

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Саратовский государственный технический

университет имени Гагарина Ю.А.»

Энгельсский технологический институт (филиал)



УТВЕРЖДАЮ

Зам. директора по СПДО

О.Г. Коваленко

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

учебной дисциплины

ОП.04 Основы алгоритмизации и программирования

по специальности:

09.02.07 Информационные системы и программирование

Методические указания
рассмотрены на заседании
предметной (цикловой) методической комиссии
специальности 09.02.07
«25» июня 2024 года, протокол № 11

Председатель ПЦМК  А.А. Сдобнова

Энгельс 2024

Пояснительная записка

В результате освоения учебной дисциплины ОП.04 «Основы алгоритмизации и программирования» обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование следующими умениями, знаниями, общими компетенциями, профессиональными компетенциями:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Общие и профессиональные 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 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 09. Пользоваться профессиональной документацией на государственном языке.

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

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

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

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

В ФОС по учебной дисциплине ОП.04 «Основы алгоритмизации и программирования» включены:

1) текущий контроль успеваемости:

- входной контроль знаний;
- рубежный контроль успеваемости;
- межсессионную аттестацию.

2) промежуточная аттестация.

Экзамен – 4 семестр

Информационное обеспечение реализации программы

Печатные издания

Электронные издания (электронные ресурсы)

1. Кудрина, Е. В. Основы алгоритмизации и программирования на языке JAVA : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2023. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/517324>
2. Казанский, А. А. Программирование на Visual JAVA : учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 192 с. — (Профессиональное образование). — ISBN 978-5-534-14130-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513400>
3. Подбельский, В. В. Программирование. Базовый курс C# : учебник для среднего профессионального образования / В. В. Подбельский. — Москва : Издательство Юрайт, 2023. — 369 с. — (Профессиональное образование). — ISBN 978-5-534-11467-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/517893>
4. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — 4-е изд. — Москва : Издательство Юрайт, 2023. — 119 с. — (Профессиональное образование). — ISBN 978-5-534-17498-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/533200>

Интернет-ресурсы

1. Справочник по WPF - <http://msdn.microsoft.com/ru-ru/library/ms754130.aspx>
2. Справочник по базам данных - <http://msdn.microsoft.com/ru-ru/library/h0y4a0f6.aspx>
3. Пошаговые руководства по JAVA - <http://msdn.microsoft.com/ruru/library/1dbsh6t3.aspx>
4. Учебные руководства - <http://msdn.microsoft.com/ru-ru/library/dd492171.aspx>

Электронно-библиотечные системы:

1. «ЭБС IPRbooks», ООО «Ай Пи Эр Медиа»
2. ЭБС «Электронная библиотека технического вуза», ООО «Политехресурс»
3. ЭБС «Лань», ООО «Издательство Лань»
4. «ЭБС elibrary», ООО «РУНЭБ»
5. ЭБС «ЮРАЙТ»
6. ЭБС «Book.ru»

Входной контроль

Входной контроль проводится в форме:

- выполнение письменной работы.

Инструкция по выполнению работы

На выполнение работы отводится 20 минут для письменного ответа на поставленные вопросы.

Варианты вопросов

- Что такое язык программирования?
- Из чего составит алфавит языка программирования?
- Что такое программа? Опишите структуру программы.
- Что такое компиляторы и интерпретаторы?
- Что такое алгоритм?
- Перечислите свойства алгоритмов
-

Критерии оценивания:

Оценка 5 «отлично» теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

Оценка 4 «хорошо» теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

Оценка 3 «удовлетворительно» теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

Оценка 2 «неудовлетворительно» теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Рубежный контроль

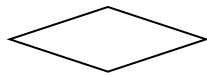
Раздел 1. Введение в программирование

Форма контроля: тестирование

Время выполнения: 20 минут

1. Определенный алгоритм должен быть применим ко всем однотипным задачам – это свойство алгоритма называется:
Проверяемые компетенции: ОК 01, ПК 1.1
 - a) Массовость
 - b) Дискретность
 - c) Детерминированность
 - d) Результативность
2. Работа алгоритма должна завершаться за определенное число шагов, при этом задача должна быть решена – это свойство алгоритма называется:
Проверяемые компетенции: ОК 01, ПК 1.1
 - a) Дискретность
 - b) Массовость
 - c) Результативность и конечность
 - d) Детерминированность
3. Алгоритм должен состоять из отдельных действий, которые выполняются последовательно друг за другом – это свойство алгоритма называется:
Проверяемые компетенции: ОК 01, ПК 1.1
 - a) Детерминированность
 - b) Массовость
 - c) Дискретность
 - d) Результативность и конечность
4. Многократное применение одного алгоритма к одному и тому же набору исходных данных всегда дает один и тот же результат – это свойство алгоритма называется:
Проверяемые компетенции: ОК 01, ПК 1.1
 - a) Однозначная определённость (детерминированность)
 - b) Массовость
 - c) Дискретность
 - d) Результативность и конечность
5. Что не является алгоритмической структурой:
Проверяемые компетенции: ОК 01, ПК 1.1
 - a) массив
 - b) следование
 - c) ветвление
 - d) цикл
6. Графическое изображение логической структуры алгоритма, в котором каждый этап процесса обработки информации представляется в виде геометрических символов (блоков) называется...
Проверяемые компетенции: ОК 01, ПК 1.1
 - a) графиком
 - b) блок-схемой
 - c) диаграммой
 - d) блочной схемой

7. Как называется элемент блок-схемы:
Проверяемые компетенции: ОК 01, ПК 1.1



- a) выполнение операции
- b) ввод-вывод данных
- c) условие
- d) начало-конец алгоритма

8. Как называется элемент блок-схемы:
Проверяемые компетенции: ОК 01, ПК 1.1



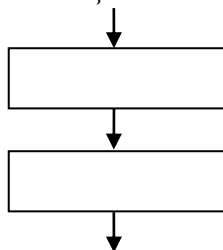
- a) условие
- b) выполнение операции
- c) начало-конец алгоритма
- d) ввод-вывод данных

9. Как называется элемент блок-схемы:
Проверяемые компетенции: ОК 01, ПК 1.1



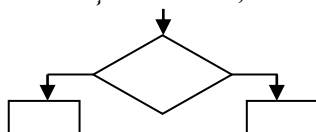
- a) начало-конец алгоритма
- b) выполнение операции
- c) условие
- d) ввод-вывод данных

10. Как называется представленная на рисунке часть блок-схемы:
Проверяемые компетенции: ОК 01, ПК 1.1



- a) ввод данных
- b) следование
- c) инициализация
- d) итерация

11. Как называется представленная на рисунке часть блок-схемы:
Проверяемые компетенции: ОК 01, ПК 1.1



- a) цикл
- b) ввод данных
- c) ветвление
- d) итерация

12. Какое из перечисленных понятий не является свойством алгоритма:

Проверяемые компетенции: ОК 01, ПК 1.1

- a) Цикличность
- b) Массовость
- c) Дискретность
- d) Результативность и конечность

13. Упорядоченная последовательность команд (инструкций) компьютера для решения задачи - это:

Проверяемые компетенции: ОК 01, ПК 1.1

- a) программа
- b) язык программирования
- c) приложение
- d) задача

14. Соотнести

Проверяемые компетенции: ОК 01, ПК 1.1

1. Если выражение-условие возвращает true, то выполнение алгоритма идет по ветке «Да», если условие не выполняется (false), то выполнение идет по ветке «Нет».	а. цикл do...while
2. Пока условие выполняется (результат логического выражения дает true), будут выполняться действия тела цикла	б. ветвление
3. В этом цикле первый раз условие проверяется лишь после выполнения действий тела цикла.	в. цикл while
4. Данный цикл также называют циклом «Для». В его заголовке указывается три параметра: начальное значение переменной (от), конечно значение (до) и ее изменение с помощью арифметической операции на каждом «обороте» цикла (шаг).	г. цикл for

15. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:

Проверяемые компетенции: ОК 01, ПК 1.1

- a) компилятор
- b) сканер
- c) транслятор
- d) интерпретатор

16. Результатом процесса формализации является...

