выполнения задания:**

- 1) обучающиеся письменно отвечают на вопросы;
- 2) время, отводимое на опрос – 20 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Напишите 5 инструментов разработки мобильных приложений и по 2 особенности каждого для Android:
2. Напишите инструмент(ы) для искусственного интеллекта и их особенности:
3. Напишите рекламные инструмент(ы) и их особенности:
4. Напишите инструмент(ы) для push-уведомлений и их особенности:
5. Напишите инструмент(ы) для мобильных платежей и их особенности:

Теоретическое занятие 18. Инструменты разработки мобильных приложений (JDK/ Android Studio/ WebView/ Phonegap и др.)

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое Android NDK?
2. Чем являются Eclipse и IDEA?
3. Что относится к преимуществам среды разработки Intel XDK?

Теоретическое занятие 19. Инструменты разработки мобильных приложений (JDK/ Android Studio/ WebView/ Phonegap и др.)

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;

- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

- 1.Опишите среду разработки Android Studio
- 2.Для чего предназначен AVD Manager?
- 3.Что представляет из себя Unity3D?

Теоретическое занятие 20. Инструменты разработки мобильных приложений (JDK/ Android Studio/ WebView/ Phonegap и др.)

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. С какой целью инструмент Intel\* Hardware Accelerated Execution Manager (Intel\* HAXM) используется в среде разработки Intel\* Beacon Mountain?
2. С какой целью инструмент Intel\* Graphics Performance Analyzers (Intel\* GPA) System Analyzer используется в среде разработки Intel\* Beacon Mountain?
3. Для чего предназначен эмулятор среды разработки Intel XDK?

Лабораторная работа 1. Установка инструментария и настройка среды для разработки мобильных приложений

**Форма контроля:** Рубежный контроль

**Задание:**

1. Название и цель работы.
2. Произвести установку Android Studio
- 3.Ответить на контрольные вопросы:
  - 3.1.Какие минимальные требования необходимы для установки Android Studio?
  - 3.2.Опишите процесс установки и настройки Android Studio.

Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК. 01.03 «Разработка мобильных приложений»

Лабораторная работа 2. Установка инструментария и настройка среды для разработки мобильных приложений

**Форма контроля:** Рубежный контроль

**Задание:**

1. Название и цель работы.
2. Произвести установку ADT Bundle
- 3.Ответить на контрольные вопросы:
  - 3.1. Какие минимальные требования необходимы для установки ADT Bundle?
  - 3.2.Опишите процесс установки и настройки ADT Bundle.

Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК. 01.03 «Разработка мобильных приложений»

Лабораторная работа 3. Установка среды разработки мобильных приложений с применением виртуальной машины

**Форма контроля:** Рубежный контроль

**Задание:**

1. Название и цель работы.
2. Создать новое виртуальное устройство

Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК. 01.03 «Разработка мобильных приложений»

Лабораторная работа 4. Установка среды разработки мобильных приложений с применением виртуальной машины

**Форма контроля:** Рубежный контроль

**Задание:**

1. Название и цель работы.
2. Запуск эмулятора

Порядок выполнения работы согласно методическим рекомендациям по лабораторным работам по МДК. 01.03 «Разработка мобильных приложений»

Самостоятельная работа 1. Среда разработки мобильных приложений

**Форма контроля:** Рубежный контроль

**Задание:**

1. Подготовить презентацию по данной теме.

Порядок выполнения работы согласно методическим рекомендациям по самостоятельным работам по МДК. 01.03 «Разработка мобильных приложений».

Теоретическое занятие 21. Инструментарий среды разработки мобильных приложений

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Перечислите инструментарий среды разработки мобильных приложений
2. Чем являются Eclipse и IDEA?
3. Что такое эмулятор

Теоретическое занятие 22. Инструментарий среды разработки мобильных приложений

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое среда Android Studio?
2. Для чего предназначена среда Android Studio?
3. Для чего предназначена среда Apache Cordova?

Практическая работа 1. Создание эмуляторов и подключение устройств

**Форма контроля:** Рубежный контроль

**Задание:**

1. Подготовить эмулятор Android Studio

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 2. Создание эмуляторов и подключение устройств

**Форма контроля:** Рубежный контроль

**Задание:**

1. Настроить и запустить эмулятор Android Studio

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 3. Настройка режима терминала

**Форма контроля:** Рубежный контроль

**Задание:**

1. Изучить команды эмулятора терминала в системе Android

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Теоретическое занятие 23. Структура типичного мобильного приложения

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое менеджер пакетов Android? Какие задачи он решает?
2. Какова структура автоматически создаваемого проекта приложения для Android? Какие компоненты создаются и в каких каталогах они размещаются?
3. Что такое файл манифеста? Какова его структура? Какие основные элементы могут встречаться в файле манифеста и для чего они нужны?
4. Что такое ant? Как он используется для сборки приложений? Какие цели содержатся в автоматически сгенерированном файле сборки?

#### Теоретическое занятие 24. Структура типичного мобильного приложения

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Какие основные блоки входят в структуру приложения Android?
2. Что собой представляет блок Activity?
3. Для чего предназначен класс Intent?
4. За что отвечает блок Service?
5. Для чего предназначен класс Content Provider?

#### Теоретическое занятие 25. Структура типичного мобильного приложения

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое фоновое приложение?
2. Какая папка в структуре Android-приложения содержит файлы с исходным кодом на языке Java?
3. Что такое виджет?
4. Какая графическая библиотека входит в набор библиотек ОС Android?

#### Теоретическое занятие 26. Структура типичного мобильного приложения

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что необходимо для запуска приложений, разработанных в Android IDE?
2. Какая активность появляется первой при загрузке приложения?
3. Что необходимо при проектировании окон мобильных приложений?
4. Какое визуальное свойство является самым главным в определении сущности объекта?

#### Практическая работа 4. Создание нового проекта

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать новый проект в среде Android Studio
2. Разработать мобильное приложение по индивидуальным заданиям.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 5. Создание нового проекта

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать графический интерфейс, созданного в Практической работе № 4 мобильного приложения
2. Разработать графический интерфейс мобильных приложений, разработанных по индивидуальным заданиям

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 6. Создание нового проекта

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать мобильное приложение, в котором на одной странице приложения мы будем вводить некоторые данные и по нажатию на кнопку будет происходить переход к другой странице приложения, которая будет отображать ранее введенные данные.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 7. Создание нового проекта

**Форма контроля:** Рубежный контроль

**Задание:**

1. В Практической работе № 6 определили MessageActivity, которая получает извне некоторые данные. Теперь определим в MainActivity код, который будет запускать MessageActivity и передавать ей некоторые данные.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 8. Изучение и комментирование кода

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создайте новый проект MetroPicker. Проект отредактировать, добавив кнопку

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 9. Изучение и комментирование кода

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать проект в Android Studio изучить файл AndroidManifest.xml.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Теоретическое занятие 27. Элементы управления и контейнеры

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое дизайн мобильного приложения?
2. Что необходимо учитывать при разработке дизайна приложения?
3. Какое средство организации интерфейса помогает визуально уравновесить элементы?
4. Что необходимо учитывать при разработке справочной системы приложения?

Теоретическое занятие 28. Элементы управления и контейнеры

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что относится к элементам ввода?
2. Для чего предназначается панель Navigation Bar?
3. Стоит ли для приложения создавать полосы прокрутки?

Теоретическое занятие 29. Элементы управления и контейнеры

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что учитывается при проектировании окон приложений?
2. Что такое комбо-элементы?
3. Какие элементы управления применяются для действий по настройке?

Теоретическое занятие 30. Элементы управления и контейнеры



**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. В каких случаях следует использовать всплывающие подсказки?
2. Что такое ProgressDialog?
3. Что такое DialogFragment?

Практическая работа 10. Изменение элементов дизайна

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать интерфейс приложения, которое ищет в сети Интернет изображения по запросу пользователя, позволяет оценивать их, скачивать, и посещать интернет-страницы сайтов, на которых было найдено изображение.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 11. Изменение элементов дизайна

**Форма контроля:** Рубежный контроль

**Задание:**

1. Добавление текстового поля в мобильное приложение, созданное в Практической работе № 10

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 12. Изменение элементов дизайна

**Форма контроля:** Рубежный контроль

**Задание:**

1. Добавить кнопки в созданное в Практической работе № 11 приложение.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 13. Обработка событий: цветовая индикация

**Форма контроля:** Рубежный контроль

**Задание:**

1. Смена фона приложения, разработанного в Практической работе № 12.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 14. Обработка событий: цветовая индикация

**Форма контроля:** Рубежный контроль

**Задание:**

1. Добавить область просмотра изображений в приложение, созданное в Практической работе № 13.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 15. Обработка событий: цветовая индикация

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать мобильное приложение с добавлением разных элементов. В качестве фона выбрать изображение. При нажатии на кнопку, должно происходить смена фона приложения.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 16. Обработка событий: подсказки

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать кнопки оценивания в приложении, созданном в Практической работе № 14.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 17. Обработка событий: подсказки

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать приложение в Android Studio, добавить всплывающие подсказки.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 18. Обработка событий: подсказки

**Форма контроля:** Рубежный контроль

**Задание:**

1. Добавить систему текстовых подсказок в мобильное приложение, созданное в Практической работе № 15.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Теоретическое занятие 31. Работа со списками

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

1) обучающиеся устно отвечают на вопросы;

- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Для чего используют списки в мобильных приложениях?
2. Какие виды списков можно создавать?
3. Что такое комбо-списки?

Теоретическое занятие 32. Работа со списками

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое раскрывающиеся списки?
2. Что такое нумерованные списки?
3. В чем заключаются плюсы и минусы использования списков?

Теоретическое занятие 33. Работа со списками

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Как осуществляются переключения между активностями?
2. Какая библиотека предназначена для упрощения загрузки изображений?
3. С какой целью используются списки?
4. Какие виды списков создаются в приложениях?

Практическая работа 19. Подготовка стандартных модулей

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать многоэкранное приложение со списком

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 20. Подготовка стандартных модулей

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создание диалогового окна

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 21. Обработка событий: переключение между экранами

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создание приложения со слайдингом из шаблона

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 22. Обработка событий: переключение между экранами

**Форма контроля:** Рубежный контроль

**Задание:**

1. Подумайте над собственным приложением, сочетающим различные возможности проектирования многооконных приложений. Создайте прототип этого приложения и настройте его пользовательский интерфейс.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 23. Передача данных между модулями

**Форма контроля:** Рубежный контроль

**Задание:**

1. Разработать приложения, получающего координаты устройства и отслеживающего их изменение.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Теоретическое занятие 34. Способы хранения данных

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Перечислите способы постоянного хранения данных на платформе Android. Объясните, в каких случаях разумно применять каждый из них.
2. Что такое механизм настроек? Для чего он предназначен? Как его применять?
3. Перечислите основные классы Android, предназначенные для работы с базой данных SQLite. На примерах объясните, как их применять.

Теоретическое занятие 35. Способы хранения данных

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое жизненный цикл базы данных? Какие средства платформы Android позволяют управлять этим жизненным циклом?
2. Назовите методы класса SQLiteDatabase, предназначенные для работы с данными. На примерах объясните, как их можно использовать.
3. В чём отличие между методами query() и rawQuery() класса SQLiteDatabase? В каких случаях применяется каждый из них?

Теоретическое занятие 36. Способы хранения данных

**Форма контроля:** оперативный контроль.

**Задание:** ответить на вопросы устно (фронтальный опрос).

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

1. Что такое курсор набора данных?
2. Для чего он предназначен?
3. На примерах объясните, как использовать курсоры.

Практическая работа 24. Тестирование и оптимизация мобильного приложения

**Форма контроля:** Рубежный контроль

**Задание:**

1. Создать тестирующее приложение

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 25. Тестирование и оптимизация мобильного приложения

**Форма контроля:** Рубежный контроль

**Задание:**

1. Произведите тестирование каждого уровня всех ранее созданных мобильных приложений. При обнаружении ошибок – оптимизируйте их.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

Практическая работа 26. Тестирование и оптимизация мобильного приложения

**Форма контроля:** Рубежный контроль

**Задание:**

1. Произведите тестирование каждым типом всех ранее созданных мобильных приложений. При обнаружении ошибок – оптимизируйте их.

Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК. 01.03 «Разработка мобильных приложений»

### 3. Критерии оценки

#### 3.1. Инвариантные критерии оценки

#### Критерии оценки устных (письменных) ответов на теоретические вопросы

Критерии оценки		Оценка
1	Демонстрирует глубокое, полное знание и понимание программного материала. Последовательно, самостоятельно раскрывает основное содержание вопроса. Выводы аргументированы, основаны на самостоятельно выполненном анализе, обобщении данных. Четко и верно даны определения понятий и научных терминов. Дает верные, самостоятельные ответы на вопросы.	5 (отлично)
2	Демонстрирует недостаточно глубокое, полное знание и понимание программного материала. Недостаточно последовательно, но самостоятельно раскрывает основное содержание вопроса. Выводы основаны на самостоятельно выполненном анализе, обобщении данных, но в отдельных случаях недостаточно аргументированы. Недостаточно четко и верно даны определения понятий и научных терминов. При ответе на вопросы допускает несущественные ошибки, которые может исправить самостоятельно.	4 (хорошо)
3	Демонстрирует в отдельных вопросах, неглубокое владение знаниями программного материала. Излагает программный материал фрагментарно, не всегда последовательно. Допущены ошибки и неточности в использовании научной терминологии. При ответе на вопросы допускает неточности.	3 (удовлетворительно)
4	Студент демонстрирует незнание и непонимание программного материала. Основное содержание учебного материала не раскрыто; допущены грубые ошибки в определении понятий, при использовании терминологии. Затрудняется отвечать на вопросы, при ответе допускает серьезные ошибки.	2 (неудовлетворительно)

### Критерии оценки результатов выполнения тестового задания

Оценка	Количество правильных ответов на вопросы в % соотношении от общего числа вопросов
Оценка 5 «отлично»	90-100%
Оценка 4 «хорошо»	76-89%
Оценка 3 «удовлетворительно»	50-75%
Оценка 2 «неудовлетворительно»	≤ 49%

### Критерии оценки результатов промежуточной аттестации:

Критерии оценки		Оценка
<b>1</b>	обучающийся показал полный объем знаний по вопросу, владеет культурой общения, навыками научного изложения материала, устанавливает связь между теоретическими знаниями и способами практической деятельности; ясно, точно и логично отвечает на заданные вопросы.	5 (отлично)
<b>2</b>	обучающийся логично и научно изложил материал, но недостаточно полно определяет практическую значимость теоретических знаний; не высказывает своей точки зрения по данному вопросу, не смог дать достаточно полного ответа на поставленные вопросы	4 (хорошо)
<b>3</b>	обучающийся при раскрытии вопроса допустил содержательные ошибки, испытывает затруднения при ответе на большинство вопросов.	3 (удовлетворительно)
<b>4</b>	обучающийся показал слабые теоретические и практические знания, допустил грубые ошибки при раскрытии вопроса, не смог ответить на заданные вопросы	2 (неудовлетворительно)

## РУБЕЖНЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения называется:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) Android SDK
- б) JDK
- в) плагин ADT
- г) Android NDK

2. Какой движок баз данных используется в ОС Android?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) InnoDB
- б) DBM
- в) MyISAM
- г) SQLite

3. Эти приложения можно сравнить с облачным хранилищем в сравнении с данными, которые хранятся на жестком диске компьютера

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) нативные
- б) гибридные
- в) Web-приложения
- г) кроссплатформенные

4. Для создания этих приложений используются популярные фреймворки, как PhoneGap, Xamarin, Titanium, Ionic и т.д.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) нативные
- б) гибридные
- в) Web-приложения
- г) кроссплатформенные

5. Какой язык разметки используется для описания иерархии компонентов графического пользовательского интерфейса Android-приложения?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) html
- б) xml
- в) gml



г) xhtml

6. Язык для разработки Web-приложений для мобильных устройств?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

а) Kotlin

б) Java

в) HTML5

г) C++

7. Инструмент разработки, позволяющий адаптировать код C/C++ для работы на Android, это -

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

а) Google Android SDK (ADT Bundle)

б) Marmalade SDK

в) Intel\* Software Manager

г) Android NDK

8. Чем являются Eclipse и IDEA?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

а) API (интерфейс прикладного программирования)

б) IDE (интегрированная среда разработки)

в) SDK (набор средств разработки)

г) ADT (инструменты разработки под Android)

9. Язык программирования мобильных приложений под платформу iOS

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

а) Swift

б) Java

в) Kotlin

г) C++

10. Не является средством разработки под Android:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

а) ProGuard

б) Device Monitor

в) AVD Manager

г) SDK Manager

**Критерии оценивания:**

<b>Отметка</b>	<b>Объем выполнения работы в %</b>
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	Менее 50

**Ключ ответов к тестовым заданиям**

<b>№ вопроса</b>	<b>Правильный ответ</b>	<b>Проверяемые компетенции</b>
1	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
2	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
3	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
4	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
5	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
6	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
7	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
8	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
9	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
10	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

## МЕЖСЕССИОННЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. С какой целью был создан Open Handset Alliance?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) писать историю развития ОС Android
- б) продавать смартфоны под управлением Android
- в) рекламировать смартфоны под управлением Android
- г) разрабатывать открытые стандарты для мобильных устройств

2. Intel XDK поддерживает разработку под:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) JavaFX Mobile
- б) Apple iOS, BlackBerry OS
- в) MtkOS, Symbian OS, Microsoft Windows 8
- г) Android, Apple iOS, Microsoft Windows 8, Tizen

3. Профессиональное решение для обеспечения высокой скорости разработки и позволяющие создавать идеальные решения для бизнеса.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) нативные
- б) гибридные
- в) Web-приложения
- г) кроссплатформенные

4. Примером каких приложений являются Вконтакте, Instagram, Twitter?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) нативные
- б) гибридные
- в) Web-приложения
- г) кроссплатформенные

5. Это объектно-ориентированный язык программирования

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) Kotlin
- б) Java
- в) Objective-C
- г) C++

6. Язык программирования для мобильных устройств разработала компания Apple

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) Kotlin
- б) Java
- в) Objective-C
- г) C++

7. К преимуществам среды разработки Intel XDK можно отнести:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) легкость разработки кроссплатформенных приложений
- б) наличие облачного хранилища для разработанных приложений
- в) наличие собственного эффективного эмулятора
- г) все варианты ответа верны

8.С какой целью инструмент Intel\* Hardware Accelerated Execution Manager (Intel\* HAXM) используется в среде разработки Intel\* Beacon Mountain?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) для ускорения работы эмулятора в среде разработки
- б) для эффективного распараллеливания C++ мобильных приложений
- в) для оптимизированной обработки данных и изображений
- г) для оптимизации загрузки системы при использовании процедур OpenGL

9.Эмулятор среды разработки Intel XDK:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) не требует больших системных ресурсов
- б) позволяет проверить работу приложений, использующих акселерометр
- в) позволяет моделировать исполнение приложения на ряде устройств
- г) все варианты ответа верны

10. IDE для работы с платформой Android, позволяющая разрабатывать мобильные приложения на различных языках программирования, таких как Java, C++ и Kotlin.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- а) Android Studio
- б) Apache Cordova
- в) Xamarin

11. Фреймворк для кроссплатформенной разработки мобильных приложений для платформ iOS, Android и Windows Phone.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- a) Android Studio
- б) Apache Cordova
- в) Xamarin

12. Инструмент разработки, позволяющий адаптировать код C/C++ для работы на Android, это –

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- a) Intel\* Software Manager
- б) Android NDK
- в) Google Android SDK (ADT Bundle)
- г) Marmalade SDK

13. Фоновые приложения ...

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- a) после настройки не предполагают взаимодействия с пользователем, большую часть времени находятся и работают в скрытом состоянии
- б) небольшие приложения, отображаемые в виде графического объекта на рабочем столе
- в) выполняют свои функции и когда видимы на экране, и когда скрыты другими приложениями
- г) большую часть времени работают в фоновом режиме, однако допускают взаимодействие с пользователем и после настройки

14. Приложение какого вида имеет смысл использовать для отображения динамической информации, такой как заряд батареи, прогноз погоды, дата и время?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- a) фоновое приложение
- б) приложение переднего плана
- в) виджет
- г) смешанное приложение

15. При загрузке приложения первой появляется активность,

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- a) чье описание первым в лексикографическом порядке
- б) чье описание находится первым в манифесте
- в) чье описание находится последним в манифесте
- г) правильный вариант ответа отсутствует

**Критерии оценивания:**

<b>Отметка</b>	<b>Объем выполнения работы в %</b>
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	Менее 50

**Ключ ответов к тестовым заданиям**

<b>№ вопроса</b>	<b>Правильный ответ</b>	<b>Проверяемые компетенции</b>
1	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
2	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
3	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
4	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
5	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
6	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
7	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
8	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
9	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
10	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
11	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
12	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
13	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
14	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
15	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

## ИТОГОВЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения называется:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) Android SDK
- Б) JDK
- В) плагин ADT
- Г) Android NDK

2. С какой целью был создан Open Handset Alliance?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) писать историю развития ОС Android
- Б) продавать смартфоны под управлением Android
- В) рекламировать смартфоны под управлением Android
- Г) разрабатывать открытые стандарты для мобильных устройств

3. Мобильное приложение, которое создается для определенной платформы и непосредственно устанавливается на устройство пользователя

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) нативные
- Б) гибридные
- В) Web-приложения
- Г) кроссплатформенные

4. Для создания этих приложений используются популярные фреймворки, как PhoneGap, Xamarin, Titanium, Ionic и т.д.

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) нативные
- Б) гибридные
- В) Web-приложения
- Г) кроссплатформенные

5. Какой язык разметки используется для описания иерархии компонентов графического пользовательского интерфейса Android-приложения?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) html
- Б) xml
- В) gml

Г) xhtml

6. Чем являются Eclipse и IDEA?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

А) API (интерфейс прикладного программирования)

Б) IDE (интегрированная среда разработки)

В) SDK (набор средств разработки)

Г) ADT (инструменты разработки под Android)

7.С какой целью инструмент Intel® Graphics Performance Analyzers (Intel® GPA) System Analyzer используется в среде разработки Intel® Beacon Mountain?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

А) позволить разработчикам эффективно распараллелить C++ мобильные приложения

Б) позволить разработчикам оптимизировать загрузку системы при использовании процедур OpenGL

В) для ускорения работы эмулятора в среде разработки

Г) для оптимизированной обработки данных и изображений

8. Intel XDK поддерживает разработку под:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

А) Mobile

Б) Apple iOS, BlackBerry OS

В) Android, Apple iOS, Microsoft Windows 8, Tizen

Г) MtkOS, Symbian OS, Microsoft Windows 8

9. Тестирование на реальных устройствах приложений, разработанных в Intel XDK

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

А) невозможно

Б) можно проводить, установив на устройство специальную программу или собрав проект и установив результат сборки на устройство

В) можно проводить, установив на устройство специальную программу

Г) можно проводить, собрав проект и установив результат сборки на устройство

10. Набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения называется

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6



- А) Android SDK
- Б) JDK
- В) Android NDK
- Г) плагин ADT

11. Фоновые приложения ...

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) после настройки не предполагают взаимодействия с пользователем, большую часть времени находятся и работают в скрытом состоянии
- Б) небольшие приложения, отображаемые в виде графического объекта на рабочем столе
- В) выполняют свои функции и когда видимы на экране, и когда скрыты другими приложениями
- Г) большую часть времени работают в фоновом режиме, однако допускают взаимодействие с пользователем и после настройки

12. Какая графическая библиотека входит в набор библиотек ОС Android?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) OpenCV
- Б) DirectX
- В) Open GL
- Г) OpenCL

13. При загрузке приложения первой появляется активность,

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) чье описание первым в лексикографическом порядке
- Б) чье описание находится первым в манифесте
- В) чье описание находится последним в манифесте
- Г) правильный вариант ответа отсутствует

14. При проектировании окон приложения необходимо:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) интегрировать в рабочее пространство инструменты, используемые наиболее часто
- Б) для каждой функции делать отдельное диалоговое окно
- В) позволять диалоговому окну оперировать только с одной функцией
- Г) все вышеперечисленное

15. Какое визуальное свойство является самым главным в определении сущности объекта?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) размер
- Б) яркость
- В) форма
- Г) цвет

16. Элементы управления "Радиокнопки":

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) должны иметь квадратную форму
- Б) должны иметь круглую форму
- В) не должны иметь круглую форму
- Г) могут иметь любую форму

17. Переключения между активностями осуществляются

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) только при помощи кнопок
- Б) только с использованием сенсорного экрана смартфона
- В) только при помощи кнопок и других элементов управления
- Г) все три варианта возможны

18. При создании справочной системы необходимо:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) использовать всплывающие подсказки
- Б) создать руководство по "быстрому старту"
- В) учитывать, что она не должна становиться костылем для продукта
- Г) все вышеперечисленное

19. Нижняя панель (Navigation Bar) предназначена

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) для вывода времени
- Б) для вывода уровня заряда батареи
- В) для вывода уровня сигнала сотовой сети
- Г) для навигации на тех устройствах, которые не имеют аппаратных навигационных клавиш

20. При проектировании окон приложения необходимо:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) интегрировать в рабочее пространство инструменты, используемые наиболее часто

- Б) для каждой функции делать отдельное диалоговое окно
- В) все вышеперечисленное
- Г) позволять диалоговому окну оперировать только с одной функцией

21. Какие элементы управления применяются для действий по настройке?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) элементы ввода
- Б) элементы отображения
- В) командные элементы управления
- Г) элементы выбора

22. Всплывающие подсказки стоит использовать, когда

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) сообщение является важным и требует немедленного прочтения и ответа
- Б) сообщение является важным, однако требует немедленного прочтения, но не ответа
- В) сообщение является важным, однако не требует немедленного прочтения и ответа
- Г) сообщение не требует ответа пользователя, но важно для продолжения его работы

23. ProgressDialog это:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) диалоговое окно, которое может содержать заголовок, до трех кнопок, список выбираемых значений или настраиваемое содержимое
- Б) контейнер для создания собственных диалоговых окон
- В) диалоговое окно, содержащее линейку процесса выполнения какого-то действия
- Г) диалоговое окно с предопределенным интерфейсом, позволяющее выбрать дату или время

24. Эмулятор среды разработки Intel XDK:

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

- А) не требует больших системных ресурсов
- Б) позволяет проверить работу приложений, использующих акселерометр
- В) позволяет моделировать исполнение приложения на ряде устройств
- Г) все варианты ответа верны

25. Приложение какого вида имеет смысл использовать для отображения динамической информации, такой как заряд батареи, прогноз погоды, дата и время?

**Проверяемые компетенции:** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

А) фоновое приложение

Б) приложение переднего плана

В) виджет

Г) смешанное приложение

**Критерии оценивания:**

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	Менее 50

**Ключ ответов к тестовым заданиям**

№ вопроса	Правильный ответ	Проверяемые компетенции
1	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
2	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
3	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
4	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
5	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
6	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
7	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
8	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
9	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
10	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
11	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
12	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
13	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
14	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК

		07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
15	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
16	Б	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
17	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
18	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
19	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
20	А	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
21	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
22	Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
23	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
24	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6
25	В	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ОК 11, ПК 1.2, ПК 1.6

**Вопросы для экзамена по МДК.01.03 «Разработка мобильных приложений»**

**Теоретические вопросы:**

1. Основные платформы мобильных приложений, сравнительная характеристика
2. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения
3. Язык для разработки мобильных приложений Java
4. Язык для разработки мобильных приложений Objective-C
5. Инструменты разработки мобильных приложений (JDK/ Android Studio)
6. Инструменты разработки мобильных приложений (WebView/ Phonegap)
7. Инструментарий среды разработки мобильных приложений
8. Структура типичного мобильного приложения
9. Элементы управления и контейнеры
10. Работа со списками
11. Способы хранения данных

**Практические задания:**

1. Установка инструментария и настройка среды для разработки мобильных приложений
2. Установка среды разработки мобильных приложений с применением виртуальной машины
3. Создание эмуляторов и подключение устройств
4. Настройка режима терминала
5. Создание нового проекта
6. Изменение элементов дизайна
7. Обработка событий: цветовая индикация
8. Обработка событий: подсказки
9. Подготовка стандартных модулей
10. Обработка событий: переключение между экранами
11. Тестирование и оптимизация мобильного приложения

# **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ОЦЕНКЕ КАЧЕСТВА ОСВОЕНИЯ СТУДЕНТАМИ ПРОГРАММ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА**

по междисциплинарному курсу  
МДК.01.04. «Системное программирование»

## **Пояснительная записка**

Фонд оценочных средств разработан в соответствии с рабочей программой профессионального модуля, требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 «Информационные системы и программирование», утверждённого приказом Министерства образования и науки РФ от 09.12.2016 г., № 1547 (ред. от 01.09.2022), ФГОС среднего общего образования и примерной основной образовательной программой.

### **1. Паспорт фонда оценочных средств для проведения контроля успеваемости**

#### **1.1 Цели и задачи контроля**

Целью текущего контроля успеваемости обучающихся является обеспечение систематического контроля и оценки уровня освоения знаний и умений, уровня сформированности общих и профессиональных компетенций **МДК 01.04 Системное программирование.**

Главной задачей текущего контроля успеваемости является повышение мотивации обучающихся к регулярной учебной и самостоятельной работе, закрепление, углубление знаний, закрепление и совершенствование умений, обеспечение соответствия результатов обучения задачам будущей профессиональной деятельности посредством внедрения эффективной системы оценки в образовательный процесс.

#### **Общие компетенции, включающие в себя способность:**

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной

сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях.

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

ОК 10. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере

### **Профессиональные компетенции, соответствующие видам деятельности:**

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

### **Знания и умения:**

#### **знать:**

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- способы оптимизации и приемы рефакторинга;
- основные принципы отладки и тестирования программных продуктов.

#### **уметь:**

- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;
- создавать программу по разработанному алгоритму как отдельный модуль;



- выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования;
- уметь выполнять оптимизацию и рефакторинг программного кода;
- оформлять документацию на программные средства.

## 1.2. Структура фонда оценочных средств

Фонд оценочных средств для проведения текущего контроля включает в себя комплекты контрольно-оценочных средств, предназначенные для проведения текущего контроля в виде:

- оперативного контроля;
- рубежного контроля.

Оперативный контроль проводится в форме:

- опрос (устный);
- тестирование;
- выполнение письменной работы (графическая работа);
- выполнение практической работы.

Рубежный контроль проводится в форме: опрос (устный);

- тестирование;
- выполнение практической работы;
- защита портфолио;

Фонд оценочных средств также, включает в себя комплект контрольно-оценочных средств для проведения промежуточной аттестации.

## 1.3. Система оценивания результатов выполнения заданий

Результаты выполнения заданий оцениваются в соответствии с разработанными критериями оценки.

Используются следующие шкалы для оценивания результатов обучения: пятибальная шкала оценки; сто бальная шкала оценки.

Перевод сто бальной шкалы учета результатов в пяти бальную оценочную шкалу:

Оценка	Количество баллов, набранных за выполнение теоретического и практического задания
Оценка 5 «отлично»	90-100
Оценка 4 «хорошо»	76-89
Оценка 3 «удовлетворительно»	50-75
Оценка 2 «неудовлетворительно»	≤ 49

Перевод пяти бальной шкалы учета результатов в пяти бальную оценочную шкалу:

Оценка	Количество баллов, набранных за выполнение теоретического и практического задания, средний балл по итогах аттестации
Оценка 5 «отлично»	4,6-5
Оценка 4 «хорошо»	3,6-4,5
Оценка 3 «удовлетворительно»	3-3,5
Оценка 2 «неудовлетворительно»	≤ 2,9

#### **1.4. Материально-техническое обеспечение для проведения контроля**

Контроль проводится в учебном кабинете/лаборатория «Программного обеспечения и сопровождения компьютерных систем».

Оборудование лаборатории и рабочих мест лаборатории:

- Автоматизированные рабочие места на 12-15 обучающихся (процессор не ниже Core i3, оперативная память объемом не менее 4 Гб) или аналоги;
- Автоматизированное рабочее место преподавателя (процессор не ниже Core i3, оперативная память объемом не менее 4 Гб) или аналоги;
- Проектор и экран;
- Маркерная доска;

Лицензионное программное обеспечение общего и профессионального назначения.

#### **1.5 Учебно-методическое и информационное обеспечение для проведения контроля**

##### **Основные учебные издания:**

1. Аблязов, Р.З. Программирование на ассемблере на платформе x86-64/Р.З. Аблязов. — 2-е изд. — Саратов: Профобразование, 2019. — 301 с. — ISBN 978-5-4488-0117-4. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/88005.html>
2. Введение в разработку приложений для ОС Android: учебное пособие / Ю. В. Березовская, О. А. Юфрякова, В. Г. Вологодина [и др.]. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 427 с. — ISBN 978-5-4497-0890-8. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/102000.html>
3. Зыков, С. В. Введение в теорию программирования. Объектно-ориентированный подход: учебное пособие для СПО / С. В. Зыков. — Саратов: Профобразование, 2021. — 187 с. — ISBN 978-5-4488-0995-8. — Текст:

- электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/102188>
- 4.Кариев, Ч. А. Разработка Windows-приложений на основе Visual C#: учебное пособие / Ч. А. Кариев. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 978 с. — ISBN 978-5-4497-0909-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/102057.html>
- 5.Котляров, В.П. Основы тестирования программного обеспечения: курс лекций / Котляров В.П. — Москва: Интуит НОУ, 2016. — 348 с. — ISBN 978-5-9556-0027-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://book.ru/book/917951>
6. Лебедева, Т. Н. Технология программирования: учебное пособие для СПО/ Т. Н. Лебедева, С. С. Юнусова. — Саратов: Профобразование, 2019. — 140 с. — ISBN 978-5-4488-0351-2. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/86081>

#### **Дополнительные учебные издания:**

7. Авдеев, В. А. Периферийные устройства: интерфейсы, схемотехника, программирование / В. А. Авдеев. — 2-е изд. — Саратов: Профобразование, 2019. — 848 с. — ISBN 978-5-4488-0053-5. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/88002.html>
8. Биллиг, В. А. Основы программирования на C#: учебное пособие / В. А. Биллиг. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 573 с. — ISBN 978-5-4497-0893-9. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/102033>
- 9.Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие для СПО / Т. М. Зубкова. — Саратов: Профобразование, 2019. — 468 с. — ISBN 978-5-4488-0354-3. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/86208>
- 10.Пирская, Л. В. Разработка мобильных приложений в среде Android Studio: учебное пособие / Л. В. Пирская. — Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2019. — 123 с. — ISBN 978-5-9275-3346-6. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/100196.html>
11. Семакова, А. Введение в разработку приложений для смартфонов на ОС Android: учебное пособие для СПО/ А. Семакова. — Саратов: Профобразование, 2021. — 102 с. — ISBN 978-5-4488-0994-1. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/102187>

### **Электронные издания (электронные ресурсы)**

12. Учебники по программированию <http://programm.ws/index.php>
13. ЭБС - <https://www.iprbookshop.ru>.
14. ЭБС - <https://book.ru>.
15. ЭБС - <https://profspo.ru>.