Проверяемые компетенции: ОК 01, ПК 1.1

- a) Описательная модель
- b) Графическая модель
- c) Математическая модель
- d) Предметная модель

17. Какой из документов является алгоритмом?

Проверяемые компетенции: ОК 01, ПК 1.1

- a) Правила техники безопасности
- b) Инструкция по получению денег в банкомате

с) Расписание уроков

18. Информационной моделью организации занятий в колледже является:

Проверяемые компетенции: ОК 01, ПК 1.1

- a) Свод правил поведения учащихся
- b) Список группы
- c) Расписание уроков
- d) Перечень учебников

19. Укажите, какие из перечисленных понятий не относятся к основным группам алгоритмов?

Проверяемые компетенции: ОК 01, ПК 1.1

- a) Линейные
- b) Разветвляющиеся
- c) Структурные
- d) Циклические

20. Процесс поиска и исправления ошибок в программе называется:

Проверяемые компетенции: ОК 01, ПК 1.1

- a) Тестированием
- b) Отладкой
- c) Трансляцией
- d) Компиляцией

Критерии оценивания:

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

Раздел 2. Операторы языка JAVA

Форма контроля: тестирование

Время выполнения: 20 минут

1. Чему равен результат выполнения следующего выражения: $1000 / 100 \% 7 * 2$?
Проверяемые компетенции: ОК 02, ПК 1.1
2. Укажите последовательность команд, в результате выполнения которых значения переменных X и Y поменяются местами:
Проверяемые компетенции: ОК 02, ПК 1.1
 - a) $B=X; X=Y; Y=X;$
 - b) $X=Y; Y=X;$
 - c) $Y=X; B=X; X=Y;$
 - d) $C=X; X=Y; Y=C.$
3. Определите значение целочисленных переменных a, b и c после выполнения фрагмента программы (запишите значения последовательно через пробел)
Проверяемые компетенции: ОК 02, ПК 1.1

```
a = 50;
b = 17;
c = (b + a) / b + a;
a = (a / b)*b;
b = c - a;
```
4. Каков будет результат выражения $!(1 \ \&\& \ !(0 \ || \ 1))$?
Проверяемые компетенции: ОК 02, ПК 1.1
 - a) True
 - b) неоднозначность
 - c) False
5. Результат выполнения следующего фрагмента кода: $!((1 \ || \ 0) \ \&\& \ 0)$
Проверяемые компетенции: ОК 02, ПК 1.1
 - a) false
 - b) true
6. Какой из следующих логических операторов - логический оператор И?
Проверяемые компетенции: ОК 02 &
7. Что произойдет при обработке следующего оператора: `Console.WriteLine(a/0);`
Проверяемые компетенции: ОК 02, ПК 1.1
 - a) На консоль будет выведено число «0»
 - b) При выполнении будет возбуждена исключительная ситуация
 - c) При выполнении будет выведен текст "NaN"
 - d) При выполнении будет выведен результат вычисления выражения
8. Укажите группу, содержащую последовательность правильно записанных на языке C# знаков арифметических операций
Проверяемые компетенции: ОК 02
 - a) +, ^, -, /, =
 - b) +, ^, -, /, %
 - c) *, +, -, /, %
 - d) +, *, -, /, ==
9. Укажите правильную последовательность операций сравнения:
Проверяемые компетенции: ОК 02
 - a) ==, >=, <=, !=

- b) =, < >, > <, >=, <=
- c) ~ >, <, =,
- d) >=, <=, =, <, >

10. Как написать следующее выражение «Если переменная x больше нуля, то найти корень числа x»:

Проверяемые компетенции: ОК 02, ПК 1.1

- a) if (x > 0) { z = Math.Sqrt(x); }
- b) if x ≥ 0 z = Math.Sqrt(x);
- c) if (x > 0) { z = Math.Sqrt(); }
- d) if (x > 0) { z = Math.Pow(x); }

11. В каких случаях необходимо использовать оператор break в теле цикла?

Проверяемые компетенции: ОК 02, ПК 1.1

- a) всегда
- b) если необходимо досрочно выйти из цикла
- c) если необходимо отменить выполнение части операторов
- d) для перехода к началу цикла

12. Тело какого цикла всегда будет выполнено хотя бы один раз, независимо от истинности условия:

Проверяемые компетенции: ОК 02, ПК 1.1

- a) while
- b) do... while
- c) for
- d) Нет такого цикла в языке C#

13. До каких пор будут выполняться операторы в теле цикла while (x < 100)?

Проверяемые компетенции: ОК 02, ПК 1.1

- a) Пока x строго меньше ста
- b) Пока x равен ста
- c) Пока x меньше или равен ста
- d) Пока x больше ста

14. Цикл с параметром в языке C# запишется в виде:

Проверяемые компетенции: ОК 02

- a) for (i=1; i<n; i++)
- b) do ... while (условие)
- c) while (условие) ...
- d) case k <последовательность операторов>

15. Оператором цикла не является оператор

Проверяемые компетенции: ОК 02

- 1. while...
- 2. for...
- 3. switch...
- 4. do...while

16. Определите значение переменной S после выполнения операторов:

Проверяемые компетенции: ОК 02, ПК 1.1

```
i = 0; S = 0;
while (i < 3)
{
    i = i+1;
    S = S+i*i;
}
```

17. Чему будет равна переменная sum после выполнения фрагмента программы:

Проверяемые компетенции: ОК 02, ПК 1.1

```
sum = 0;
for(i = 7; i <= 9; i++)
    sum = sum + i;
```

18. В каком случае можно не использовать фигурные скобочки в операторе выбора if?

Проверяемые компетенции: ОК 02

- a) если в теле оператора if всего один оператор
- b) нет правильного ответа
- c) если в теле оператора if два и более операторов
- d) если в теле оператора if нет ни одного оператора

19. Что появится на экране, после выполнения этого фрагмента кода?

Проверяемые компетенции: ОК 02, ПК 1.1

```
int a = 1, b = 2;
if (a == b);
```

20. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?