Таблица 1

**Распределение знаний и умений в соответствии с профессиональными компетенциями**

<b>Профессиональные компетенции</b>	<b>Знания</b>	<b>Умения</b>
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.	<p>Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования.</p> <p><i>Дополнительно для квалификаций "Программист":</i> Знание API современных мобильных операционных систем.</p>	<p>Создавать программу по разработанному алгоритму как отдельный модуль. Оформлять документацию на программные средства.</p> <p><i>Дополнительно для квалификаций "Программист":</i> Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.</p> <p><i>Дополнительно для квалификации "Специалист по тестированию в области информационных технологий":</i> Осуществлять разработку модулей для различных видов тестирования.</p>

<p>ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.</p>	<p>Основные принципы отладки и тестирования программных продуктов. Инструментарий отладки программных продуктов.</p>	<p>Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства.</p> <p><i>Дополнительно для квалификаций "Программист" и "Специалист по тестированию в области информационных технологий":</i> Применять инструментальные средства отладки программного обеспечения.</p>
---	--	--

## 2. Контрольно-оценочные средства

### Теоретическое занятие 1

#### **Тема: Подсистема управления ресурсами**

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

#### **Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

#### **Вопросы:**

- 1) Понятие ресурса?
- 2) Виды ресурсов?
- 3) Подсистема управления памятью?

### Теоретическое занятие 2

#### **Тема: Управление процессами**

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

#### **Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

#### **Вопросы:**

- 1) Мультипрограммирование?
- 2) Фазы управления процессами?
- 3) Создание процессов?
- 4) Модель управления процессами с двумя состояниями?

### Теоретическое занятие 3

#### **Тема: Управление потоками**

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

#### **Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

#### **Вопросы:**

- 1) Структуры данных ОС, связанные с потоком
- 2) Планирование потока. Создание потока. Управление потоками
- 3) Уничтожение потока
- 4) Управление потоками

#### Теоретическое занятие 4

##### **Тема: Параллельная обработка потоков**

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

##### **Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

##### **Вопросы:**

- 1) Краткая история появления параллелизма в архитектуре ЭВМ?
- 2) Реализация параллельного потока?
- 3) Менеджер параллельных потоков?

#### Теоретическое занятие 5

##### **Тема: Создание процессов и потоков**

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

##### **Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

##### **Вопросы:**

- 1) Модели процессов и потоков?
- 2) Поддержка множества потоков внутри одного процесса?
- 3) Параллельная обработка потоков?

#### Теоретическое занятие 6

##### **Тема: Обмен данными между процессами. Передача сообщений**

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

##### **Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

##### **Вопросы:**

- 1) Способы передачи данных между процессами?
- 2) Канал передачи данных?
- 3) Функции передачи сообщений?

#### Теоретическое занятие 7

##### **Тема: Анонимные и именованные каналы**



**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

- 1) Анонимные каналы?
- 2) Именованные каналы
- 3) Использование именованных каналов

#### Теоретическое занятие 8

**Тема:** Сетевое программирование

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

- 1) Сокеты?
- 2) Правила использования номеров портов?
- 3) Простейшие клиент и сервер?

#### Теоретическое занятие 9

**Тема:** Динамически подключаемые библиотеки DLL

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

- 1) Назначение библиотеки DLL?
- 2) Преимущества и недостатки библиотеки DLL?
- 3) Работа с библиотеками динамической компоновки (DLL)

#### Теоретическое занятие 10

**Тема:** Сервисы

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

- 1) Сервис-ориентированное программирование
- 2) Средства вызова служб?
- 3) Реализация сервиса: составной сервис?

Теоретическое занятие 11

**Тема: Виртуальная память. Выделение памяти процессам**

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

- 1) Способы выделения виртуальной памяти?
- 2) Реализация виртуальной памяти?
- 3) Алгоритм выделения памяти? Элементы адресного пространства?

Теоретическое занятие 12

**Тема: Работа с буфером экрана**

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль

**Условия выполнения задания:**

- 1) обучающиеся устно отвечают на вопросы;
- 2) время, отводимое на опрос – 10 мин;
- 3) максимальный балл за задание – 5 баллов

**Вопросы:**

- 1) Стандартные библиотеки.?
- 2) Ассемблерные вставки?
- 3) Работа с буфером экрана, буфером клавиатуры, памятью, системной информацией, файловой системой?

## Практическая работа 1-8

### Тема: Использование потоков

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль.

#### **Задание:**

#### **Задание 1**

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading;
6
7  namespace ConsoleApplication1
8  {
9      public delegate int BinaryOp(int data, int time);
10
11      class Program
12      {
13          static void Main()
14          {
15              Console.WriteLine("Синхронный вызов метода: \n");
16              // Синхронный вызов метода
17              DelegateThread(1, 5000);
18
19              Console.WriteLine("\nАсинхронный вызов метода с двумя потоками: \n");
20              // Асинхронный вызов метода с применением делегата
21              BinaryOp bo = DelegateThread;
22
23              IAsyncResult ar = bo.BeginInvoke(1, 5000, null, null);
24              while (!ar.IsCompleted)
25              {
26                  // Выполнение операций в главном потоке
27                  Console.Write(".");
28                  Thread.Sleep(50);
29              }
30              int result = bo.EndInvoke(ar);
31              Console.WriteLine("Result: " + result);
32
33              Console.ReadLine();
34          }
35
36          static int DelegateThread(int data, int time)
37          {
38              Console.WriteLine("DelegateThread запущен");
39              // Делаем задержку, для эмуляции длительной операции
40              Thread.Sleep(time);
41              Console.WriteLine("DelegateThread завершен");
42              return ++data;
43          }
44      }
45  }
```

После запуска этого приложения можно увидеть, что главный поток и поток делегата выполняются параллельно, а после завершения работы потока делегата главный поток прекращает проход по циклу:

```
file:///C:/Documents and Settings/Admin/Local Settings/Application Data/Temporary Projec...
Синхронный вызов метода:
DelegateThread запущен
DelegateThread завершен

Асинхронный вызов метода с двумя потоками:
.DelegateThread запущен
.....DelegateThread завершен
Result: 2
-
```

## Задание 2

### *Получение статистики о текущем потоке*

Вспомните, что точка входа исполняемой сборки (т.е. метод Main()) запускается в первичном потоке выполнения. Чтобы проиллюстрировать базовое применение типа Thread, предположим, что имеется новое консольное приложение. Как известно, статическое свойство Thread.CurrentThread извлекает объект Thread, представляющий текущий выполняющийся поток. После получения текущего потока можно вывести разнообразную статистику о нем:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading;
6
7 namespace ConsoleApplication1
8 {
9     class Program
10     {
11         static void Main()
12         {
13             Console.Title = "Информация о главном потоке программы";
14
15             Thread thread = Thread.CurrentThread;
16             thread.Name = "MyThread";
17             Console.WriteLine(@"Имя домена приложения: {0}
18 ID контекста: {1}
19 Имя потока: {2}
20 Запущен ли поток? {3}
21 Приоритет потока: {4}
22 Состояние потока: {5}",
23 Thread.GetDomain().FriendlyName, Thread.CurrentContext.ContextID, thread.Name, thread.IsAlive, thread.Priority, thread.ThreadState);
24             Console.ReadLine();
25         }
26     }
27 }
```

```
Информация о главном потоке программы
Имя домена приложения: ConsoleApplication1.vshost.exe
ID контекста: 0
Имя потока: MyThread
Запущен ли поток? True
Приоритет потока: Normal
Состояние потока: Running
```

### Задание 3

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading;
6  using System.Windows;
7
8  namespace ConsoleApplication1
9  {
10     public class MyThread
11     {
12         public void ThreadNumbers()
13         {
14             // Информация о потоке
15             Console.WriteLine("{0} поток использует метод ThreadNumbers", Thread.CurrentThread.Name);
16             // Выводим числа
17             Console.Write("Числа: ");
18             for (int i = 0; i < 10; i++)
19             {
20                 Console.Write(i + ", ");
21                 // Используем задержку
22                 Thread.Sleep(3000);
23             }
24             Console.WriteLine();
25         }
26     }
27
28     class Program
29     {
30         static void Main()
31         {
32             Console.Write("Сколько использовать потоков (1 или 2)?");
33             string number = Console.ReadLine();
34
35             Thread mythread = Thread.CurrentThread;
36             mythread.Name = "Первичный";
37
38             // Выводим информацию о потоке
39             Console.WriteLine("--> {0} главный поток", Thread.CurrentThread.Name);
40             MyThread mt = new MyThread();
41
42             switch (number)
43             {
44                 case "1":
45                     mt.ThreadNumbers();
46                     break;
47                 case "2":
48                     // Создаем поток
49                     Thread backgroundThread = new Thread(new ThreadStart(mt.ThreadNumbers));
50                     backgroundThread.Name = "Вторичный";
51                     backgroundThread.Start();
52                     break;
53                 default:
54                     Console.WriteLine("использую 1 поток");
55                     goto case "1";
56             }
57             MessageBox.Show("Сообщение ...", "Работа в другом потоке");
58             Console.ReadLine();
59         }
60     }
61 }
```

Внутри Main() сначала пользователю предлагается решить, сколько потоков применять для выполнения работы приложения: один или два. Если пользователь запрашивает один поток, нужно просто вызвать метод ThreadNumbers() внутри первичного потока. Если же пользователь отдает предпочтение двум потокам, необходимо создать делегат ThreadStart, указывающий на ThreadNumbers(), передать объект делегата конструктору нового объекта Thread и вызвать метод Start(), информируя среду CLR, что этот поток готов к обработке.

Если теперь запустить эту программу в одном потоке, обнаружится, что финальное окно сообщения не отображает сообщения, пока вся последовательность чисел не будет выведена на консоль. Поскольку после вывода каждого числа установлена пауза примерно в 2 секунды, это создаст не

слишком приятное впечатление у пользователя. Однако в случае выбора двух потоков окно сообщения отображается немедленно, поскольку для вывода чисел на консоль выделен отдельный уникальный объект Thread:



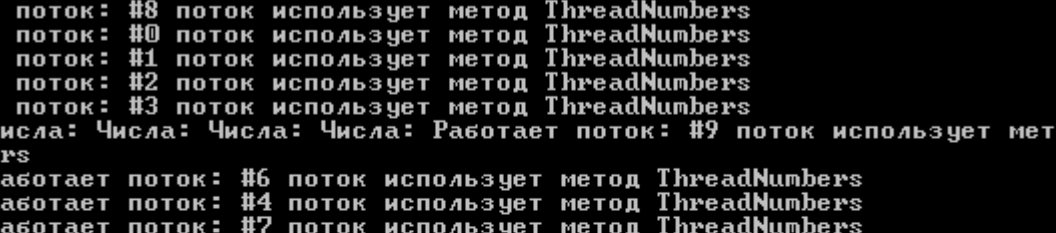
Зачастую в многопоточной программе требуется, чтобы основной поток был последним потоком, завершающим ее выполнение. Формально программа продолжает выполняться до тех пор, пока не завершатся все ее приоритетные потоки. Поэтому требовать, чтобы основной поток завершал выполнение программы, совсем не обязательно. Тем не менее этого правила принято придерживаться в многопоточном программировании, поскольку оно явно определяет конечную точку программы.

#### Задание 4

```
1 using System;
2 using System.Threading;
3
4 namespace ConsoleApplication1
5 {
6     public class MyTheard
7     {
8         public void ThreadNumbers()
9         {
10             // Информация о потоке
11             Console.WriteLine("{0} поток использует метод ThreadNumbers", Thread.CurrentThread.Name);
12             // Выводим числа
13             Console.Write("Числа: ");
14             for (int i = 0; i < 10; i++)
15             {
16                 Random rand = new Random();
17                 Thread.Sleep(1000 * rand.Next(5));
18                 Console.Write(i + " ");
19             }
20             Console.WriteLine();
21         }
22     }
23
24     class Program
25     {
26         static void Main()
27         {
28             MyTheard mt = new MyTheard();
29
30             // Создаем 10 потоков
31             Thread[] threads = new Thread[10];
32
33             for (int i = 0; i < 10; i++)
34             {
35                 threads[i] = new Thread(new ThreadStart(mt.ThreadNumbers));
36                 threads[i].Name = string.Format("Работает поток: #{0}", i);
37             }
38
39             // Запускаем все потоки
40             foreach (Thread t in threads)
41                 t.Start();
42
43             Console.ReadLine();
44         }
45     }
46 }
```

Прежде чем посмотреть на тестовые запуски, давайте еще раз проясним проблему. Первичный поток внутри этого домена приложений начинает свое

существование, порождая десять вторичных рабочих потоков. Каждый рабочий поток должен вызвать метод `ThreadNumbers()` на одном и том же экземпляре `MyTheard`. Учитывая, что никаких мер для блокировки разделяемых ресурсов этого объекта (консоли) не предпринималось, есть хороший шанс, что текущий поток будет отключен, прежде чем метод `ThreadNumbers()` сможет напечатать полные результаты. Поскольку в точности не известно, когда это может случиться (и может ли вообще), будут получаться непредвиденные результаты. Например, может появиться следующий вывод:



```

C:\Users\komputer-pc\Desktop\prob\Zadachi C#\robnik2\robnik2\bin\Debug\r...
Работает поток: #8 поток использует метод ThreadNumbers
Работает поток: #0 поток использует метод ThreadNumbers
Работает поток: #1 поток использует метод ThreadNumbers
Работает поток: #2 поток использует метод ThreadNumbers
Работает поток: #3 поток использует метод ThreadNumbers
Числа: Числа: Числа: Числа: Числа: Работает поток: #9 поток использует метод Thr
eadNumbers
Числа: Работает поток: #6 поток использует метод ThreadNumbers
Числа: Работает поток: #4 поток использует метод ThreadNumbers
Числа: Работает поток: #7 поток использует метод ThreadNumbers
Числа: Работает поток: #5 поток использует метод ThreadNumbers
Числа: 0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
1, 2, 3, 4, 5, 6, 7, 8, 9,
0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3,
3, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7,
8, 8, 7, 7, 8, 8, 8, 8, 8, 9,
9,
9,
8, 9,
9,
9,
9,
9,

```

## Задание 5

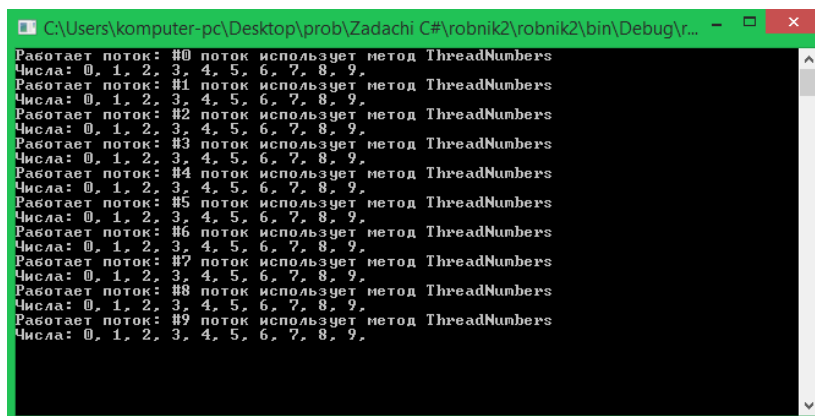
Модифицировать предыдущий пример добавив в него синхронизацию:

```
public class MyTheard
{
    private object threadLock = new object();

    public void ThreadNumbers()
    {
        // Используем маркер-блокировки
        lock (threadLock)
        {
            // Информация о потоке
            Console.WriteLine("{0} поток использует метод ThreadNumbers", Thread.CurrentThread.Name);
            // Выводим числа
            Console.WriteLine("Числа:");
            for (int i = 0; i < 10; i++)
            {
                Random rand = new Random();
                Thread.Sleep(1000 * rand.Next(5));
                Console.WriteLine(i + ", ");
            }
            Console.WriteLine();
        }
    }
}
```

Как только поток войдет в контекст lock, маркер блокировки (в данном случае – текущий объект) станет недоступным другим потокам до тех пор, пока блокировка не будет снята по выходе из контекста lock. Таким образом, если поток А захватит маркер блокировки, другие потоки не смогут войти ни в один из контекстов, использующих тот же маркер, до тех пор, пока поток А не освободит его.

Чтобы блокировать код в статическом методе, нужно объявить приватную статическую переменную-член, которая будет служить в качестве маркера блокировки. Если теперь запустить приложение, можно увидеть, что каждый поток получил возможность выполнить свою работу до конца:



```
C:\Users\komputer-pc\Desktop\prob\Zadachi C#\robnik2\robnik2\bin\Debug\r...
Работает поток: #0 поток использует метод ThreadNumbers
Числа: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Работает поток: #1 поток использует метод ThreadNumbers
Числа: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Работает поток: #2 поток использует метод ThreadNumbers
Числа: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Работает поток: #3 поток использует метод ThreadNumbers
Числа: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Работает поток: #4 поток использует метод ThreadNumbers
Числа: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Работает поток: #5 поток использует метод ThreadNumbers
Числа: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Работает поток: #6 поток использует метод ThreadNumbers
Числа: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Работает поток: #7 поток использует метод ThreadNumbers
Числа: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Работает поток: #8 поток использует метод ThreadNumbers
Числа: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Работает поток: #9 поток использует метод ThreadNumbers
Числа: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
```

## Задание 6

```
1 using System;
2 using System.Threading;
3
4 namespace _Pulse__Wait
5 {
6     class TickTock
7     {
8         private object lockOn = new object();
9
10        public void Tick(bool running)
11        {
12            lock (lockOn)
13            {
14                if (!running)
15                {
16                    // Остановить часы
17                    Monitor.Pulse(lockOn);
18                    return;
19                }
20
21                Console.Write("Тик ");
22                // Разрешить выполнение метода Tock()
23                Monitor.Pulse(lockOn);
24
25                // Ожидать завершения Tock()
26                Monitor.Wait(lockOn);
27            }
28        }
29
30        public void Tock(bool running)
31        {
32            lock (lockOn)
33            {
34                if (!running)
35                {
36                    Monitor.Pulse(lockOn);
37                    return;
38                }
39
40                Console.WriteLine("так");
41                Monitor.Pulse(lockOn);
42                Monitor.Wait(lockOn);
43            }
44        }
45    }
46 }
```



```

46
47     class MyThread
48     {
49         public Thread thrd;
50         TickTock ttobj;
51
52         // Новый поток
53         public MyThread(string name, TickTock tt)
54         {
55             thrd = new Thread(this.Run);
56             ttobj = tt;
57             thrd.Name = name;
58             thrd.Start();
59         }
60
61         void Run()
62         {
63             if (thrd.Name == "Tick")
64             {
65                 for (int i = 0; i < 5; i++)
66                     ttobj.Tick(true);
67                 ttobj.Tick(false);
68             }
69             else
70             {
71                 for (int i = 0; i < 5; i++)
72                     ttobj.Tock(true);
73                 ttobj.Tock(false);
74             }
75         }
76     }
77
78     class Program
79     {
80         static void Main()
81         {
82             TickTock tt = new TickTock();
83             MyThread mt1 = new MyThread("Tick", tt);
84             MyThread mt2 = new MyThread("Tock", tt);
85             mt1.thrd.Join();
86             mt2.thrd.Join();
87
88             Console.WriteLine("Часы остановлены");
89             Console.ReadLine();
90         }
91     }
92 }

```



```

C:\Users\komputer-pc\Desktop\prob\Zadachi C#\robnik2\robnik2\bin\Debug\r...
Тик так
Тик так
Тик так
Тик так
Тик так
Часы остановлены

```

## Задание 7

```
1  using System;
2  using System.Threading;
3
4  namespace ConsoleApplication1
5  {
6      // В этом классе содержится общий ресурс в виде переменной Count,
7      // а так же мьютекс mtx
8      class SharedRes
9      {
10         public static int Count;
11         public static Mutex mtx = new Mutex();
12     }
13
14     // В этом классе Count инкрементируется
15     class IncThread
16     {
17         int num;
18         public Thread Thrd;
19
20         public IncThread(string name, int n)
21         {
22             Thrd = new Thread(this.Run);
23             num = n;
24             Thrd.Name = name;
25             Thrd.Start();
26         }
27
28         // Точка входа в поток
29         void Run()
30         {
31             Console.WriteLine(Thrd.Name + " ожидает мьютекс");
32
33             // Получить мьютекс
34             SharedRes.mtx.WaitOne();
35
36             Console.WriteLine(Thrd.Name + " получает мьютекс");
37
38             do
39             {
40                 Thread.Sleep(500);
41                 SharedRes.Count++;
42                 Console.WriteLine("В потоке {0}, Count={1}", Thrd.Name, SharedRes.Count);
43                 num--;
44             } while (num > 0);
45
46             Console.WriteLine(Thrd.Name + " освобождает мьютекс");
47
48             SharedRes.mtx.ReleaseMutex();
49         }
50     }
51
52     class DecThread
53     {
54         int num;
55         public Thread Thrd;
56
57         public DecThread(string name, int n)
58         {
59             Thrd = new Thread(new ThreadStart(this.Run));
60             num = n;
61             Thrd.Name = name;
62             Thrd.Start();
63         }
64
65         // Точка входа в поток
66         void Run()
67         {
68             Console.WriteLine(Thrd.Name + " ожидает мьютекс");
69
70             // Получить мьютекс
71             SharedRes.mtx.WaitOne();
72
73             Console.WriteLine(Thrd.Name + " получает мьютекс");
74
75             do
76             {
77                 Thread.Sleep(500);
78                 SharedRes.Count--;
79                 Console.WriteLine("В потоке {0}, Count={1}", Thrd.Name, SharedRes.Count);
80                 num--;
81             } while (num > 0);
82
83             Console.WriteLine(Thrd.Name + " освобождает мьютекс");
84
85             SharedRes.mtx.ReleaseMutex();
86         }
87     }
```

```

88
89     class Program
90     {
91     public static void Main()
92     {
93         IncThread mt1 = new IncThread("Inc thread", 5);
94
95         // разрешить инкрементирующему потоку начаться
96         Thread.Sleep(1);
97
98         DecThread mt2 = new DecThread("Dec thread", 5);
99
100        mt1.Thrd.Join();
101        mt2.Thrd.Join();
102
103        Console.ReadLine();
104    }
105 }
106

```

```

C:\Users\komputer-pc\Desktop\prob\Zadachi C#\robnik2\robnik2\bin\Debug\r...
Dec thread ожидает мьютекс
Inc thread ожидает мьютекс
Inc thread получает мьютекс
в потоке Inc thread, Count=1
в потоке Inc thread, Count=2
в потоке Inc thread, Count=3
в потоке Inc thread, Count=4
в потоке Inc thread, Count=5
Inc thread освобождает мьютекс
Dec thread получает мьютекс
в потоке Dec thread, Count=4
в потоке Dec thread, Count=3
в потоке Dec thread, Count=2
в потоке Dec thread, Count=1
в потоке Dec thread, Count=0
Dec thread освобождает мьютекс
-

```

## Задание 8

```

1  using System;
2  using System.Threading;
3
4  namespace ConsoleApplication1
5  {
6      class MyThread
7      {
8          public Thread Thrd;
9          ManualResetEvent mre;
10
11         public MyThread(string name, ManualResetEvent evt)
12         {
13             Thrd = new Thread(this.Run);
14             Thrd.Name = name;
15             mre = evt;
16             Thrd.Start();
17         }
18
19         void Run()
20         {
21             Console.WriteLine("Внутри потока " + Thrd.Name);
22
23             for (int i = 0; i < 5; i++)
24             {
25                 Console.WriteLine(Thrd.Name);
26                 Thread.Sleep(500);
27             }
28
29             Console.WriteLine(Thrd.Name + " завершен!");
30
31             // Уведомление о событии
32             mre.Set();
33         }
34     }

```

```

35
36     class Program
37     {
38         static void Main()
39         {
40             ManualResetEvent evtObj = new ManualResetEvent(false);
41
42             MyThread mt1 = new MyThread("Событийный поток 1", evtObj);
43
44             Console.WriteLine("Основной поток ожидает событие");
45
46             evtObj.WaitOne();
47
48             Console.WriteLine("Основной поток получил уведомление о событии от первого потока");
49
50             evtObj.Reset();
51
52             mt1 = new MyThread("Событийный поток 2", evtObj);
53
54             evtObj.WaitOne();
55
56             Console.WriteLine("Основной поток получил уведомление о событии от второго потока");
57             Console.ReadLine();
58         }
59     }
60 }

```

```

C:\Users\komputer-pc\Desktop\prob\Zadachi C#\robnik2\robnik2\bin\Debug\r...
Основной поток ожидает событие
Внутри потока Событийный поток 1
Событийный поток 1
Событийный поток 1
Событийный поток 1
Событийный поток 1
Событийный поток 1 завершен!
Основной поток получил уведомление о событии от первого потока
Внутри потока Событийный поток 2
Событийный поток 2
Событийный поток 2
Событийный поток 2
Событийный поток 2
Событийный поток 2 завершен!
Основной поток получил уведомление о событии от второго потока

```

### Контрольные вопросы:

1. Что такое потоки?
2. Создание потоков?
3. Запуск потоков?
4. Приостановление потоков?
5. Приоритеты потоков?
6. Изменение типов потоков?
7. Свойства потоков?
8. Локальное хранилище потоков.

## Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК 01.04 Системное программирование

### Практическая работа 9-16

#### Тема: Обмен данными

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль.

#### **Задание:**

##### **Задание 1**

```
1  using System;
2
3  namespace ConsoleApplication1
4  {
5      class UserInfo
6      {
7          public string Name { get; set; }
8          public byte Age { get; set; }
9
10         public UserInfo() { }
11         public UserInfo(string Name, byte Age)
12         {
13             this.Name = Name;
14             this.Age = Age;
15         }
16
17         public override string ToString()
18         {
19             return String.Format(@"Имя пользователя: {0}
20 Возраст: {1}", Name, Age);
21         }
22     }
23
24     class Program
25     {
26         static void Main()
27         {
28             // Создание нового объекта UserInfo в управляемой куче
29             // Возвращаемая ссылка на этот объект user1:
30             UserInfo User1 = new UserInfo("Alex", 26);
31             Console.WriteLine(User1.ToString());
32             Console.ReadLine();
33         }
34
35         public static void MyUser()
36         {
37             UserInfo ui = new UserInfo();
38         }
39     }
40 }
```

При создании приложений на C# можно смело полагать, что исполняющая среда .NET будет сама заботиться об управляемой куче без непосредственного вмешательства со стороны программиста. На самом деле "золотое правило" по управлению памятью в .NET звучит просто:

Размещайте объект в управляемой куче с использованием ключевого слова `new` и забывайте об этом.

После создания объект будет автоматически удален сборщиком мусора тогда, когда в нем отпадет необходимость. Разумеется, возникает вопрос о том, каким образом сборщик мусора определяет момент, когда в объекте отпадает необходимость? В двух словах на этот вопрос можно ответить так: сборщик мусора удаляет объект из кучи тогда, когда тот становится недостижимым ни в одной части программного кода.

Как станет ясно со временем, программирование в среде с автоматической сборкой мусора значительно облегчает разработку приложений. Программистам на C++ хорошо известно, что если они специально не позаботятся об удалении размещаемых в куче объектов, вскоре обязательно начнут возникать "утечки памяти". На самом деле отслеживание проблем, связанных с утечкой памяти, является одним из самых длительных (и утомительных) аспектов программирования в неуправляемых средах. Благодаря назначению ответственным за уничтожение объектов сборщика мусора, обязанности по управлению памятью, по сути, сняты с плеч программиста и возложены на CLR-среду.

## Задание 2

```
01.     using System;
02.     using System.Collections.Generic;
03.     using System.Linq;
04.     using System.Windows.Forms;
05.
06.     namespace From1FormTo2
07.     {
08.         static class Program
09.         {
10.             // The main entry point for the application.
11.             [STAThread]
12.             static void Main()
13.             {
14.                 Application.EnableVisualStyles();
15.                 Application.SetCompatibleTextRenderingDefault(false);
16.                 Application.Run(new MainForm());
17.             }
18.         }
19.     }
```

Код главной формы выглядит так:

```
01.     using System;
02.     using System.Collections.Generic;
03.     using System.ComponentModel;
04.     using System.Data;
05.     using System.Drawing;
06.     using System.Linq;
07.     using System.Text;
08.     using System.Windows.Forms;
09.
10.     namespace From1FormTo2
11.     {
12.         public partial class MainForm : Form
13.         {
14.             //вторая форма
15.             SecondForm secondForm;
16.
17.             //конструктор
18.             public MainForm()
19.             {
20.                 InitializeComponent();
21.             }
22.
23.             //обработчик события передачи данных
24.             //от главной формы ко второй
25.             private void btn_mainForm_Click(object sender, EventArgs e)
26.             {
27.                 secondForm = new SecondForm(tb_mainForm.Text.Trim());
28.                 secondForm.ShowDialog();
29.
30.                 if (secondForm.DialogResult == DialogResult.OK)
31.                     tb_mainForm.Text = secondForm.ReturnData();
32.             }
33.         }
34.     }
```

Соответственно, не забудьте подключить кнопку на событие **Click**. Здесь, в классе главной формы, есть поле **SecondForm secondForm**, представляющее объект-вторую форму. При нажатии на кнопке «Передать», создается вторая форма (вызывается перегруженный конструктор, его мы еще создадим) и запускается методом **ShowDialog()**. В данном случае нам подходит именно этот метод. При чем, после этого мы обязательно проверяем, не закрыли ли вторую форму, а выполнили клик по её кнопке. Если, на второй форме был выполнен клик по кнопке, значит первая форма должна принять данные от второй. Это происходит путем вызова метода **ReturnData()** у второй формы.

Теперь самое интересное – код второй формы:

```

01. using System;
02. using System.Collections.Generic;
03. using System.ComponentModel;
04. using System.Data;
05. using System.Drawing;
06. using System.Linq;
07. using System.Text;
08. using System.Windows.Forms;
09.
10. namespace From1FormTo2
11. {
12.     public partial class SecondForm : Form
13.     {
14.         //перегруженный конструктор
15.         public SecondForm(string data)
16.         {
17.             InitializeComponent();
18.             tb_secondForm.Text = data;
19.         }
20.
21.         //обработчик события передачи данных
22.         //от второй формы к главной
23.         private void btn_secondForm_Click
24.             (object sender, EventArgs e)
25.         {
26.             this.DialogResult = DialogResult.OK;
27.         }
28.
29.         //открытый метод для доступа к
30.         //текстовому полю данной формы
31.         public string ReturnData()
32.         {
33.             return (tb_secondForm.Text.Trim());
34.         }
35.     }
36. }

```

Как видим, имеется единственная перегрузка конструктора, который принимает тип **string**. Помним, что мы пытаемся передавать текст из **TextBox**. В конструкторе происходит плановая инициализация компонент и установка текста текстового поля в переданное значение от первой формы. Далее, подписавшись на событие **Click** для кнопки второй формы, мы создали обработчик **btn\_secondForm\_Click**, который и имитирует работу кнопки «Ok» любого диалогового окна. Таким образом, нажимая на кнопке «Отправить» (второй формы), мы приводим в исполнение условие

**(secondForm.DialogResult == DialogResult.OK)**

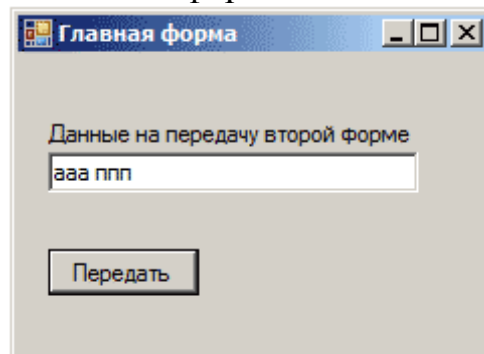
первой формы, ввиду чего, вызывая метод **secondForm.ReturnData()**, мы устанавливаем текстовое поле первой формы в значение текстового поля второй формы.

Работа данного метода, я думаю, уже не требует пояснений. Он просто возвращает текст из единственного текстового поля, при этом, оставляя его приватным.

В итоге, мы передали данные во вторую форму из первой и со второй в первую не нарушая принципы инкапсуляции.



Попробуйте внести текст «aaa» в текстовое поле первой формы и выполнить нажатие на кнопке. Вы увидите в открывшейся второй форме этот текст в её текстовом поле. Попробуйте изменить текст на «aaa ппп» и нажать на кнопку. Вы увидите как после закрытия второй формы данный текст отобразится в текстовом поле главной формы.



### Задание 3

Взаимодействие между различными приложениями может осуществляться по разному, например, через сокет. Но в .NET 4.0 была введена новая функциональность, которая представляет собой создание участка общей разделяемой памяти для приложений.

Основной функционал новой технологии заключается в пространстве имен System.IO.MemoryMappedFiles.

Создадим два консольных приложения, одно из которых будет посылать сообщение в общую память, а другое - считывать это сообщение.

Для этого нам потребуются два класса: MemoryMappedFile - он будет создавать участок разделяемой памяти и MemoryMappedViewAccessor - с его помощью мы будем проводить взаимодействие (чтение/запись) с общей памятью. Код приложения, записывающего сообщение в память.

Программа записи:

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.IO.MemoryMappedFiles;
6
7  namespace WriteMemoryAp
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("Введите сообщение");
14             //Ввод выражения для записи в общую память
15             char[] message = Console.ReadLine().ToCharArray();
16             //Размер введенного сообщения
17             int size = message.Length;
18
19             //Создание участка разделяемой памяти
20             //Первый параметр - название участка,
21             //второй - длина участка памяти в байтах: тип char занимает 2 байта
22             //плюс четыре байта для одного объекта типа Integer
23             MemoryMappedFile sharedMemory = MemoryMappedFile.CreateOrOpen("MemoryFile", size * 2 + 4);
24             //Создаем объект для записи в разделяемый участок памяти
25             using (MemoryMappedViewAccessor writer = sharedMemory.CreateViewAccessor(0, size * 2 + 4))
26             {
27                 //запись в разделяемую память
28                 //запись размера с нулевого байта в разделяемой памяти
29                 writer.Write(0, size);
30                 //запись сообщения с четвертого байта в разделяемой памяти
31                 writer.WriteArray<char>(4, message, 0, message.Length);
32             }
33
34             Console.WriteLine("Сообщение записано в разделяемую память");
35             Console.WriteLine("Для выхода из программы нажмите любую клавишу");
36             Console.ReadLine();
37         }
38     }
39 }