Проверяемые компетенции: ОК 02

Критерии оценивания:

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

Раздел 3. Программирование базовых алгоритмов

Форма контроля: выполнение письменной работы

Время выполнения: 20 минут

Варианты вопросов

Опишите и нарисуйте блок-схему:

- Рекуррентного алгоритма
- Алгоритма вычисления конечных сумм и произведений.
- Алгоритма вычисления бесконечных сумм
- Алгоритма поиска делителей натурального числа.
- Алгоритма, раскладывающего натуральное число на цифры
- Алгоритма нахождения общего делителя двух натуральных чисел

Критерии оценивания:

Оценка 5 «отлично» теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

Оценка 4 «хорошо» теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

Оценка 3 «удовлетворительно» теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

Оценка 2 «неудовлетворительно» теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.

Раздел 8. Классы

Форма контроля: тестирование

Время выполнения: 20 минут

1. К проявлению какого элемента ООП можно отнести следующую фразу: "Перемещая рукоятку коробки передач автомобиля, человек может не задумываться о самом механизме переключения. С точки зрения водителей все коробки передач работают одинаково, хотя их механизмы на самом деле могут отличаться".

Проверяемые компетенции: ПК 1.1, ПК 1.2

2. Драконы умеют летать (как, например, птицы) и ползать (как, например, ящерицы). С точки зрения ООП, примером чего является данная ситуация (выберите наиболее точный вариант)?

Проверяемые компетенции: ПК 1.1, ПК 1.2

- a) инкапсуляция
- b) композиция
- c) наследование
- d) множественное наследование
- e) полиморфизм

3. Какому принципу ООП удовлетворяет обоняние собаки если: "Собака чует кошку - она лает, собака чует пищу - она бежит к миске"?

Проверяемые компетенции: ПК 1.1, ПК 1.2

4. Принцип объектно-ориентированного программирования, заключается в объединении атрибутов и методов объекта с целью обеспечения сохранности данных, называется:

Проверяемые компетенции: ПК 1.1, ПК 1.2

5. Возможность использовать одинаковые имена для методов входящих различные классы называется...

Проверяемые компетенции: ПК 1.1, ПК 1.2

6. Скрытие полей объекта с целью обеспечения доступа к ним только посредством методов класса называют...

Проверяемые компетенции: ПК 1.1, ПК 1.2

- a) наследованием
- b) делегированием
- c) полиморфизмом
- d) виртуализацией
- e) инкапсуляцией

7. Термин «Наследование» обозначает, что...

Проверяемые компетенции: ПК 1.1, ПК 1.2

- a) в производных классах присутствует часть состояния родительского класса
- b) производные классы содержат поля и методы родительского класса
- c) производные классы содержат методы родительского класса
- d) производные классы содержат поля родительского класса

8. Как называется принцип создания класса на базе существующего:

Проверяемые компетенции: ПК 1.1, ПК 1.2

9. Соотнести

Проверяемые компетенции: ПК 1.1, ПК 1.2

1. шаблон, определяющий характеристики объекта и описывающий его поведение и внешний вид	а. полиморфизм
2. свойство, которое позволяет одно и то же имя использовать для решения двух или более схожих, но технически разных задач	б. инкапсуляция
3. способность подкласса перенимать характеристики класса, на котором он базируется	в. класс
4. механизм, который объединяет данные и код	г. наследование

10. Соотнести

Проверяемые компетенции: ПК 1.1, ПК 1.2

1. наследование	а. множество объектов, которые обладают одинаковой структурой, поведением
2. инкапсуляция	б. определенное воздействие одного объекта на другой с целью вызвать соответствующую реакцию
3. класс	в. свойство объекта, заключающееся в том, что характеристики одного объекта (объекта-предка) могут передаваться другому объекту (объекту-потомку) без их повторного описания
4. операция	г. сокрытие внутренней структуры данных и реализации методов объекта от остальной программы

11. Механизм программирования, связывающий воедино код и данные, которыми он манипулирует, а также ограждающий их от внешнего доступа и неправильного применения – это...

Проверяемые компетенции: ПК 1.1, ПК 1.2

12. Свойство, с помощью которого один объект может приобретать свойства другого, при этом поддерживается концепция иерархической классификации, имеющей направление сверху вниз – это...

Проверяемые компетенции: ПК 1.1, ПК 1.2

13. Свойство, которое позволяет нескольким объектам, имеющим некоторые общие характеристики, получать доступ к одному интерфейсу и реализовывать упомянутые в нем методы – это...

Проверяемые компетенции: ПК 1.1, ПК 1.2

14. Модификаторы прав доступа: обозначение для членов класса, доступных в рамках объявляемого класса.

Проверяемые компетенции: ПК 1.1, ПК 1.2

- а) protected
- б) private

15. Модификаторы прав доступа: обозначение для общедоступных членов класса. К ним можно обратиться из любого метода любого класса программы.

Проверяемые компетенции: ПК 1.1, ПК 1.2

- а) public
- б) protected

16. Модификаторы прав доступа: обозначение для членов класса, доступных из методов классов, объявляемых в рамках сборки, содержащей объявление данного класса.

Проверяемые компетенции: ПК 1.1, ПК 1.2

- а) public

b) protected

17. Какое утверждение относительно языка JAVA не верно?

Проверяемые компетенции: ПК 1.1, ПК 1.2

- a) Допустимо множественное наследование
- b) Класс может реализовать несколько интерфейсов
- c) Интерфейс может наследоваться от множества других интерфейсов

18. Модификаторы прав доступа: обозначение для членов класса, доступных в рамках объявляемого класса и из методов производных классов.

Проверяемые компетенции: ПК 1.1, ПК 1.2

- a) public
- b) protected
- c) internal
- d) protected internal

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

Раздел 9. Объектно-ориентированное программирование

Форма контроля: тестирование

Время выполнения: 20 минут

1. Соотнести

1. метка	а. TextBox
2. флажок	б. Label
3. текстовое поле	в. ComboBox
4. выпадающий список	г. CheckBox

Проверяемые компетенции: ОК 09, ПК 1.1

2. Вкладка Events в Object Inspector предназначена для...

Проверяемые компетенции: ОК 09, ПК 1.2

- а) изменения свойств компонентов
- б) редактирования кода программы
- с) поиска компонентов
- д) просмотра структуры проекта
- е) определения событий

3. Для создания контекстного меню используется компонент класса:

Проверяемые компетенции: ОК 09, ПК 1.2

- а) Button
- б) TextBox
- с) CheckBox
- д) PopupMenu
- е) ContextMenuStrip

4. Для создания главного меню используется компонент класса:

Проверяемые компетенции: ОК 09, ПК 1.2

- а) Label
- б) Menu
- с) MenuStrip
- д) TMainMenu
- е) TPopupMenu

5. Для создания однострочного редактируемого текстового поля используется компонент класса:

Проверяемые компетенции: ОК 09, ПК 1.2

6. Свойство, отвечающее за высоту формы:

Проверяемые компетенции: ОК 09, ПК 1.2

7. Событие OnResize генерируется, когда:

Проверяемые компетенции: ОК 09, ПК 1.2

- а) окно стало активным
- б) изменяется размер окна
- с) окно закрывается
- д) двигается мышка
- е) нет такого события

8. Для создания строки состояния используется компонент класса:

Проверяемые компетенции: ОК 09, ПК 1.2

- а) ProgressBar
- б) StatusStrip

- c) ToolBar
- d) StatusBar
- e) TrackBar

9. Какое свойство компонента класса TextBox отвечает за текст находящийся в нём:

Проверяемые компетенции: ОК 09, ПК 1.2

10. Для того чтобы сделать любой компонент невидимым предназначено свойство...