```

### Программа чтения:

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.IO.MemoryMappedFiles;
6
7  namespace ReadMemoryAp
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             //Массив для сообщения из общей памяти
14             char[] message;
15             //Размер введенного сообщения
16             int size;
17
18             //Получение существующего участка разделяемой памяти
19             //Параметр - название участка
20             MemoryMappedFile sharedMemory = MemoryMappedFile.OpenExisting("MemoryFile");
21             //Сначала считываем размер сообщения, чтобы создать массив данного размера
22             //Integer занимает 4 байта, начинается с первого байта, поэтому передаем цифры 0 и 4
23             using (MemoryMappedViewAccessor reader = sharedMemory.CreateViewAccessor(0, 4, MemoryMappedFileAccess.Read))
24             {
25                 size = reader.ReadInt32(0);
26             }
27
28             //Считываем сообщение, используя полученный выше размер
29             //Сообщение - это строка или массив объектов char, каждый из которых занимает два байта
30             //Поэтому вторым параметром передаем число символов умножив на из размер в байтах плюс
31             //А первый параметр - смещение - 4 байта, которое занимает размер сообщения
32             using (MemoryMappedViewAccessor reader = sharedMemory.CreateViewAccessor(4, size * 2, MemoryMappedFileAccess.Read))
33             {
34                 //Массив символов сообщения
35                 message = new char[size];
36                 reader.ReadArray<char>(0, message, 0, size);
37             }
38             Console.WriteLine("Получено сообщение :");
39             Console.WriteLine(message);
40             Console.WriteLine("Для выхода из программы нажмите любую клавишу");
41             Console.ReadLine();
42         }
43     }
44 }
45

```

#### Задание 4

1. Создайте пример текстового файла в Блокноте. Выполните приведенные ниже действия.

- a) Вставьте текст hello world в Блокнот.
- b) Сохраните файл как Sample.txt.

2. Запустите Microsoft Visual Studio.

3. В меню Файл выберите пункт Создать, а затем выберите Проект.

4. Выберите Проекты Visual C# в разделе Типы проектов, а затем выберите Консольное приложение в разделе Шаблоны.

5. Добавьте следующий код в начало файла Class1.cs:

***using System.IO;***

6. Добавьте указанный ниже код в метод Main:

```
String line;
try
{
    //Pass the file path and file name to the StreamReader constructor
    StreamReader sr = new StreamReader("C:\\Sample.txt");
    //Read the first line of text
    line = sr.ReadLine();
    //Continue to read until you reach end of file
    while (line != null)
    {
        //write the line to console window
        Console.WriteLine(line);
        //Read the next line
        line = sr.ReadLine();
    }
    //close the file
    sr.Close();
    Console.ReadLine();
}
catch(Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
finally
{
    Console.WriteLine("Executing finally block.");
}
```

7. В меню Отладка выберите Пуск для компиляции и запуска приложения. Нажмите клавишу ВВОД, чтобы закрыть окно консоли. В окне консоли отображается содержимое файла Sample.txt:

Вывод консоли: Hello world.

```

1  //Read a Text File
2  using System;
3  using System.IO;
4  namespace readwriteapp
5  {
6      class Class1
7      {
8          [STAThread]
9          static void Main(string[] args)
10         {
11             String line;
12             try
13             {
14                 //Pass the file path and file name to the StreamReader constructor
15                 StreamReader sr = new StreamReader("C:\\Sample.txt");
16                 //Read the first line of text
17                 line = sr.ReadLine();
18                 //Continue to read until you reach end of file
19                 while (line != null)
20                 {
21                     //write the line to console window
22                     Console.WriteLine(line);
23                     //Read the next line
24                     line = sr.ReadLine();
25                 }
26                 //close the file
27                 sr.Close();
28                 Console.ReadLine();
29             }
30             catch(Exception e)
31             {
32                 Console.WriteLine("Exception: " + e.Message);
33             }
34             finally
35             {
36                 Console.WriteLine("Executing finally block.");
37             }
38         }
39     }
40 }

```

## Задание 5

В следующем коде используется класс StreamWriter для открытия, записи и закрытия текстового файла. Аналогично тому, как используется класс StreamReader, можно передать путь к текстовому файлу в конструктор StreamWriter для автоматического открытия файла. Метод WriteLine записывает всю текстовую строку в текстовый файл.

1. Запустите Visual Studio.
2. В меню Файл выберите пункт Создать, а затем выберите Проект.
3. Выберите Проекты Visual C# в разделе Типы проектов, а затем выберите Консольное приложение в разделе Шаблоны.
4. Добавьте следующий код в начало файла Class1.cs:  
**using System.IO;**
5. Добавьте указанный ниже код в метод Main:

```

try
{
    //Pass the filepath and filename to the StreamWriter Constructor
    StreamWriter sw = new StreamWriter("C:\\Test.txt");
    //Write a line of text
    sw.WriteLine("Hello World!!");
    //Write a second line of text
    sw.WriteLine("From the StreamWriter class");
    //Close the file
    sw.Close();
}
catch(Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
finally
{
    Console.WriteLine("Executing finally block.");
}
}

```

6. В меню Отладка выберите Пуск для компиляции и запуска приложения. Этот код создает файл с именем Test.txt на диске C. Откройте Test.txt в текстовом редакторе, например в Блокноте. Test.txt содержит две текстовые строки:

Hello World!!

From the StreamWriter class

```

1  //Write a text file - Version-1
2  using System;
3  using System.IO;
4  namespace readwriteapp
5  {
6      class Class1
7      {
8          [STAThread]
9          static void Main(string[] args)
10         {
11             try
12             {
13                 //Pass the filepath and filename to the StreamWriter Constructor
14                 StreamWriter sw = new StreamWriter("C:\\Test.txt");
15                 //Write a line of text
16                 sw.WriteLine("Hello World!!");
17                 //Write a second line of text
18                 sw.WriteLine("From the StreamWriter class");
19                 //Close the file
20                 sw.Close();
21             }
22             catch(Exception e)
23             {
24                 Console.WriteLine("Exception: " + e.Message);
25             }
26             finally
27             {
28                 Console.WriteLine("Executing finally block.");
29             }
30         }
31     }
32 }

```

## Задание 6

В следующем коде используется класс `StreamWriter` для открытия, записи и закрытия текстового файла. В отличие от предыдущего примера, этот код передает в конструктор два дополнительных параметра. Первый параметр — путь к файлу и имя файла. Второй параметр, `true`, указывает, что файл открыт в режиме добавления. Если вы зададите `false` для второго параметра, содержимое файла перезаписывается при каждом запуске кода. Третий параметр задает `Unicode`, чтобы кодирование файла в `StreamWriter` выполнялось в формате Юникода. Можно также указать следующие методы кодирования для третьего параметра:

- ASCII
- Юникод
- UTF7
- UTF8

Метод `Write` аналогичен методу `WriteLine`, за исключением того, что метод `Write` не вставляет автоматически сочетание символов возврата каретки или перевода строки (CR/LF). Это полезно, когда нужно одновременно записывать по одному символу.

1. Запустите Visual Studio.

2. В меню Файл выберите пункт Создать и затем пункт Проект.

3. Нажмите Проекты Visual C# в разделе Типы проектов, а затем нажмите Консольное приложение в разделе Шаблоны.

4. Добавьте следующий код в начало файла `Class1.cs`:

***using System.IO;***

***using System.Text;***

5. Добавьте указанный ниже код в метод `Main`:

```
Int64 x;
try
{
    //Open the File
    StreamWriter sw = new StreamWriter("C:\\Test1.txt", true, Encoding.ASCII);

    //Write out the numbers 1 to 10 on the same line.
    for(x=0; x < 10; x++)
    {
        sw.Write(x);
    }

    //close the file
    sw.Close();
}
catch(Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
}
finally
{
    Console.WriteLine("Executing finally block.");
}
```

6. В меню Отладка выберите Пуск для компиляции и запуска приложения. Этот код создает файл с именем `Test1.txt` на диске C. Откройте

Test1.txt в текстовом редакторе, например в Блокноте. Test1.txt содержит одну текстовую строку: 0123456789.

```
1 //Write a text file - Version 2
2 using System;
3 using System.IO;
4 using System.Text;
5 namespace readwriteapp
6 {
7     class Class1
8     {
9         [STAThread]
10        static void Main(string[] args)
11        {
12            Int64 x;
13            try
14            {
15                //Open the File
16                StreamWriter sw = new StreamWriter("C:\\Test1.txt", true, Encoding.ASCII);
17                //Writeout the numbers 1 to 10 on the same line.
18                for(x=0; x < 10; x++)
19                {
20                    sw.Write(x);
21                }
22                //close the file
23                sw.Close();
24            }
25            catch(Exception e)
26            {
27                Console.WriteLine("Exception: " + e.Message);
28            }
29            finally
30            {
31                Console.WriteLine("Executing finally block.");
32            }
33        }
34    }
35 }
```

### Устранение неполадок

Перенос кода в блок **try-catch-finally** для обработки ошибок и исключений является хорошей практикой программирования, когда речь заходит о выполнении любых операций с файлом. В частности, может потребоваться освободить дескрипторы файла в окончательном блоке, чтобы файл не был заблокирован на неопределенный срок. Некоторые возможные ошибки включают файл, который не существует, или файл, который уже используется.

### Задание 7

Создать новое консольное приложение в котором нужно отобразить содержимое текстового файла. Текстовый файл помещается в папку нового проекта с названием **Text.txt**.



```

1  using System;
2  using System.IO;
3
4  class Example
5  {
6      static void Main()
7      {
8          FileStream fin;
9          string s;
10         try
11         {
12             fin = new FileStream("C:/Temp/test.txt", FileMode.Open);
13         }
14         catch (IOException exc)
15         {
16             Console.WriteLine("Ошибка открытия файла:\n" + exc.Message);
17             return;
18         }
19         StreamReader fstr_in = new StreamReader(fin);
20         try
21         {
22             while ((s = fstr_in.ReadLine()) != null)
23             {
24                 Console.WriteLine(s);
25             }
26         }
27         catch (IOException exc)
28         {
29             Console.WriteLine("Ошибка ввода-вывода:\n" + exc.Message);
30         }
31
32         finally
33         {
34             fstr_in.Close();
35         }
36         Console.WriteLine("Нажмите любую кнопку!");
37         Console.ReadKey();
38     }
39 }
40
41 }

```

Пример полученного вывода:



```
(Lennon/McCartney)

Oh! Darling, please believe me
I'll never do you no harm
Believe me when I tell you
I'll never do you no harm

Oh! Darling, if you leave me
I'll never make it alone
Believe me when I beg you
Don't ever leave me alone

When you told me you didn't need me anymore
Well you know I nearly broke down and cried
When you told me you didn't need me anymore
Well you know I nearly broke down and died

Oh! Darling, if you leave me
I'll never make it alone
Believe me when I tell you
I'll never do you no harm

When you told me you didn't need me anymore
Well you know I nearly broke down and cried
When you told me you didn't need me anymore
Well you know I nearly broke down and died

Oh! Darling, please believe me
I'll never let you down
Believe me when I tell you
I'll never do you no harm
Нажмите любую кнопку!
```

## Задание 8

Создать новое консольное приложение: Сортировка целочисленного массива. Бинарный поиск элементов. Полученные значения, вывод консоли записать в текстовый файл.

```

1  using System;
2
3  class arraySort
4  {
5
6      static void print(int[] alpha)
7      {
8          foreach (int i in alpha)
9              Console.WriteLine(i + " ");
10         Console.WriteLine();
11     }
12
13     static void Main()
14     {
15
16         int[] alpha = { -4, 8, 28, -6, -98, -47, 12 };
17         //Отобразить исходный массив
18
19         Console.WriteLine("Исходный массив ...");
20         print(alpha);
21
22         //Найти значение 8
23         int pattern = 8;
24         int indx = Array.BinarySearch(alpha, pattern);
25         Console.WriteLine("Индекс элемента массива со значением 8: " + indx);
26
27         //Сортируем массив
28         Array.Sort(alpha);
29         Console.WriteLine("Сортировка по возрастанию ...");
30         //Отображаем массив после сортировки
31         print(alpha);
32
33         //Сортируем массив по убыванию
34         Array.Reverse(alpha);
35         //Отображаем массив после сортировки по убыванию
36         Console.WriteLine("Сортировка по убыванию ...");
37         print(alpha);
38         Console.WriteLine("Нажмите любую кнопку!");
39         Console.ReadKey();
40     }
41 }

```

Пример полученного вывода:

```
Исходный массив ...
-4
8
28
-6
-98
-47
12

Индекс элемента массива со значением 8: -7
Сортировка по возрастанию ...
-98
-47
-6
-4
8
12
28

Сортировка по убыванию ...
28
12
8
-4
-6
-47
-98

Нажмите любую кнопку!
```

### **Контрольные вопросы**

- 1) Обмен данными?
- 2) Типы обмена данными?
- 3) Различия обмена данных?
- 4) Обмен данными между процессами?
- 5) Использование буфера обмена для IPC?
- 6) Использование COM для IPC?
- 7) Использование копирования данных для IPC?
- 8) Использование DDE для IPC?

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК 01.04 Системное программирование**

Практическая работа 17-24

**Тема: Сетевое программирование сокетов**

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль.

## Задание:

### Задание 1

#### Сервер TCP

Создание структуры сервера показано на следующей функциональной диаграмме:



Вот полный код программы SocketServer.cs:

```

1  // SocketServer.cs
2  using System;
3  using System.Text;
4  using System.Net;
5  using System.Net.Sockets;
6
7  namespace SocketServer
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             // Устанавливаем для сокета локальную конечную точку
14             IPHostEntry ipHost = Dns.GetHostEntry("localhost");
15             IPAddress ipAddr = ipHost.AddressList[0];
16             IPEndPoint ipEndPoint = new IPEndPoint(ipAddr, 11000);
17
18             // Создаем сокет Tcp/Ip
19             Socket sListener = new Socket(ipAddr.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
20
21             // Назначаем сокет локальной конечной точке и слушаем входящие сокеты
22             try
23             {
24                 sListener.Bind(ipEndPoint);
25                 sListener.Listen(10);
26
27                 // Начинаем слушать соединения
28                 while (true)
29                 {
30                     Console.WriteLine("Ожидаем соединение через порт {0}", ipEndPoint);
31
32                     // Программа приостанавливается, ожидая входящее соединение
33                     Socket handler = sListener.Accept();
34                     string data = null;
35
36                     // Мы дождались клиента, пытающегося с нами соединиться
37
38                     byte[] bytes = new byte[1024];
39                     int bytesRec = handler.Receive(bytes);
40
41                     data += Encoding.UTF8.GetString(bytes, 0, bytesRec);
42
43                     // Показываем данные на консоли
44                     Console.Write("Полученный текст: " + data + "\n\n");
45
46                     // Отправляем ответ клиенту\
47                     string reply = "Спасибо за запрос в " + data.Length.ToString()
48                                     + " символов";
49                     byte[] msg = Encoding.UTF8.GetBytes(reply);
50                     handler.Send(msg);
51
52                     if (data.IndexOf("<TheEnd>") > -1)
53                     {
54                         Console.WriteLine("Сервер завершил соединение с клиентом.");
55                         break;
56                     }
57
58                     handler.Shutdown(SocketShutdown.Both);
59                     handler.Close();
60                 }
61             }
62             catch (Exception ex)
63             {
64                 Console.WriteLine(ex.ToString());
65             }
66             finally
67             {
68                 Console.ReadLine();
69             }
70         }
71     }
72 }

```

## Задание 2

### Клиент на TCP

Функции, которые используются для создания приложения-клиента, более или менее напоминают серверное приложение. Как и для сервера, используются те же методы для определения конечной точки, создания экземпляра сокета, отправки и получения данных и закрытия сокета:

```
1  // SocketClient.cs
2  using System;
3  using System.Text;
4  using System.Net;
5  using System.Net.Sockets;
6
7  namespace SocketClient
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             try
14             {
15                 SendMessageFromSocket(11000);
16             }
17             catch (Exception ex)
18             {
19                 Console.WriteLine(ex.ToString());
20             }
21             finally
22             {
23                 Console.ReadLine();
24             }
25         }
26
27         static void SendMessageFromSocket(int port)
28         {
29             // Буфер для входящих данных
30             byte[] bytes = new byte[1024];
31
32             // Соединяемся с удаленным устройством
33
34             // Устанавливаем удаленную точку для сокета
35             IPHostEntry ipHost = Dns.GetHostEntry("localhost");
36             IPAddress ipAddr = ipHost.AddressList[0];
37             IPEndPoint ipEndPoint = new IPEndPoint(ipAddr, port);
38
39             Socket sender = new Socket(ipAddr.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
40
41             // Соединяем сокет с удаленной точкой
42             sender.Connect(ipEndPoint);
```

```

43
44     Console.WriteLine("Введите сообщение: ");
45     string message = Console.ReadLine();
46
47     Console.WriteLine("Сокет соединяется с {0} ", sender.RemoteEndPoint.ToString());
48     byte[] msg = Encoding.UTF8.GetBytes(message);
49
50     // Отправляем данные через сокет
51     int bytesSent = sender.Send(msg);
52
53     // Получаем ответ от сервера
54     int bytesRec = sender.Receive(bytes);
55
56     Console.WriteLine("\nОтвет от сервера: {0}\n\n", Encoding.UTF8.GetString(bytes, 0, bytesRec));
57
58     // Используем рекурсию для неоднократного вызова SendMessageFromSocket()
59     if (message.IndexOf("<TheEnd>") == -1)
60     {
61         SendMessageFromSocket(port);
62     }
63
64     // Освобождаем сокет
65     sender.Shutdown(SocketShutdown.Both);
66     sender.Close();
67 }

```

Единственный новый метод - метод **Connect()**, используется для соединения с удаленным сервером. На рисунке ниже показаны клиент и сервер в действии:

```

file:///C:/AddInTest/SocketServer/SocketServer/bin/Debug/SocketServer.EXE
Ожидает соединение через порт [::1]:11000
Полученный текст: Привет
Ожидает соединение через порт [::1]:11000
Полученный текст: Я клиент
Ожидает соединение через порт [::1]:11000
Полученный текст: <TheEnd>
Сервер завершил соединение с клиентом.

file:///C:/AddInTest/SocketClient/SocketClient/bin/Debug/SocketClient.EXE
Введите сообщение: Привет
Сокет соединяется с [::1]:11000
Ответ от сервера: Спасибо за запрос в 6 символов

Введите сообщение: Я клиент
Сокет соединяется с [::1]:11000
Ответ от сервера: Спасибо за запрос в 8 символов

Введите сообщение: <TheEnd>
Сокет соединяется с [::1]:11000
Ответ от сервера: Спасибо за запрос в 8 символов

```

### Задание 3

В предыдущем задании было рассмотрено самое простое приложение клиент-сервер, теперь с помощью исключения **SocketException** создадим нечто более интересное.

В следующем примере создадим собственную программу сканирования портов, которая пытается соединиться с **localhost** по каждому порту, указанному в цикле. Сообщаем об успешных соединениях, а если установить

соединение не удастся, перехватываем порождаемое в этом случае исключение **SocketException**. В качестве графической среды используется **WPF**.

Сканер портов может использоваться для получения списка открытых портов на вашем компьютере. В открытых портах проявляется потенциальная слабость системы, которой могут воспользоваться приложения-нарушители. Вот полный код программы, включая исходную **XAML**-разметку:

```
1 <Window x:Class="SocketPortScanner.MainWindow"
2       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4       Title="Сканер портов" WindowState="Maximized">
5     <ListView Name="listview_scanner" Margin="5">
6       <ListView.Resources>
7         <Style TargetType="{x:Type ListView}">
8           <Setter Property="ItemContainerStyle">
9             <Setter.Value>
10              <Style TargetType="ListViewItem">
11                <Setter Property="HorizontalContentAlignment" Value="Center"/>
12              </Style>
13            </Setter.Value>
14          </Setter>
15        </Style>
16      </ListView.Resources>
17      <ListView.View>
18        <GridView>
19          <GridView.Columns>
20            <GridViewColumn Header="Port ID" DisplayMemberBinding="{Binding Path=PortNumber}"
21                          Width="150"/>
22            <GridViewColumn Header="Local Adress" DisplayMemberBinding="{Binding Path=Local}"
23                          Width="250"/>
24            <GridViewColumn Header="Remote Adress" DisplayMemberBinding="{Binding Path=Remote}" Width="250"/>
25            <GridViewColumn Header="State" DisplayMemberBinding="{Binding Path=State}" Width="250"/>
26          </GridView.Columns>
27        </GridView>
28      </ListView.View>
29    </ListView>
30 </Window>
```



```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Windows;
5  using System.Windows.Data;
6  using System.Net;
7  using System.Net.Sockets;
8  using System.Net.NetworkInformation;
9
10 namespace SocketPortScanner
11 {
12     public partial class MainWindow : Window
13     {
14         public MainWindow()
15         {
16             InitializeComponent();
17
18             List<PortInfo> pi = MainWindow.GetOpenPort();
19             listview_scanner.ItemsSource = pi;
20         }
21
22         private static List<PortInfo> GetOpenPort()
23         {
24             IPGlobalProperties properties = IPGlobalProperties.GetIPGlobalProperties();
25             IPEndPoint[] tcpEndPoints = properties.GetActiveTcpListeners();
26             TcpConnectionInformation[] tcpConnections = properties.GetActiveTcpConnections();
27
28             return tcpConnections.Select(p =>
29                 {
30                     return new PortInfo(
31                         i: p.LocalEndPoint.Port,
32                         local: String.Format("{0}:{1}", p.LocalEndPoint.Address, p.LocalEndPoint.Port),
33                         remote: String.Format("{0}:{1}", p.RemoteEndPoint.Address, p.RemoteEndPoint.Port),
34                         state: p.State.ToString());
35                 }).ToList();
36         }
37     }
38
39     class PortInfo
40     {
41         public int PortNumber { get; set; }
42         public string Local { get; set; }
43         public string Remote { get; set; }
44         public string State { get; set; }
45
46         public PortInfo(int i, string local, string remote, string state)
47         {
48             PortNumber = i;
49             Local = local;
50             Remote = remote;
51             State = state;
52         }
53     }
54 }

```

Port ID	Local Address	Remote Address	State
16201	72.27.89.88:16201	80.89.133.162:80	Established
23964	72.27.89.88:23964	93.88.162.77:443	Established
23965	72.27.89.88:23965	93.88.162.48:443	Established
23981	72.27.89.88:23981	93.88.162.78:80	CloseWait
24053	72.27.89.88:24053	109.229.11.157:63073	SynSent
24056	72.27.89.88:24056	78.132.154.114:60874	SynSent
24057	72.27.89.88:24057	78.106.172.228:26290	SynSent
24059	72.27.89.88:24059	78.26.220.69:63803	TimeWait
24060	72.27.89.88:24060	78.26.144.147:50210	SynSent
24061	72.27.89.88:24061	77.223.93.24:52029	SynSent
24063	72.27.89.88:24063	77.35.187.9:35691	SynSent
24065	72.27.89.88:24065	91.146.60.162:48966	SynSent
24066	72.27.89.88:24066	77.35.249.72:62902	SynSent
24067	72.27.89.88:24067	128.0.129.69:11890	SynSent
24068	72.27.89.88:24068	77.35.25.143:62813	SynSent
24069	72.27.89.88:24069	62.182.65.28:10457	SynSent
24070	72.27.89.88:24070	62.32.68.19:63259	SynSent
24071	72.27.89.88:24071	88.215.137.104:32022	Established
24072	72.27.89.88:24072	128.70.107.239:35691	SynSent
24073	72.27.89.88:24073	128.72.122.33:61384	SynSent
24074	72.27.89.88:24074	134.255.154.42:35691	SynSent
24075	72.27.89.88:24075	141.101.235.9:42575	SynSent
24076	72.27.89.88:24076	46.229.143.101:43276	SynSent
62676	72.27.89.88:62676	23.60.69.151:80	CloseWait
65296	72.27.89.88:65296	75.126.203.247:80	CloseWait

## Задание 4

### Императивная безопасность

Этот синтаксис для обеспечения безопасности создает новый экземпляр класса **SocketPermission**. Синтаксис императивной безопасности можно использовать для выполнения требований и переопределений, но не запросов. Прежде чем вызвать соответствующий критерий безопасности, необходимо через конструктор инициализировать состояние класса **SocketPermission**, чтобы он представлял конкретную форму разрешения, которую вы ищете.

В следующем приложении демонстрируется основное использование класса **SocketPermission**. Поскольку этот код ведет себя как клиент, до выполнения этой программы нужно запустить приложение **SocketServer.cs**, созданное ранее:

```

1  using System;
2  using System.Text;
3  using System.Net;
4  using System.Net.Sockets;
5  using System.Security;
6  using System.Security.Permissions;
7
8  namespace SocketPermissionSample
9  {
10     class Program
11     {
12         static void Main(string[] args)
13         {
14             // Устанавливаем удаленную точку для сокета
15             IPHostEntry ipHost = Dns.GetHostEntry("localhost");
16             IPAddress ipAddr = ipHost.AddressList[0];
17             IPEndPoint ipEndPoint = new IPEndPoint(ipAddr, 11000);
18
19             Socket sender = new Socket(ipAddr.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
20
21             // Настройка разрешений
22             SocketPermission permisSocket = new SocketPermission(
23                 NetworkAccess.Connect, TransportType.Tcp, "localhost",
24                 SocketPermission.AllPorts);
25
26             permisSocket.Assert();
27
28             try
29             {
30                 // Соединяем сокет с удаленной endPoint, перехватываем все ошибки
31                 sender.Connect(ipEndPoint);
32                 Console.WriteLine("Сокет подключен к {0}", sender.RemoteEndPoint.ToString());
33
34                 byte[] bytes = new byte[1024];
35                 byte[] msg = Encoding.UTF8.GetBytes("Простой тест");
36
37                 // Отправляем данные через сокет
38                 int bytesSend = sender.Send(msg);
39
40                 // Получаем ответ от удаленного устройства
41                 int bytesRec = sender.Receive(bytes);
42
43                 Console.WriteLine("Текст ответа: {0}",
44                     Encoding.UTF8.GetString(bytes, 0, bytesRec));
45             }
46             catch (Exception ex)
47             {
48                 Console.WriteLine(ex.ToString());
49             }
50             finally
51             {
52                 if (sender.Connected)
53                 {
54                     // Освобождаем сокет
55                     sender.Shutdown(SocketShutdown.Both);
56                     sender.Close();
57                 }
58             }
59             Console.ReadLine();
60         }
61     }
62 }

```

Приведенный код показывает, как, используя императивный синтаксис, реализовать безопасность доступа к коду. Важность этого кода в данном случае заключается в вызове одного метода класса **SocketPermission** - **Assert()**, который указывает, что приложению разрешается давать согласие на запросы соединения от Интернета и местного ресурса.

## Задание 5

### Декларативная безопасность

Декларативная безопасность использует атрибуты .NET, чтобы поместить информацию о безопасности внутрь метаданных кода. Атрибуты можно поместить на уровне сборки, класса или члена и указать необходимый тип запроса, требования или переопределения. Для использования этого синтаксиса безопасности сначала через декларативный синтаксис нужно инициализировать данные состояния объекта **SocketPermissionAttribute**, чтобы он представлял форму разрешения, соблюдение которого обеспечивается в коде.

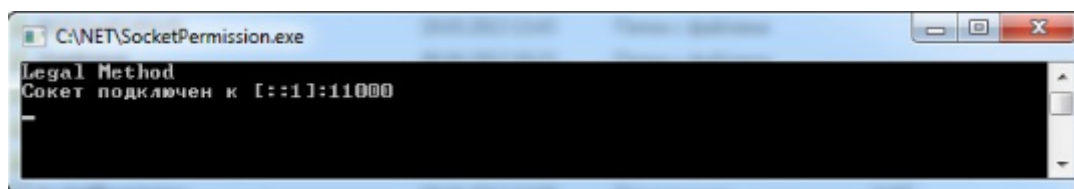
В следующем примере демонстрируется, как обеспечить выполнение разрешения с использованием **SocketPermissionAttribute**:

```
1  using System;
2  using System.Text;
3  using System.Net;
4  using System.Net.Sockets;
5  using System.Security;
6  using System.Security.Permissions;
7
8  namespace SocketPermissionSample
9  {
10     class Program
11     {
12         static void Main(string[] args)
13         {
14             LegalMethod();
15             Console.ReadLine();
16         }
17
18         [SocketPermission(SecurityAction.Assert, Access="Connect",
19             Host="localhost", Port="All", Transport="Tcp")]
20         public static void LegalMethod()
21         {
22             Console.WriteLine("Legal Method");
23
24             // Устанавливаем удаленную точку для сокета
25             IPHostEntry ipHost = Dns.GetHostEntry("localhost");
26             IPAddress ipAddr = ipHost.AddressList[0];
27             IPEndPoint ipEndPoint = new IPEndPoint(ipAddr, 11000);
28
29             Socket sender = new Socket(ipAddr.AddressFamily,
30                 SocketType.Stream, ProtocolType.Tcp);
```