Проверяемые компетенции: ОК 09, ПК 1.2

11. Ограничение на количество вводимых символов для компонента TextBox задаётся в свойстве...

Проверяемые компетенции: ОК 09, ПК 1.2

- a) Text
- b) Left
- c) Font
- d) AutoSize
- e) MaxLength

12. Когда нажата кнопка мыши, генерируется событие:

Проверяемые компетенции: ОК 09, ПК 1.2

- a) MouseMove
- b) MouseUp
- c) MouseDown
- d) MouseWheel
- e) Click

13. Вкладка Properties в Properties Window предназначена для...

Проверяемые компетенции: ОК 09, ПК 1.2

- a) редактирования кода программы
- b) поиска компонентов
- c) просмотра структуры проекта
- d) определения событий
- e) изменения свойств компонентов

14. Свойство формы, отвечающее за заголовок окна.

Проверяемые компетенции: ОК 09, ПК 1.2

- a) Caption
- b) Text
- c) Name
- d) Font
- e) String

15. Текст подсказки, появляющийся при наведении указателя мыши на кнопку, задаётся в свойстве...

Проверяемые компетенции: ОК 09, ПК 1.2

- a) Caption
- b) ShowHint
- c) Hint
- d) Text
- e) String

16. Какое имя получит обработчик события соответствующий щелчку мыши на кнопке Button?

Проверяемые компетенции: ОК 09, ПК 1.1

- a) void button1_KeyPress(object sender, KeyPressEventArgs e)
- b) void button1_MouseDown(object sender, MouseEventArgs e)
- c) void textBox3_TextChanged(object sender, EventArgs e)
- d) void button1_Click(object sender, EventArgs e)

17. Microsoft.NET (.NET Framework)

Проверяемые компетенции: ОК 09, ПК 1.2

- a) программная платформа.
- b) общая спецификация языков программирования.
- c) среда Времени Выполнения или Виртуальная Машина.

18. Для создания многострочного редактируемого текстового поля используется компонент класса:

Проверяемые компетенции: ОК 09, ПК 1.2

- a) Label
- b) MainMenu
- c) Memo
- d) RichTextBox
- e) Form

19. Свойство формы, задающее степень ее прозрачности:

Проверяемые компетенции: ОК 09, ПК 1.2

- a) AlphaBlendValue
- b) Opacity
- c) AlphaBlend
- d) AutoSize
- e) BorderStyle

20. Для выравнивания текста по правому или левому краю у компонента класса TextBox имеется свойство...

Проверяемые компетенции: ОК 09, ПК 1.2

- a) TextAlign
- b) AutoSize
- c) Font
- d) Alignment
- e) Left

21. Для установки графического значка отображающегося в заголовке окна используется свойство...

Проверяемые компетенции: ОК 09, ПК 1.2

- a) BorderStyle
- b) Icon
- c) Caption
- d) Glyph
- e) Bitmap

22. Какое событие генерируется при отпускании кнопки на клавиатуре?

Проверяемые компетенции: ОК 09, ПК 1.2

- a) Click
- b) KeyPress
- c) KeyUp

- d) KeyDown
- e) MouseDown

23. CLS (Common Language Specification)

Проверяемые компетенции: ОК 09, ПК 1.2

- a) программная платформа.
- b) общая спецификация языков программирования.
- c) среда Времени Выполнения или Виртуальная Машина.

24. CLR (Common Language Runtime)

Проверяемые компетенции: ОК 09, ПК 1.2

- a) программная платформа.
- b) общая спецификация языков программирования.
- c) среда Времени Выполнения или Виртуальная Машина.

25. Поддержка ввода-вывода

Проверяемые компетенции: ОК 09, ПК 1.2

- a) System.IO
- b) System.Net
- c) System.Reflection , System.Reflection.Emit
- d) System.Runtime.InteropServices, System.Runtime.Remoting

26. Работа с элементами интерфейса Windows

Проверяемые компетенции: ОК 09, ПК 1.2

- a) System.Security
- b) System.Threading
- c) System.WEB
- d) System.Windows.Forms

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

Межсессионный контроль (тестирование)

Тест «Основные понятия и операторы языка C#»

1. Результат работы фрагмента программы

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

```
char a, b, c;  
a = 'b';  
b = 'c';  
c = a;  
Console.WriteLine("{0}{1}{2}{3}", a, b, c, 'c');
```

2. Значение переменной c после выполнения фрагмента программы равно

```
int a = 3, b = 2, c;  
c = a / b;
```

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

3. В результате работы программного фрагмента переменная c примет значение

```
int c;  
Console.WriteLine(c);
```

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- a. 0
- b. никакое, т.к. произойдет ошибка этапа компиляции
- c. никакое, т.к. произойдет ошибка этапа выполнения программы
- d. неопределенное значение

4. Результат работы фрагмента программы

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

```
for (int i=1; i<=3; i++)  
{  
Console.Write("i");  
}
```

5. Результат работы фрагмента программы

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

```
int x = 10;  
while (x < 0)  
{  
Console.Write("{0}\t", x);  
x = x - 3;  
}
```

- a. 10 7 4
- b. тело цикла не выполнится ни разу
- c. 10 7 4 1
- d. фрагмент кода содержит ошибку

6. Работа каких операторов дает верный ответ

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- a. `if (x/2==0) Console.WriteLine("x четное");`
 `else Console.WriteLine("x нечетное");`
- b. `if (x%2==0) Console.WriteLine("x четное");`
 `else Console.WriteLine("x нечетное");`
- c. `if (x%2!=0) Console.WriteLine("x нечетное");`
 `else Console.WriteLine("x четное");`
- d. `if (x%2!=0) Console.WriteLine("x четное");`
 `else Console.WriteLine("x нечетное");`

7. Результат выполнения фрагмента программы

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

```
int x = 10;
do
{
    Console.Write("{0}\t", x);
    x = x - 3;
}
while (x < 0);
```

8. Результат работы фрагмента программы

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

```
int x = 10;
while (x > 0)
{
    Console.Write("{0}\t", x);
    x = x - 3;
}
```

- a. 10 7 4 1
- b. 7 4 1
- c. 10 7 4
- d. 10
- e. 1

9. Результат работы фрагмента программы

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

```
int x = 10;
do
{
    Console.Write("{0}\t", x);
    x = x - 3;
}
while (x > 0);
```

- a. 10 7 4 1
- b. 10

- c. тело цикла не выполнится ни разу
- d. фрагмент кода содержит ошибку

10. Что будет выведено на экран после выполнения программного фрагмента
Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

```
int i;  
for (i = 1; i <= 5; )  
{  
    i++;  
}  
Console.WriteLine("{0}", i);
```

11. Что будет выведено на экран после выполнения программного фрагмента
Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

```
int i;  
for (i = 1; i >= 5; )  
{  
    i--;  
}  
Console.WriteLine("{0}", i);
```

1

12. Работа какого оператора дает верный ответ, при условии, что a,b – переменные типа int, имеющие ненулевое значение

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

a. if (a*b>0)

Console.WriteLine("одно из двух данных чисел положительное,
второе – отрицательное");

b. if (a*b<0)