```

32     try
33     {
34         // Соединяем сокет с удаленной endPoint
35         sender.Connect(ipEndPoint);
36         Console.WriteLine("Сокет подключен к {0}",
37             sender.RemoteEndPoint.ToString());
38     }
39     catch (Exception ex)
40     {
41         Console.WriteLine(ex.ToString());
42     }
43     finally
44     {
45         if (sender.Connected)
46         {
47             // Освобождаем сокет
48             sender.Shutdown(SocketShutdown.Both);
49             sender.Close();
50         }
51     }
52 }
53 }
54 }
55 }

```



По функциональности приведенная выше программа похожа на предыдущую, однако, в последней программе вместо императивной безопасности используется синтаксис декларативной безопасности, т.е. разрешения для сокета задаются в атрибуте `SocketPermission`.

### Контрольные вопросы

1. Понятие технологии Windows Sockets.
2. Основные этапы работы с сокетами.
3. Понятие технологии «клиент-сервер».
4. Что такое полнодуплексное взаимодействие?
5. Что такое динамически подключаемые библиотеки (DLL)?
6. Что такое прикладной программный интерфейс (WinAPI)?
7. Блокирующие и неблокирующие сокеты.
8. Модель OSI, модель TCP/IP.

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК 01.04 Системное программирование**

Практическая работа 25-30

**Тема: Работы с буфером экрана**

**Форма контроля:** входной контроль; оперативный контроль; рубежный контроль.

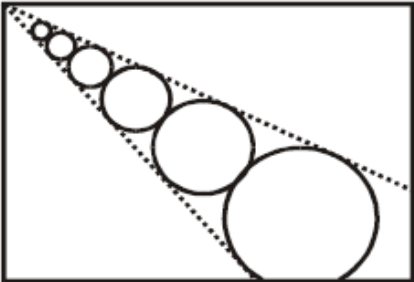
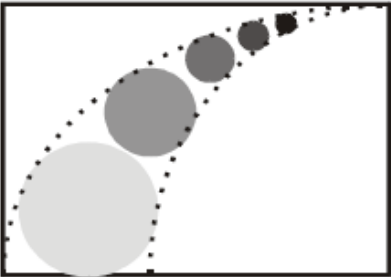
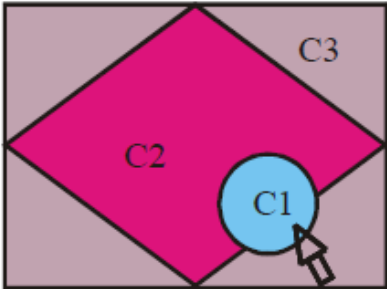
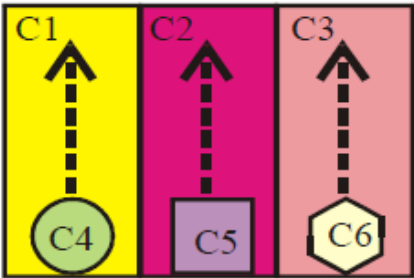
**Задание:**


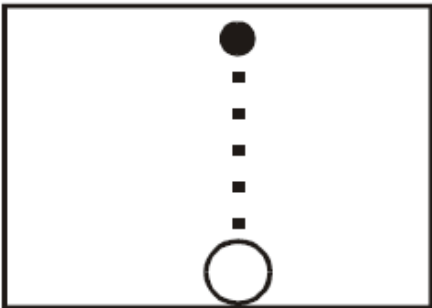
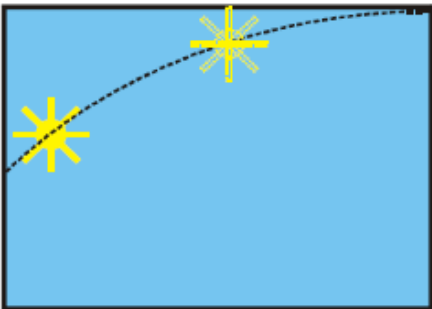
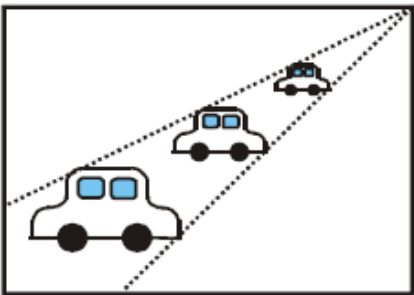
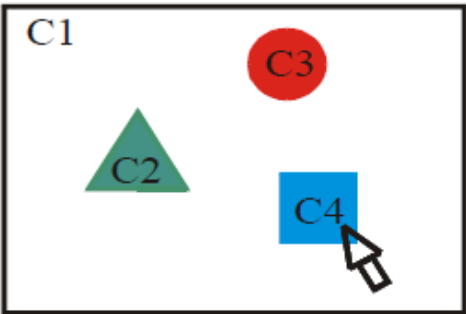
**Задание**

Фигуры, на которые на рисунке указывает курсор мыши, двигаются по экрану при нажатой левой клавиши мыши по траектории движения мыши.

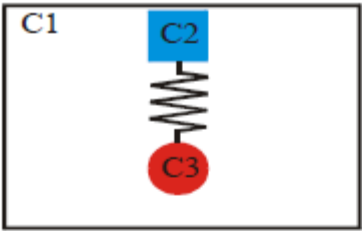
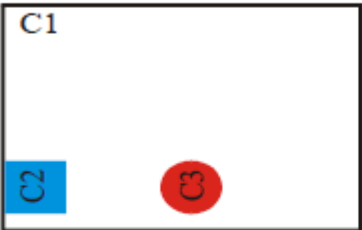
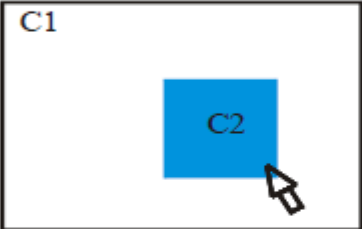
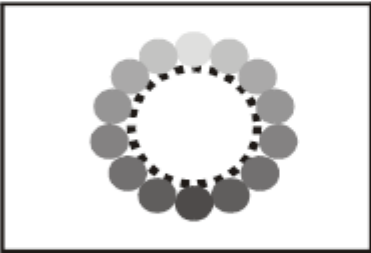
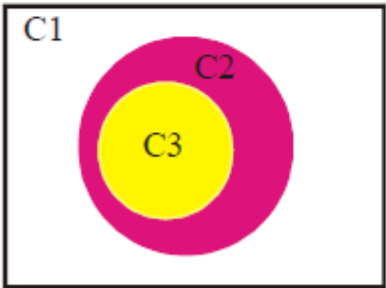
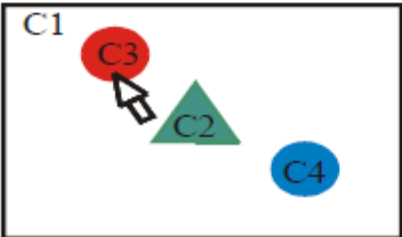
Переменные C1, C2, C3... и т.д. обозначают цвета фигур и участков экрана. Если индексы в переменных разные, то и цвета должны быть разные.

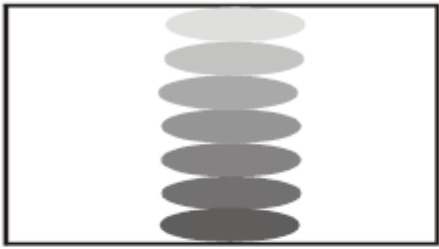
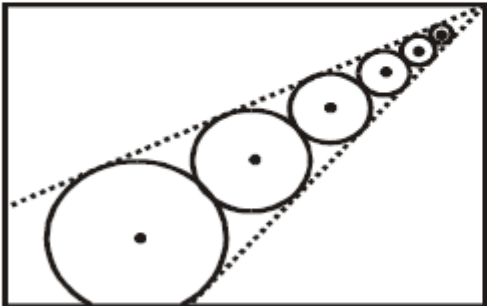
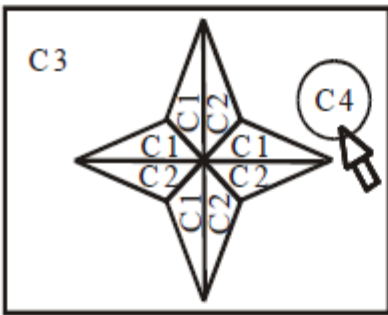

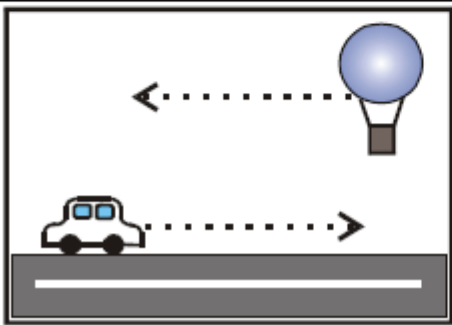
Вывод всех рисунков на экран осуществляется методом двойной буферизации.

№	Рисунок	Пояснения
1		По экрану по показанной траектории движется круг, изменяя свой радиус.
2		По экрану по показанной траектории движется круг, изменяя свой радиус и цвет.
3		На экране цвета C3 через $\Delta t_1$ появляется фигура цвета C2, а через $\Delta t_2$ фигура цвета C1.
4		Фигуры из каждого сектора экрана через $\Delta t_1$ после запуска программы начинают двигаться вверх.

№	Рисунок	Пояснения
5		Через $\Delta t_1$ после запуска программы на экране получить данную картинку.
6		По экрану по показанной траектории движется круг, изменяя свой радиус и цвет.
7		В программе имитировать восход солнца.
8		В программе имитировать приближение автомобиля к плоскости экрана.
9		Фигуры появляются на экране друг за другом последовательно через $\Delta t_1$ .



№	Рисунок	Пояснения
10		<p>На экране имитировать движение маятника.</p> <p>Левая кнопка мыши колебания начинаются, правая – заканчиваются.</p>
11		<p>На экране имитировать движение маятника.</p> <p>Правая кнопка мыши колебания начинаются, левая – заканчиваются.</p>
12		<p>Фигура появляется на экране через <math>\Delta t_1</math>.</p>
13		<p>По траектории показанной на рисунке движется круг, изменяя свой цвет в зависимости от своего местоположения.</p>
14		<p>На экране цвета C1 через <math>\Delta t_1</math> появляется круг цвета C2, а через <math>\Delta t_2</math> круг цвета C3, который начинает хаотично двигаться внутри другого круга.</p>
15		<p>Фигуры появляются на экране друг за другом последовательно через <math>\Delta t_1</math>.</p>

№	Рисунок	Пояснения
16		По экрану по показанной траектории движется эллипс, изменяя свой цвет.
17		По экрану по показанной траектории движется круг, изменяя свой радиус.
18		На экране получить разноцветную фигуру, по которой может двигаться круг.
19		На экране симитировать мерцание звезд на ночном небе
20		На экране проиллюстрировать одновременное движение автомобиля и полет воздушного шара в указанных направлениях. Предметы начинают движение по левому щелчку мыши.

**Порядок выполнения работы согласно методическим рекомендациям по практическим работам по МДК 01.04 Системное программирование**

## Самостоятельная работа 1

### Тема: Подсистемы управления ресурсами

**Задание:** подготовка доклада, реферата, презентации и т.д.

#### **Задание 1.**

1) Дать основные понятия и определения:

- процессоры,
- основная память,
- таймеры,
- наборы данных,
- накопители информации и внешние устройства,
- принтеры,
- сетевые устройства,
- пользовательский интерфейс.

2) Управление процессами

3) Состояние регистров и программный счетчик процессора.

5) Режимы работы процессора.

6) Указатели на открытые файлы.

7) Информация о незавершенных операциях ввода-вывода.

8) Коды ошибок выполняемых процессором системных вызовов.

#### **Задание 2.**

1) Управление памятью.

2) Распределение физической памяти между всеми существующими в системе процессами.

3) Загрузка кодов и данных процессов в отдельные области памяти.

4) Настройки адресно-зависимых частей кодов процесса на физические адреса выделенной области.

5) Защита областей памяти каждого процесса.

6) Управление файлами

7). Действия операционной системы по управлению фалами и каталогами.

8) Защита данных и администрирование.

**Порядок выполнения работы согласно методическим рекомендациям по самостоятельным работам по МДК 01.04 Системное программирование**

## РУБЕЖНЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Ресурсы принадлежащие каждому потоку в ОС Windows:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. потоки интерфейса пользователя
- B. пользовательские потоки
- C. код исполняемой функции
- D. потоки исполняемой функции
- E. набор регистров процессора
- F. библиотека общих элементов
- G. интерфейс пользователя
- H. стек для работы приложения

2. Действия менеджера потоков, во время переключения процессора на исполнение другого потока:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. обслуживает очередь запускаемого потока
- B. сохранить контекст прерываемого потока
- C. восстановить контекст запускаемого потока
- D. передать управление запускаемому потоку
- E. управляет доступом для ОС
- F. выполняет функции ввода и вывода
- G. обеспечивает функции для вывода графики
- H. обеспечивает функции для взаимодействия

3. Параметры функции CreateThread:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. TerminateThread
- B. lpThreadAttributes
- C. AttachThreadInput
- D. CreateProcess
- E. CreateThread
- F. ExitThread
- G. dwStackSize
- H. lpStrtAddress

4. Функции работы с процессами и потоками в Win32 API:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. TerminateThread
- B. lpThreadAttributes
- C. AttachThreadInput
- D. CreateProcess
- E. CreateThread
- F. ExitThread
- G. dwStackSize
- H. lpStrtAddress

5. Функция Sleep()\_\_\_\_\_:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. задерживает выполнения потока
- B. возвращает нулевое значение
- C. удаляет поток из очереди
- D. выполняет выход из потока
- E. освобождает память
- F. делает короткие паузы
- G. завершает выполнение функций
- H. освобождает ресурсы

6. Классификация программ в зависимости от количества определяемых ими параллельных потоков управления:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. дуплексный
- B. многозадачный
- C. многопоточный
- D. системный
- E. с параллельным потоком
- F. параллельный
- G. однопоточный
- H. с прямым потоком

7. Операции над потоком связанные с операционной системой:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. Create
- B. Run
- C. Exit
- D. Interrupt
- E. Open
- F. Block

- G. Read
- H. Write

8. Ресурсы каждого процесса ОС Windows:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. виртуальное адресное пространство
- B. код исполняемой функции
- C. маркер доступа
- D. набор регистров процессора
- E. стек для работы приложения
- F. страницы в реальной памяти
- G. стек для работы ОС
- H. консольный ввод

9. Объектам синхронизации первого класса в Windows, которые служат только для решения задач синхронизации параллельных потоков:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. мьютекс (mutex)
- B. событие (event)
- C. семафор (semaphore)
- D. ожидающий таймер
- E. работа (job)
- F. процесс (process)
- G. поток (thread)
- H. консольный ввод (console input)

10. К третьему классу синхронизации относятся объекты, которые переходят в сигнальное состояние по завершении своей работы:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. мьютекс (mutex)
- B. событие (event)
- C. работа (job)
- D. процесс (process)
- E. поток (thread)
- F. семафор (semaphore)
- G. ожидающий таймер
- H. консольный ввод (console input)

11. Системные объекты, созданные менеджером объектов в Win32 API:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. класс
- B. поток
- C. события
- D. семафор
- E. процесс
- F. критический раздел
- G. приоритет потока
- H. контекст

12. Логические комбинации флагов параметра dwDesiredAccess:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. EVENT\_ALL\_ACCESS
- B. EVENT\_MODIFY\_STATE
- C. ACTIONS\_AFTER\_EVENT
- D. SYNCHRONIZE
- E. SEMAPHORE\_ALL\_ACCESS
- F. SEMAPHORE\_MODIFY\_STATE
- G. ACTIONS\_BEFORE\_EVENT
- H. CREATE\_NEW\_CONSOLE

13. Логические комбинации флагов параметра dwDesiredAccess, определяющий доступ к семафору:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. EVENT\_ALL\_ACCESS
- B. EVENT\_MODIFY\_STATE
- C. SYNCHRONIZE
- D. SEMAPHORE\_ALL\_ACCESS
- E. ACTIONS\_AFTER\_EVENT
- F. SEMAPHORE\_MODIFY\_STATE
- G. ACTIONS\_BEFORE\_EVENT
- H. CREATE\_NEW\_CONSOLE

14. Значение функции WaitForSingleObject, в случае успешного завершения:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. wait\_object\_p
- B. wait\_forsingle
- C. wait\_object\_o
- D. wait\_object
- E. wait\_abandoned
- F. wait\_timeout

- G. wait\_exit
- H. wait\_finally

15. Прототип функции WaitForMultipleObject:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. DWORD nCount
- B. CONST HANDLE \*lpHandles
- C. BOOL bWaitAll
- D. DWORD dwReserved
- E. BOOL bResume
- F. LPOVERLAPPED lpOverLapped
- G. HANDLE hThread
- H. DWORD IDThread

16. Значения функции WaitForMultipleObjects:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. ОТ WAIT\_OBJECT\_0 ДО (WAIT\_OBJECT\_0 + nCount - 1);
- B. ОТ WAIT\_OBJECT\_0 ДО (WAIT\_OBJECT\_0 );
- C. ОТ WAIT\_OBJECT\_0 ДО (WAIT\_OBJECT\_0 + nCount);
- D. ОТ WAIT\_ABANDONED\_0 ДО (WAIT\_ABANDONED\_0 + nCount - 1);
- E. WAIT\_COMPLETION
- F. WAIT\_TIMEOUT
- G. ОТ WAIT\_ABANDONED\_0 ДО (WAIT\_ABANDONED\_0 + nCount + 1);
- H. ОТ WAIT\_ABANDONED\_0 ДО (WAIT\_ABANDONED\_0)

17. Состояние блока адресов в адресном пространстве:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. выделен
- B. защищен
- C. зарезвирован
- D. свободен
- E. смещен
- F. удален
- G. копирован
- H. прикреплен

18. Интерфейсы (API) для управления памятью:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.



- A. Base Services
- B. Common Control Library
- C. Virtual Memory
- D. Network Services
- E. Memory Mapped File
- F. Heap Memory
- G. Windows Shell
- H. Windows System Information

19. Куча – это \_\_\_\_\_:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. объекты памяти
- B. страница памяти
- C. динамическая область памяти
- D. указатель блока памяти
- E. блок памяти
- F. указатель строки
- G. мелкие фрагменты памяти
- H. указатель функции

20. Функции для управления памятью кучи:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. HeapAlloc()
- B. HeapDestroy()
- C. HeapReAlloc()
- D. HeapFree()
- E. HeapSize()
- F. HeapFree()
- G. HeapCreate()
- H. MapViewOfFile()

21. Форматы реального и виртуального адресов:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. номер реальной страницы
- B. номер фактической страницы
- C. номер оперативной памяти
- D. номер виртуальной страницы
- E. смещение в адресной строке
- F. смещение в реальной и виртуальной странице
- G. файлы страницы
- H. номер динамической страницы

22. Форматы реального и виртуального адресов:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. a
- B. r
- C. b
- D. v
- E. c
- F. d
- G. e
- H. k

23. Описание линейного адреса процесса в ОС Windows:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. 32 бит
- B. 16 бит
- C. от 0x00000000 до 0xFFFFFFFF
- D. 2 Гбайт логической памяти
- E. 4 Гбайт виртуальной памяти
- F. 2 Гбайт виртуальной памяти
- G. 4 Гбайт логической памяти
- H. от 0x00000000 до 0x0000FFFF

24. Состояние страницы процесса виртуальной памяти:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. (блокирован);
- B. (выделен)
- C. free (свободный)
- D. (не выделен)
- E. committed (распределены)
- F. (готов)
- G. (не готов)
- H. reserved (зарезервированный)

25. Функция файловой системы:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. совместно использовать объект файла
- B. открытие доступа к существующему файлу

- С. выделяют мелкие фрагменты файла
- D. закрытие доступа к существующему файлу
- Е. закрепление виртуальной памяти
- Г. установка указателя файла на нужную запись
- Г. физическое или форматирование низкого уровня
- Н. разбиение диска на разделы

### Критерии оценивания:

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	менее 50

### Ключ ответов к тестовым заданиям

№ Вопроса	Правильный ответ	Проверяемые компетенции
1.	С,Е,Н	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
2.	В,С,D	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
3.	В,Г,Н	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
4.	С,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
5.	А,С,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
6.	С,Е,Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
7.	В,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
8.	А,С,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
9.	А,В,С	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
10.	С,D,E	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
11.	С,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
12.	А,В,D	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
13.	С,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.

14.	C,E,F	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
15.	A,B,C	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
16.	A,D,F	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
17.	A,C,D	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
18.	C,E,F	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
19.	C,E,G	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
20.	A,C,F	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
21.	A,D,F	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
22.	B,D,F	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
23.	A,C,G	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
24.	C,E,H	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.
25.	B,D,F	OK 01, OK 02, OK 03, OK 04, OK 05, OK 06, OK 07, OK 08, OK 09, OK 10, ПК 1.2, ПК 1.3.

## МЕЖСЕССИОННЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Значение функций GetFileType:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. FILE\_SHARE\_WRITE
- B. FILE\_SHARE\_READ
- C. GENERIC\_READ
- D. FILE\_TYPE\_DISK
- E. FILE\_TYPE\_CHAR
- F. TRUNCATE\_EXISTING
- G. FILE\_TYPE\_PIPE
- H. OPEN\_ALWAYS

2. Значения параметра функции CopyFile:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. lpExistingFileName
- B. lpThreadAttributes
- C. AttachThreadInput
- D. lpNewFileName
- E. dwStackSize
- F. lpStrtAddress
- G. bFailIfExists
- H. TerminateThread

3. Верхние уровни, составляющие основу структуры реестра Windows:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. HKEY\_LOCAL
- B. HKEY\_LOCAL\_WORD
- C. HKEY\_LOCAL\_MACHINE
- D. HKEY\_CURRENT\_USER
- E. HKEY\_CLASSES\_ROOT
- F. HKEY\_CURRENT\_MACHINE
- G. HKEY\_CLASSES\_MACHINE
- H. HKEY\_CLASSES\_USER

4. API функции, применяемые для выполнения операций с реестром:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. RegReserverd
- B. RegCloseKey
- C. RegClass
- D. RegSubKeys
- E. RegCreateKey
- F. RegOpen
- G. RegQery
- H. RegDeleteKey

5. API функции применяемые в среде Windows NT:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. RegQueryValue()
- B. RegSetKeySecurity()
- C. RegCloseKey()
- D. RegFlushKey()
- E. RegGetKeySecurity()
- F. RegLoadKey()
- G. параметры безопасности
- H. RegOpenKey()

6. HKey идентифицирует текущий раздел или predetermined дескрипторы:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. HKEY\_LOCAL\_MACHINE
- B. HKEY\_LOCAL\_WORD
- C. HKEY\_CLASSES\_ROOT
- D. HKEY\_CURRENT\_MACHINE
- E. HKEY\_CLASSES\_USER
- F. HKEY\_CURRENT\_USER
- G. HKEY\_CLASSES\_MACHINE
- H. HKEY\_USERS

7. Параметры функции WriteFile:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. nNumberOfBytes
- B. lpThreadAttributes
- C. hFile
- D. lpBuffer
- E. dwStackSize
- F. lpNewFileName
- G. bFailIfExists
- H. nNumberOfBytesToWrite

8. Системный вызов для файла:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. chown()
- B. lchown()
- C. chmod
- D. fchmod
- E. stat()
- F. lstat()
- G. fchown()
- H. read()

9. Метаданные файла связанные со временем:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. st\_blocks
- B. st\_dev
- C. st\_ctime
- D. st\_uid
- E. st\_atime
- F. st\_gid
- G. st\_size
- H. st\_mtime

10. Функции для перемещения файла:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. lpNewFileName
- B. replaceFile