Console.WriteLine("одно из двух данных чисел положительное,
второе –
отрицательное");

c. if (a*b<0)

Console.WriteLine("оба числа одинакового знака");

13. Что будет выведено в результате работы следующего фрагмента

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

```
int i = 17;  
while (i != 1)  
{  
    Console.Write("{0}", i);  
    i = 3 * i + 1;  
    while (i % 2 == 0)  
        i /= 2;  
}
```

a. 14 13 5 4 1

b. 17 13 5

- c. 17 13 5 1
- d. 17 15 13 11 9 7 5 3 1

1. Массив объявлен следующим образом

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

```
int[] myArray = new int[10];
```

Какой из следующих фрагментов программы будет правильно выводить элементы данного массива:

- a.

```
for (i = 1; i <= 10; i++)  
Console.WriteLine(myArray[i]);
```
- b.

```
for (i = 0; i <= 10; i++)  
Console.WriteLine(myArray[i]);
```
- c.

```
for (i = 0; i < 10; i++)  
Console.WriteLine(myArray[i]);
```
- d.

```
for (i = 1; i < 10; i++)  
Console.WriteLine(myArray[i]);
```

2. Какие из следующих объявлений выполнены корректно:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- a.

```
int [][] a= new int [3][];  
a[0]=new int [5];  
a[1]=new int [3];  
a[2]=new int [4];
```
- b.

```
int [3][] a= {new int [5], new int [3], new int [4]};
```
- c.

```
int [][] a= {new int [5], new int [3], new int [4]};
```
- d.

```
int [3][] a= new int [][];  
a[0]=new int [5];  
a[1]=new int [3];  
a[2]=new int [4];
```
- e.

```
int [][] a= new int [][];  
a[0]=new int [5];  
a[1]=new int [3];  
a[2]=new int [4];
```

3. Массив объявлен следующим образом:

```
int[ , , ] a=new int[5,2,3];
```

Чему равно значение a.Length?

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

4. Массив объявлен следующим образом:

```
int[, ] MyArray = new int[4, 7];
```

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

Какой метод возвращает количество столбцов данного массива?

- a. `MyArray.Length[0]`
- b. `MyArray.GetLength(0)`
- c. `MyArray.GetLength(1)`
- d. `MyArray.Rank(0)`
- e. `MyArray.Length[1]`

5. Что будет выведено на экран после выполнения следующего программного фрагмента

```
int [ , ] my_array= new int[4,4];
int i, j;
for (i = 0; i < 4; i++)
for (j = 0; j < 4; j++)
my_array[i,j] = j;
for (i = 0; i < 4; i++)
{
    for (j = 0; j < 4; j++)
        Console.Write(my_array[i, j]);
}
Console.WriteLine();
```

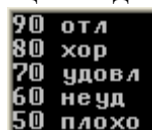
Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- a. 0123012301230123
- b. 1234123412341234
- c. 3210321032103210
- d. 432143214321

19. Пусть задан массив баллов и массив соответствующих им оценок

```
double[] ball = { 90.0, 80.0, 70.0, 60.0, 50.0 };
string[] grade = { "отл", "хор", "удовл", "неуд", "плохо"};
```

Какой из следующих фрагментов выводит таблицу соответствия оценок баллам в следующем виде:



```
90 отл
80 хор
70 удовл
60 неуд
50 плохо
```

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- a. `foreach (double mark in boundary)`
`Console.WriteLine(mark+" "+grade[mark]);`
- b. `for (int i = 0; i < ball.Length; i++)`
`Console.WriteLine(ball[i] + " " + grade[i]);`
- c. `foreach (double mark in ball, string symbol in grade)`
`Console.WriteLine(mark+" "+symbol);`
- d. `foreach (double mark in ball)`
`foreach(string symbol in grade)`

20. Массив инициализирован следующим образом:


```
int [,]a={{4, 5}, {1, 3}};
```

Элемент a[1,1] имеет значение:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

21. Массив определен следующим образом:

```
int []a=new int [10];  
for (int i=0; i<a.Length; ++i)  
    a[i]=2*i+1;
```

Элемент a[4] имеет значение:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

22. В результате выполнения фрагмента программы:

```
int[,] a ={ { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };  
int s=0;  
foreach (int i in a)  
    s+=i;  
Console.WriteLine(s);
```

на экран будет выведено значение:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

23. В результате выполнения фрагмента программы:

```
int[,] a ={ { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };  
int s=a.Length;  
Console.WriteLine(s);
```

на экран будет выведено значение: 9

24. В результате выполнения фрагмента программы:

```
char [] a={'м', 'и', 'р', 'а', 'ж'};  
string line=new string (a, 1, 3);  
Console.WriteLine(line);
```

на экран будет выведено:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

25. В результате выполнения фрагмента программы:

```
string line="кол около колокола";  
char []a={'л','о'};  
Console.WriteLine(line.IndexOfAny(a));
```

на экран будет выведено:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- a. 1
- b. 2
- c. 3

26. В результате выполнения фрагмента программы:

```
StringBuilder str=new StringBuilder("кол около колокола");  
str.Replace("о","");  
Console.WriteLine(str);
```

на экран будет выведено:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

27. В результате выполнения фрагмента программы:

```
string str="кол около колокола";  
str.Replace("о","");  
Console.WriteLine(str);
```

на экран будет выведено:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

28. Что будет выведено в результате выполнения следующего программного фрагмента:

```
string s = "Hello!\n";  
Console.WriteLine("{0}",s.Length);
```

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

29. В результате какой последовательности операторов на экран будет выведена строка «ученье – свет»

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

a.

```
StringBuilder s = new StringBuilder();  
s.Append("ученье тьма" );  
s.Remove(7,4);  
s.Append(" свет" );  
s.Insert(7,'-');  
Console.WriteLine(s);
```

b.

```
String s = new String();  
s.Append("ученье тьма" );  
s.Remove(7,4);  
s.Append(" свет" );  
s.Insert(7,'-');  
Console.WriteLine(s);
```

c.

```
StringBuilder s = new StringBuilder();  
s.Append("ученье тьма" );  
s.Remove(8,4);  
s.Append(" свет" );  
s.Insert(8,'-');  
Console.WriteLine(s);
```

d.

```
StringBuilder s = new StringBuilder();  
s.Append("ученье тьма" );  
s.Remove(7,4);  
s.Insert(7,'-');  
s.Append("свет" );
```

```
Console.WriteLine(s);
```

30. В результате выполнения следующего фрагмента программы

```
double[] a=new double[10];  
for (int i=1; i<=10; i++)  
    a[i] = i ;  
Console.WriteLine("Массив заполнен") ;
```

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- a. появится сообщение “Массив заполнен”
- b. произойдет ошибка компиляции из-за несоответствия типов в операторе присваивания
- c. будет сгенерировано исключение `IndexOutOfRangeException`

31. Какие операции допустимы с логическими переменными?

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. логическое сложение
- B. логическое умножение
- C. отрицание
- D. логическое вычитание

32. Как написать следующее выражение «Если переменная a меньше b то мы инкрементируем переменную count »?

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. `if (a < b) { c++; }`
- B. `if (a < b) { c--; }`
- C. `if (a >= b) { +c; }`
- D. `if (a < b) { --c; }`

33. Как написать следующее выражение «Если переменная index больше size то мы инкрементируем переменную count »?