- C. readFile
- D. copyFile
- E. hFile
- F. moveFile
- G. write\_File
- H. delete\_File

11. Механизм отображения файлов в память:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. отображение содержимого файла
- B. установка времени таймера
- C. представление или вид файла

- D. когерентность данных
- E. указывать на функцию завершения
- F. создать файловое пространство
- G. открыть файл приложения
- H. создать файл загрузки

12. Параметры функции CreateFileMapping (создание объекта отображающий файл в память):

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. lpBuffer
- B. numberOfBytes
- C. completionKey
- D. flProtect
- E. readFile
- F. lpAttributes
- G. waitCommEvent
- H. hFile

13. Значения параметра flProtect :

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. PAGE\_READ
- B. PAGE\_READONLY
- C. PAGE\_WRITEONLY
- D. PAGE\_READWRITE
- E. PAGE\_READCOPY
- F. PAGE\_WRITECOPY
- G. PAGE\_WRITE
- H. PAGE\_COPYONLY

14. Прототип функции MapViewOfFile:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. HANDLE hFileMappingObject
- B. DWORD dwFileOffsetHigh
- C. DWORD dwFileOffsetLow
- D. DWORD dwReserved
- E. BOOL bResume
- F. LPOVERLAPPED lpOverLapped
- G. HANDLE hThread
- H. DWORD IDThread



15. Прототип функции MapViewOfFileEx, отображающий файл в адресное пространство с некоторого заданного виртуального адреса:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. HANDLE hFileMappingObject
- B. DWORD dwReserved
- C. DWORD DesiredAccess
- D. LPVOID lpBaseAddress
- E. BOOL bResume
- F. LPOVERLAPPED lpOverLapped
- G. HANDLE hThread
- H. DWORD IDThread

16. Механизм отображения файлов в память:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. узнать какой поток ее вызывает
- B. файл отображен несколькими процессами
- C. хранить указатели на захваченную память
- D. запускать приложение
- E. когерентность отображений
- F. обмен данными между процессами
- G. открыть файл приложения
- H. создать файл загрузки

17. Функции файла stdio.h языка программирования C, создающие стандартную библиотеку ввода-вывода:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. stdout— файл ввода
- B. stlib —файлы библиотеки
- C. stdin — стандартный файл ввода
- D. tanh —математикалық шамаларды шығаратын файл
- E. stdout — стандартный файл вывода
- F. strcmp —функция сравнения
- G. stderr —файл вывода сообщения об ошибке
- H. main— программаны орындауды бастайтын файл

18. Прототип функцииDllMain:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. HINSTANCE
- B. DWORD

- C. LPVOID
- D. HMODULE
- E. LPCTSTR
- F. HANDLE
- G. DWORD
- H. LPCSTR

19. Параметр `fdwReason` может иметь одно из следующих значений, которое указывает на причину, по которой операционная система вызывает функцию `DllMain`:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. `dont_resolve_dll_references`
- B. `load_library_as_datafile`
- C. `load_with_altered_search_path`
- D. `dll_process_attach`
- E. `dll_thread_attach`
- F. `dll_process_detach`
- G. `case dll_process_attach`
- H. `dwPrewTlsIndex= dwFirstTlsIndex`

20. Прототип функции `LoadLibraryEx`, для загрузки динамически подключаемых библиотек:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. LPCTSTR
- B. HANDLE
- C. DWORD
- D. HINSTANCE
- E. DWORD\_B
- F. LPVOID
- G. HMODULE
- H. LPCSTR

21. Значения параметра `dwFlags`, задающий флаги управления загрузкой модуля:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. `dont_resolve_dll_references`
- B. `load_library_as_datafile`
- C. `load_with_altered_search_path`
- D. `dll_process_attach`
- E. `dll_thread_attach`
- F. `dll_process_detach`

- G. case dll\_process\_attach
- H. dwPrewTlsIndex= dwFirstTlsIndex

22. Действия необходимые для статической загрузки DLL:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. создать файл
- B. поместить библиотеку и файл в каталог
- C. хранить указатели на захваченную память
- D. запускать приложение
- E. узнать какой поток ее вызывает
- F. ввести имя используемой библиотеки импорта
- G. сопоставить каждому потоку свой указатель
- H. описать импортируемые из DLL имена в приложении

23. Задачи динамической локальной памяти потока:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. создать DLL
- B. поместить библиотеку в каталог
- C. хранить указатели на захваченную память
- D. поместить файл импорта этой библиотеки в каталог
- E. узнать какой поток ее вызывает
- F. ввести имя используемой библиотеки импорта
- G. сопоставить каждому потоку свой указатель
- H. описать импортируемые из DLL имена в приложении

24. Порядок работы с локальной памятью потока:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. распределение указателя
- B. сохранить указатель
- C. использовать указатель
- D. работа с указателем
- E. освобождение указателя
- F. завершение функции
- G. возвращает значение
- H. запись значения

25. Динамические подключаемые библиотеки предназначены \_\_\_\_\_ :

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. для разработки функционально-замкнутых библиотек функций

- В. для снижения затрат на разработку ПО
- С. для открытия файла, который будет отображаться в динамической памяти
- Д. для загрузки в адресное пространство процесса
- Е. для завершения нового потока в процессе
- Ф. для сохранения файла и каталога
- Г. для уменьшения физической памяти
- Н. для перехода в адресную строку

### Критерии оценивания:

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	менее 50

### Ключ ответов к тестовым заданиям

№ Вопроса	Правильный ответ	Проверяемые компетенции
1.	D,E,G	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
2.	A,D,G	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
3.	C,D,E	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
4.	B,E,H	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
5.	B,E,G	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
6.	C,F,H	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
7.	C,D,H	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
8.	A,B,G	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
9.	C,E,H	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
10.	B,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
11.	A,C,D	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
12.	D,F,H	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК

		07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
13.	B,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
14.	A,B,C	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
15.	A,C,D	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
16.	B,E,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
17.	C,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
18.	A,B,C	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
19.	D,E,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
20.	A,B,C	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
21.	A,B,C	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
22.	B,F,H	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
23.	C,E,G	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
24.	A,D,E	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
25.	A,B,G	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.

## ИТОГОВЫЙ КОНТРОЛЬ (ТЕСТИРОВАНИЕ)

1. Структура и описание каталога:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. древовидная
- B. кольцевая
- C. корневой
- D. табличная
- E. \ (обратная косая)
- F. прямая
- G. выпуклая
- H. плоская

2. Имена каталогов и файлов не должны содержать:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. +
- B. <
- C. \*
- D. :
- E. /
- F. ?
- G. !
- H. .

3. Параметр `dwDesiredAccess` задает способ доступа к файлу и принимает значения:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. 0
- B. `FILE_SHARE_WRITE`
- C. `FILE_SHARE_READ`
- D. `GENERIC_READ`
- E. `GENERIC_WRITE`
- F. `OPEN_EXISTING`
- G. `OPEN_ALWAYS`
- H. `TRUNCATE_EXISTING`

4. Значения параметра `dwCreationDisposition` при открытии файла:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. 0
- B. FILE\_SHARE\_WRITE
- C. FILE\_SHARE\_READ
- D. GENERIC\_READ
- E. GENERIC\_WRITE
- F. OPEN\_EXISTING
- G. OPEN\_ALWAYS
- H. TRUNCATE\_EXISTING

5. Правильная запись имени файла:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. "C:\\demo\_file.dat"
- B. "C:\\demo\_file.dot"
- C. "C:\\demo\\file.dat"
- D. "C\\demo\_file.dat"
- E. "C:\\new\_file.dat"
- F. "C:\\ new\_file.dat"
- G. "C:\\back\_file.dat"
- H. "\\back\_file.dat"

6. Функция для работы локальной памятью потока:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. для разработки функционально-замкнутых библиотек функций
- B. для снижения затрат на разработку ПО
- C. для открытия файла, который будет отображаться в динамической памяти
- D. для загрузки в адресное пространство процесса
- E. для завершения нового потока в процессе
- F. для сохранения файла и каталога
- G. для уменьшения физической памяти
- H. для перехода в адресную строку

7. Параметры функции CreateFileMapping (создание объекта отображающий файл в память):

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. TlsAlloc
- B. LpBuffer
- C. TlsSetValue
- D. TlsGetValue

- E. CompletionKey
- F. ReadFile
- G. NumberOfBytes
- H. lpAttributes

8. Распределение локальной памяти потока в DLL:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. case DLL\_PROCESS\_ATTACH
- B. dwPrewTlsIndex= dwFirstTlsIndex
- C. dwPrewTlsIndex= dwNextTlsIndex
- D. dwlsIndex=TlsAlloc(); If (dwlsIndex ==-1)
- E. break;
- F. return 0;
- G. dDll=LoadLibrary;
- H. returnGetLastError();

9. Статическая локальная память:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. оповещает параллельные потоки
- B. использует спецификатор памяти declspec(thread)
- C. определяет локальные переменные
- D. устанавливает соединения между потоками данных
- E. принимает записи
- F. содержит очередь пакетов
- G. обслуживает очередь пакетов
- H. создает отдельный переменный

10. Значения аргумента dwCreationDisposition для создания файла:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. CREATE\_FILE
- B. CREATE\_THREAD
- C. CREATE\_NEW
- D. OPEN
- E. CREATE\_ALWAYS
- F. CREATE
- G. OPEN\_ALWAYS
- H. CREATE\_NEWFILE



11. Системные объекты, созданные менеджером объектов в Win32 API:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. класс
- B. поток
- C. события
- D. семафор
- E. процесс
- F. критический раздел
- G. приоритет потока
- H. контекст

12. Логические комбинации флагов параметра dwDesiredAccess:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. EVENT\_ALL\_ACCESS
- B. EVENT\_MODIFY\_STATE
- C. ACTIONS\_AFTER\_EVENT
- D. SYNCHRONIZE
- E. SEMAPHORE\_ALL\_ACCESS
- F. SEMAPHORE\_MODIFY\_STATE
- G. ACTIONS\_BEFORE\_EVENT
- H. CREATE\_NEW\_CONSOLE

13. Логические комбинации флагов параметра dwDesiredAccess, определяющий доступ к семафору:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. EVENT\_ALL\_ACCESS
- B. EVENT\_MODIFY\_STATE
- C. SYNCHRONIZE
- D. SEMAPHORE\_ALL\_ACCESS
- E. ACTIONS\_AFTER\_EVENT
- F. SEMAPHORE\_MODIFY\_STATE
- G. ACTIONS\_BEFORE\_EVENT
- H. CREATE\_NEW\_CONSOLE

14. Значение функции WaitForSingleObject, в случае успешного завершения:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. wait\_object\_p
- B. wait\_forsingle
- C. wait\_object\_o

- D. wait\_object
- E. wait\_abandoned
- F. wait\_timeout
- G. wait\_exit
- H. wait\_finally

15. Прототип функции WaitForMultipleObject:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. DWORD nCount
- B. CONST HANDLE \*lpHandles
- C. BOOL bWaitAll
- D. DWORD dwReserved
- E. BOOL bResume
- F. LPOVERLAPPED lpOverLapped
- G. HANDLE hThread
- H. DWORD IDThread

16. Значения функции WaitForMultipleObjects:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. ОТ WAIT\_OBJECT\_0 ДО (WAIT\_OBJECT\_0 + nCount - 1);
- B. ОТ WAIT\_OBJECT\_0 ДО (WAIT\_OBJECT\_0 );
- C. ОТ WAIT\_OBJECT\_0 ДО (WAIT\_OBJECT\_0 + nCount);
- D. ОТ WAIT\_ABANDONED\_0 ДО (WAIT\_ABANDONED\_0 + nCount - 1);
- E. WAIT\_COMPLETION
- F. WAIT\_TIMEOUT
- G. ОТ WAIT\_ABANDONED\_0 ДО (WAIT\_ABANDONED\_0 + nCount + 1);
- H. ОТ WAIT\_ABANDONED\_0 ДО (WAIT\_ABANDONED\_0)

17. Состояние блока адресов в адресном пространстве:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. выделен
- B. защищен
- C. зарезвирован
- D. свободен
- E. смещен
- F. удален
- G. копирован
- H. прикреплен

18. Интерфейсы (API) для управления памятью:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. Base Services
- B. Common Control Library
- C. Virtual Memory
- D. Network Services
- E. Memory Mapped File
- F. Heap Memory
- G. Windows Shell
- H. Windows System Information

19. Куча – это \_\_\_\_\_:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. объекты памяти
- B. страница памяти
- C. динамическая область памяти
- D. указатель блока памяти
- E. блок памяти
- F. указатель строки
- G. мелкие фрагменты памяти
- H. указатель функции

20. Функции для управления памятью кучи:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. HeapAlloc()
- B. HeapDestroy()
- C. HeapReAlloc()
- D. HeapFree()
- E. HeapSize()
- F. HeapFree()
- G. HeapCreate()
- H. MapViewOfFile()

21. Форматы реального и виртуального адресов:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. номер реальной страницы
- B. номер фактической страницы
- C. номер оперативной памяти
- D. номер виртуальной страницы
- E. смещение в адресной строке
- F. смещение в реальной и виртуальной странице

- G. файлы страницы
- H. номер динамической страницы

22. Форматы реального и виртуального адресов:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. a
- B. r
- C. b
- D. v
- E. c
- F. d
- G. e
- H. k

23. Описание линейного адреса процесса в ОС Windows:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. 32 бит
- B. 16 бит
- C. от 0x00000000 до 0xFFFFFFFF
- D. 2 Гбайт логической памяти
- E. 4 Гбайт виртуальной памяти
- F. 2 Гбайт виртуальной памяти
- G. 4 Гбайт логической памяти
- H. от 0x00000000 до 0x0000FFFF

24. Состояние страницы процесса виртуальной памяти:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- A. (блокирован);
- B. (выделен)
- C. free (свободный)
- D. (не выделен)
- E. committed (распределены)
- F. (готов)
- G. (не готов)
- H. reserved (зарезервированный)

25. Функция файловой системы:

**Проверяемые компетенции:** ПК 1.2, ПК 1.3.

- А. совместно использовать объект файла
- В. открытие доступа к существующему файлу
- С. выделяют мелкие фрагменты файла
- Д. закрытие доступа к существующему файлу
- Е. закрепление виртуальной памяти
- Г. установка указателя файла на нужную запись
- Г. физическое или форматирование низкого уровня
- Н. разбиение диска на разделы

### Критерии оценивания:

Отметка	Объем выполнения работы в %
«5» (отлично)	90 – 100
«4» (хорошо)	70 – 89
«3» (удовлетворительно)	50 – 69
«2» (неудовлетворительно)	менее 50

### Ключ ответов к тестовым заданиям

№ Вопроса	Правильный ответ	Проверяемые компетенции
1.	А,С,Е	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
2.	В,Д,Е	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
3.	А,Д,Е	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
4.	Г,Н	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
5.	А,Е,Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
6.	А,С,Д	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
7.	Д,Г,Н	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
8.	А,Д,Е	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
9.	В,С,Н	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
10.	С,Е,Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
11.	С,Д,Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
12.	А,В,Д	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
13.	С,Д,Г	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК

		07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
14.	C,E,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
15.	A,B,C	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
16.	A,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
17.	A,C,D	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
18.	C,E,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
19.	C,E,G	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
20.	A,C,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
21.	A,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
22.	B,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
23.	A,C,G	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
24.	C,E,H	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.
25.	B,D,F	ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 1.2, ПК 1.3.

### Варианты заданий для промежуточной аттестации

1. Подсистемы управления ресурсами.
2. Управление процессами.
3. Управление потоками.
4. Параллельная обработка потоков. Создание процессов и потоков.
5. Обмен данными между процессами. Передача сообщений.
6. Анонимные и именованные каналы.
7. Сетевое программирование сокетов.
8. Динамически подключаемые библиотеки DLL.
9. Сервисы. Виртуальная память. Выделение памяти процессам.
10. Работа с буфером экрана.
11. Что представляет собой язык программирования Ассемблер.
12. Виды программного обеспечения: системное, прикладное и промежуточное (middleware) программное обеспечение.
13. Тенденции развития программного обеспечения.

14.Разработка программного обеспечения. Этапы разработки программного обеспечения.

15.Системное программирование – основные определения.

16.Команды языка ассемблера. Дизассемблирование кода. Анализ кода.

17.Виды программного обеспечения.

18.Работа с middleware.

19.Основные функции middleware

20.Работа с графическими интерфейсами.

21.Содержательная постановка задачи.

22.Разработка модели и выбор метода решения.

23.Кодирование алгоритма.

24.Компиляция программы.

25.Тестирование программы.

26.Сопровождение программы.

27.Создание документации.

28.Мнемоника языка Ассемблера.

29.Операнды языка Ассемблера.

30.Комментарии языка Ассемблера.

31.Составление программы на языке Ассемблера.