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. `if (index>size) { count++; }`
- B. `if (index<size) { count--; }`
- C. `if (index>=size) { ++count; }`
- D. `if (index<size) { --count; }`

34. Что определяет для массива X[n] следующий алгоритм

```
S = 0;  
for (k = 0; k < n; k++)  
    if (X[k] < 0) S = S+X[k];
```

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. минимальный элемент массива;
- B. сумму отрицательных элементов массива;
- C. максимальный элемент массива;
- D. количество отрицательных элементов массива;
- E. индекс последнего отрицательного элемента массива;

35. Глобальные переменные действуют:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. во всех процедурах;
- B. во всех функциях;
- C. во всех модулях;
- D. во всей программе.

36. Какой цикл лучше использовать для работы с массивами, если тело цикла надо выполнить для каждого элемента массива:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. do while
- B. while
- C. for
- D. foreach

37. Какой из документов является алгоритмом?

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. Правила техники безопасности
- B. Инструкция по получению денег в банкомате
- C. Расписание уроков

Укажите, какие из перечисленных понятий не относятся к основным группам алгоритмов?

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. Линейные
- B. Разветвляющиеся
- C. Структурные
- D. Циклические

38. Процесс поиска и исправления ошибок в программе называется:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. Тестированием
- B. Отладкой
- C. Трансляцией
- D. Компиляцией

39. Ошибки, связанные с нарушением правил языка программирования называются:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. Семантическими
- B. Содержательными
- C. Синтаксическими
- D. Тупиковыми

40. Какое из обязательных свойств алгоритмов нарушено при организации бесконечного цикла?

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

- A. Результативность
- B. Дискретность
- C. Массовость
- D. Понятность
- E. Определенность (точность) алгоритма

41. Укажите функцию, позволяющую возвести число x в степень:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

42. Укажите функцию, позволяющую вычислить корень числа:

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

43. Что будет выведено в результате выполнения программы?

```
int b1 = 456;  
int b2 = 123;  
b2 = b2*1000+b1;  
Console.WriteLine(b2);
```

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

44. Сколько строк напечатает программа:

```
int k, l;  
for (k = 8; k >= 1; k--)  
for (l = 10; l <= 14; l++)  
Console.WriteLine("Мне нравится JAVA")
```

Проверяемые компетенции: ОК 01, ОК 02, ПК 1.1

Промежуточная аттестация

1.1 Цели и задачи промежуточной аттестации

Целью промежуточной аттестации является проверка и оценка уровня освоения обучающимися знаний, умений ОП.04 «Основы алгоритмизации и программирования» и сформированности компетенций.

Главной задачей промежуточной аттестации обучающихся является установление соответствия результата освоения знаний и умений, сформированности общих и профессиональных компетенций требованиям Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование.

1.2. Форма промежуточной аттестации

– Экзамен (4 семестр).

1.3 Система оценивания результатов выполнения заданий

Оценивание результатов выполнения заданий промежуточной аттестации осуществляется на основе следующих принципов:

достоверности оценки – оценивается уровень сформированности знаний, умений, практического опыта, общих и профессиональных компетенций, продемонстрированных обучающимися в ходе выполнения задания;

адекватности оценки – оценка выполнения заданий должна проводиться в отношении тех компетенций, которые необходимы для эффективного выполнения задания; надежности оценки – система оценивания выполнения заданий должна обладать высокой степенью устойчивости при неоднократных оценках уровня сформированности знаний, умений, практического опыта, общих и профессиональных компетенций обучающихся;

комплексности оценки – система оценивания выполнения заданий должна позволять интегративно оценивать общие и профессиональные компетенции обучающихся;

объективности оценки – оценка выполнения конкурсных заданий должна быть независимой от особенностей профессиональной ориентации или предпочтений преподавателей, осуществляющих контроль или аттестацию.

При выполнении процедур оценки заданий используются следующие основные методы:

метод расчета первичных баллов;

метод расчета сводных баллов.

Результаты выполнения заданий оцениваются в соответствии с разработанными критериями оценки. Используется пятибалльная шкала для оценивания результатов обучения.

Перевод пятибалльной шкалы учета результатов в пятибалльную 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
Оценка 3 «удовлетворительно»	3-3,5
Оценка 2 «неудовлетворительно»	≤ 2,9

1.4. Материально-техническое обеспечение для проведения промежуточной аттестации

Аттестация проводится в «Кабинет информатики» и лаборатории «Лаборатория программирования и баз данных».

1.5 Контрольно-оценочные средства

Задания для экзамена

Форма аттестации: выполнение экзаменационного задания.

Задание:

1. Теоретическое: ответить на вопросы.
2. Практическое: выполнение практического задания.

Условия выполнения задания:

- 1) Задание выполняется в лаборатории «Лаборатория программирования и баз данных».
- 2) Обучающиеся устно отвечают на 1 теоретический вопрос, при выполнении задания обучающийся может в письменной форме дать ответ или составить план ответа на вопрос. После истечения времени, отводимого на подготовку, обучающийся дает устный ответ на вопрос задания.
- 3) Обучающиеся решают 1 практическое задание, задание выполняется на компьютере и сохраняется в форме электронного документа.
- 4) Время, отводимое на выполнение задания одним обучающимся – 90 минут, в том числе:
собеседование по вопросу – 30 мин (20 мин. на подготовку, 10 мин. на собеседование);
выполнение практического задания – 60 мин.
- 5) Максимальный балл за задание – 5 баллов, в том числе:
собеседование по вопросу – 2 балла;
решение практического задания – 3 балла.

Материально-техническое обеспечение: персональный компьютер, MS Visual Studio

Перечень вопросов теоретического задания:

1. Понятие алгоритмизации, свойства алгоритмов, основные свойства алгоритмической конструкции
2. Основные элементы языка программирования JAVA
3. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных
4. Операции и выражения
5. Правила формирования и вычисления выражений
6. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор
7. Условный оператор. Оператор выбора
8. Цикл с постусловием. Цикл с предусловием
9. Цикл с параметром. Вложенные циклы
10. Методы, механизм передачи параметров
11. Рекурсивные методы
12. Одномерные массивы
13. Двумерные массивы. Ступенчатые массивы
14. Сортировка: метод «пузырька»,
15. Сортировка вставками
16. Сортировка: алгоритм Шелла,
17. Быстрая сортировка
18. Поиск: последовательный, двоичный
19. Символы и строки. Стандартные функции и процедуры для работы со строками
20. Типы файлов. Организация доступа к файлам. Иерархия потоков
21. Абстрактный тип данных: список, стек, очередь
22. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс
23. Основные принципы ООП: инкапсуляция, наследование, полиморфизм
24. Классы объектов. Компоненты и их свойства. Событийно-управляемая модель программирования
25. Классы ООП: виды, назначение, свойства, методы, события. Перегрузка методов
26. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение

Практические задания:

Составить программу на языке JAVA в среде разработки Visual Studio (по вариантам).
Готовое решение сохранить в папке «Рабочий стол».

Классы и объекты. Методы класса.

<p>Вариант 1 Создать класс Автобус Описать поля: Марка Госномер Комфортность Наличие кондиционера Количество купленных мест Цена билета Создать метод, рассчитывающий выручку. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>	<p>Вариант 5 Создать класс Кинофильм Описать поля: название режиссёр количество частей (серий) продолжительность одной части страна-производитель Создать метод, рассчитывающий продолжительность фильма. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>
<p>Вариант 2 Создать класс Сотрудник Описать поля: фамилия имя пол часовая оплата отработанное время в часах Создать метод, рассчитывающий зарплату. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>	<p>Вариант 6 Создать класс Спортклуб Описать поля: название площадь зала наличие бассейна количество абонементов цена абонемента Создать метод, рассчитывающий общую выручку. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>
<p>Вариант 3 Создать класс Книга Описать поля: название жанр год издания количество экземпляров цена книги Создать метод, рассчитывающий выручку. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>	<p>Вариант 7 Создать класс Журнал Описать поля: название жанр тираж цена количество экземпляров Создать метод, рассчитывающий общую стоимость журналов. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>
<p>Вариант 4 Создать класс Отдел Описать поля: название отдела этаж</p>	<p>Вариант 8 Создать класс Товар Описать поля: название код товара</p>

<p>№ комнаты количество работников по штату оклад одного работника Создать метод, рассчитывающий фонд оплаты труда. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>	<p>цена цвет количество товара Создать метод, рассчитывающий общую стоимость товара. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>
<p>Вариант 9 Создать класс Автомобиль Описать поля: Марка Производитель Наличие кондиционера Цена 1 литра бензина Пробег в км Расход бензина в литрах на 100 км Создать метод, рассчитывающий сумму израсходованных денег на топливо. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>	<p>Вариант 15 Создать класс Бассейн Описать поля: название клуба здоровья ширина бассейна длина бассейна глубина бассейна наличие возможности предварительного заказа билета цена билета Создать метод, рассчитывающий объём бассейна. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>
<p>Вариант 10 Создать класс Легкоатлет Описать поля: фамилия имя пол номер спортсмена расстояние, которое преодолел спортсмен затраченное время Создать метод, рассчитывающий среднюю скорость бега спортсмена. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>	<p>Вариант 16 Создать класс Школа Описать поля: номер школы название наличие спортзала площадь одного класса количество классов Создать метод, рассчитывающий общую полезную площадь для занятий. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>
<p>Вариант 11 Создать класс Бригада Описать поля: номер бригады выполняемая работа количество отработанных часов количество человек в бригаде стоимость отработанного часа Создать метод, рассчитывающий общий заработок всей бригады. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>	<p>Вариант 17 Создать класс Автобусный рейс Описать поля: направление номер рейса стоимость всей поездки общее расстояние между конечными пунктами наличие кондиционера Создать метод, рассчитывающий стоимость проезда 1 км. Создать 2 объекта. Для каждого вызвать метод, и вывести результат расчёта на консоль.</p>

	консоль.
<p>Вариант 12</p> <p>Создать класс Грузчик</p> <p>Описать поля:</p> <p>фамилия</p> <p>номер заказа</p> <p>количество этажей</p> <p>стоимость подъёма 1 мешка груза на 1 этаж</p> <p>количество мешков груза</p> <p>Создать метод, рассчитывающий общую стоимость оказанной услуги (заказа).</p> <p>Создать 2 объекта. Для каждого вызвать метод,</p> <p>и вывести результат расчёта на консоль.</p>	<p>Вариант 18</p> <p>Создать класс Такси</p> <p>Описать поля:</p> <p>марка машины</p> <p>номер машины</p> <p>номер заказа</p> <p>наличие кондиционера</p> <p>общая стоимость проезда</p> <p>общее расстояние поездки</p> <p>Создать метод, рассчитывающий стоимость проезда 1 км.</p> <p>Создать 2 объекта. Для каждого вызвать метод,</p> <p>и вывести результат расчёта на консоль.</p>
<p>Вариант 13</p> <p>Создать класс Кошка</p> <p>Описать поля:</p> <p>кличка</p> <p>возраст</p> <p>пол</p> <p>количество необходимых пакетов корма в день</p> <p>цена 1 пакета корма</p> <p>Создать метод, рассчитывающий общие затраты на содержание животного в месяц (30 дней).</p> <p>Создать 2 объекта. Для каждого вызвать метод,</p> <p>и вывести результат расчёта на консоль.</p>	<p>Вариант 19</p> <p>Создать класс Телефон</p> <p>Описать поля:</p> <p>номер</p> <p>оператор</p> <p>наличие скидки</p> <p>скидка в процентах</p> <p>стоимость 1 мин разговора</p> <p>количество израсходованных минут</p> <p>Создать метод, рассчитывающий общую стоимость разговоров.</p> <p>Создать 2 объекта. Для каждого вызвать метод,</p> <p>и вывести результат расчёта на консоль.</p>
<p>Вариант 14</p> <p>Создать класс Курьер</p> <p>Описать поля:</p> <p>фамилия</p> <p>номер курьера</p> <p>стоимость доставки 1 заказа</p> <p>количество выполненных заказов</p> <p>наличие личного транспорта</p> <p>Создать метод, рассчитывающий общую оплату за доставку заказов.</p> <p>Создать 2 объекта. Для каждого вызвать метод,</p> <p>и вывести результат расчёта на консоль.</p>	<p>Вариант 20</p> <p>Создать класс Склад</p> <p>Описать поля:</p> <p>название товара</p> <p>код товара</p> <p>цена 1 кг товара</p> <p>вес 1 ящика товара</p> <p>количество ящиков</p> <p>Создать метод, рассчитывающий общую стоимость товара.</p> <p>Создать 2 объекта. Для каждого вызвать метод,</p> <p>и вывести результат расчёта на консоль.</p>

Методы. Конструкторы. Несколько конструкторов.

<p>Вариант 21</p> <p>Создать класс Автомобиль</p> <p>Описать поля:</p> <ul style="list-style-type: none"> • марка • производитель • наличие кондиционера 	<p>Вариант 22</p> <p>Создать класс Бассейн</p> <p>Описать поля:</p> <ul style="list-style-type: none"> • название центра здоровья • ширина бассейна • длина бассейна
---	---

<ul style="list-style-type: none"> • цена 1 литра бензина • пробег в км • расход бензина в литрах на 100 км <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>	<ul style="list-style-type: none"> • глубина бассейна • наличие возможности предварительного заказа билета • цена билета <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>
<p>Вариант 23 Создать класс Легкоатлет Описать поля:</p> <ul style="list-style-type: none"> • фамилия • имя • пол • номер спортсмена • расстояние, которое преодолел спортсмен • затраченное время <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>	<p>Вариант 24 Создать класс Школа Описать поля:</p> <ul style="list-style-type: none"> • номер школы • Ф.И.О директора • адрес (улица, дом) • наличие спортзала • площадь одного класса • количество классов <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>
<p>Вариант 25 Создать класс Бригада Описать поля:</p> <ul style="list-style-type: none"> • номер бригады • выполняемая работа • количество отработанных часов • количество человек в бригаде • стоимость отработанного часа <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>	<p>Вариант 26 Создать класс Автобусный рейс Описать поля:</p> <ul style="list-style-type: none"> • направление • номер рейса • стоимость всей поездки • общее расстояние между конечными пунктами • наличие кондиционера <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>
<p>Вариант 27 Создать класс Грузчик Описать поля:</p> <ul style="list-style-type: none"> • фамилия • номер заказа • количество этажей • стоимость подъёма 1 мешка груза на 1 этаж • количество мешков груза <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>	<p>Вариант 28 Создать класс Такси Описать поля:</p> <ul style="list-style-type: none"> • марка машины • номер машины • номер заказа • наличие кондиционера • общая стоимость проезда • общее расстояние поездки <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>
<p>Вариант 29 Создать класс Кошка Описать поля:</p> <ul style="list-style-type: none"> • кличка 	<p>Вариант 30 Создать класс Телефон Описать поля:</p> <ul style="list-style-type: none"> • номер

<ul style="list-style-type: none"> • возраст • пол • количество необходимых пакетов корма в день • цена 1 пакета корма <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>	<ul style="list-style-type: none"> • оператор • наличие скидки • скидка в процентах • стоимость 1 мин разговора • количество израсходованных минут <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>
<p>Вариант 31 Создать класс Курьер Описать поля:</p> <ul style="list-style-type: none"> • фамилия • номер курьера • стоимость доставки 1 заказа • количество выполненных заказов • наличие личного транспорта <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>	<p>Вариант 32 Создать класс Склад Описать поля:</p> <ul style="list-style-type: none"> • название товара • код товара • цена 1 кг товара • вес 1 ящика товара • количество ящиков <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>
<p>Вариант 33 Создать класс Автобус Описать поля:</p> <ul style="list-style-type: none"> • Марка • Госномер • Комфортность • Наличие кондиционера • Количество купленных мест • Цена билета <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>	<p>Вариант 34 Создать класс Журнал Описать поля:</p> <ul style="list-style-type: none"> • название • жанр • тираж • цена • количество экземпляров <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>
<p>Вариант 35 Создать класс Сотрудник Описать поля:</p> <ul style="list-style-type: none"> • фамилия • имя • пол • часовая оплата • отработанное время в часах <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>	<p>Вариант 36 Создать класс Спортклуб Описать поля:</p> <ul style="list-style-type: none"> • название • площадь зала • наличие бассейна • количество абонементов • цена абонемента <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>
<p>Вариант 37 Создать класс Книга Описать поля:</p> <ul style="list-style-type: none"> • название • жанр 	<p>Вариант 38 Создать класс Кинофильм Описать поля:</p> <ul style="list-style-type: none"> • название • режиссёр

<ul style="list-style-type: none"> • год издания • количество экземпляров • цена книги <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>	<ul style="list-style-type: none"> • количество частей (серий) • продолжительность одной части • страна-производитель <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>
<p>Вариант 39</p> <p>Создать класс Отдел</p> <p>Описать поля:</p> <ul style="list-style-type: none"> • название отдела • этаж • № комнаты • количество работников по штату • оклад одного работника <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>	<p>Вариант 40</p> <p>Создать класс Товар</p> <p>Описать поля:</p> <ul style="list-style-type: none"> • название • код товара • цена • цвет • количество товара <p>Создать в классе несколько конструкторов для возможности задания сразу всех указанных параметров или нескольких при создании экземпляров.</p>

1.7 Критерии оценки

1.7.1. Критерии оценки (Экзамен)

Критерии оценки результатов выполнения теоретического задания		Баллы в соответствии с критериями оценки
		Максимальный балл – 2,0 балла
	<p>Демонстрирует глубокое, полное знание и понимание программного материала.</p> <p>Последовательно, самостоятельно раскрывает основное содержание вопроса.</p> <p>Выводы аргументированы, основаны на самостоятельно выполненном анализе, обобщении данных.</p> <p>Четко и верно даны определения понятий и научных терминов.</p> <p>Дает верные, самостоятельные ответы на вопросы.</p>	2,0
	<p>Демонстрирует недостаточно глубокое, полное знание и понимание программного материала.</p> <p>Недостаточно последовательно, но самостоятельно раскрывает основное содержание вопроса.</p> <p>Выводы основаны на самостоятельно выполненном анализе, обобщении данных, но в отдельных случаях недостаточно аргументированы.</p> <p>Недостаточно четко и верно даны определения понятий и научных терминов.</p> <p>При ответе на вопросы допускает несущественные ошибки, которые может исправить самостоятельно.</p>	1,5
	Демонстрирует в отдельных вопросах, неглубокое	0,8

	<p>владение знаниями программного материала.</p> <p>Излагает программный материал фрагментарно, не всегда последовательно.</p> <p>Допущены ошибки и неточности в использовании научной терминологии.</p> <p>При ответе на вопросы допускает неточности.</p>	
	<p>Студент демонстрирует незнание и непонимание программного материала.</p> <p>Основное содержание учебного материала не раскрыто;</p> <p>допущены грубые ошибки в определении понятий, при использовании терминологии.</p> <p>Затрудняется отвечать на вопросы, при ответе допускает серьезные ошибки.</p>	0
	ИТОГО	2

Критерии оценки результатов выполнения практического задания		Баллы в соответствии с критериями оценки
Соблюдение алгоритма выполнения задания		Максимальный балл – 1,0 балл
	- выполнение задания осуществляется по предложенному алгоритму, к каждому шагу выполнения предоставлена копия экрана	1
	- алгоритм выполнения задания отсутствует	0
Оформление задания в качестве текстового документа		Максимальный балл – 1,0 балл
	- верно оформлено описание практического задания, представлены все копии экрана, подтверждающие шаги выполнения	1
	- описание задания оформлено с незначительными неточностями, 1-2 копии экрана отсутствуют или представлены неверно	0,5
	- описание практического задания оформлено неверно	0
Достижение результата после выполнения задания		Максимальный балл – 0,5 балла
	- итоговый результат достигнут в полном объеме	0,5
	- достижение результата достигнуто не в полном объеме, отсутствуют отдельные моменты	0,3
	- результат выполнения не достигнут	0
Устное объяснение выполненного задания, вывод о проделанной работе		Максимальный балл – 0,5 балла
	- объяснение выполнения задания последовательно, связно, логично, вывод аргументирован и обоснован; правильно и обстоятельно дается ответ (ответы) на сопутствующие вопрос (вопросы)	0,5
	- незначительно нарушена последовательность, логика объяснения выполнения задания, выводы аргументированы и обоснованы; студент испытывает незначительные затруднения, отвечая на сопутствующие вопросы	0,3
	- значительно нарушена последовательность, логика	0

	объяснения выполнения задания (студент не может объяснить, каким образом пришел к полученному результату), выводы не могут считаться аргументированными и обоснованными; студент испытывает значительные затруднения, отвечая на сопутствующие вопросы	
	ИТОГО	3

Результаты выполнения теоретического задания и результаты выполнения практического задания суммируются. Формируется свод результатов, полученные результаты соотносятся с 5-бальной системой оценки:

Оценка		Количество баллов, набранных за выполнение теоретического и практического задания
Оценка 5 «отлично»		4,6-5
Оценка 4 «хорошо»		3,6-4,5
Оценка 3 «удовлетворительно»	3	3-3,5
Оценка 2 «неудовлетворительно»	2	≤ 2,9