

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

**Методические указания  
к выполнению практических занятий  
ПМ.03 Сопровождение и обслуживание программного обеспечения  
компьютерных систем**

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

**ОРГАНИЗАЦИЯ - РАЗРАБОТЧИК:**

Энгельсский технологический институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Саратовский государственный технический университет имени Гагарина Ю.А.»

**РАЗРАБОТЧИК:** Зотов А.А., преподаватель спецдисциплин ОСПДО

### **Пояснительная записка**

По учебному плану в соответствии с рабочей программой на изучение **ПМ.03 Сопровождение и обслуживание программного обеспечения компьютерных систем** обучающимися предусмотрено аудиторных занятий 278 часов, из них практических занятий – 122 часа. В методические указания включены 30 практических занятий по темам курса. Каждое практическое занятие содержит сведения о цели его проведения и практическом использовании результатов исследования, необходимых для проведения работы, включает краткие теоретические сведения, этапы выполнения работы.

Целью практических занятий по **ПМ.03 Сопровождение и обслуживание программного обеспечения компьютерных систем** является закрепление студентами теоретического материала по специальности и приобретение практического навыка.

**Планируемые результаты:**

#### **Перечень общих компетенций:**

Код	Наименование общих компетенций
ОК 01.	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;
ОК 02.	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях.
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

#### **Перечень профессиональных компетенций:**

**ВД 4 Сопровождение и обслуживание программного обеспечения компьютерных систем**

ПК 4.1. Осуществлять установку, настройку и обслуживание программного обеспечения компьютерных систем.

ПК 4.2 Осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем

ПК 4.3 Выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика

ПК 4.4 Обеспечивать защиту программного обеспечения компьютерных систем программными средствами.

В результате освоения профессионального модуля студент должен:

**Уметь:**

- подбирать и настраивать конфигурацию программного обеспечения компьютерных систем;
- использовать методы защиты программного обеспечения компьютерных систем;
- проводить установку программного обеспечения компьютерных систем;
- производить настройку отдельных компонентов программного обеспечения компьютерных систем;
- анализировать риски и характеристики качества программного обеспечения.

**Знать:**

- **основные методы и средства эффективного анализа функционирования программного обеспечения;**
- основные виды работ на этапе сопровождения программного обеспечения;
- основные принципы контроля конфигурации и поддержки целостности конфигурации программного обеспечения;
- средства защиты программного обеспечения в компьютерных системах.

## Практические занятия

Номер и тема раздела	Номер практического занятия	Наименование темы занятия	Кол-во часов (ауд.)
1	2	3	4
<b>Раздел 1.</b> <b>Обеспечение внедрения и поддержки программного обеспечения компьютерных систем</b> <b>МДК. 03.01</b> <b>Внедрение и поддержка компьютерных систем</b> <b>Тема 1.1</b> <b>Основные методы внедрения и анализа функционирования программного обеспечения</b>	1	Практическое занятие № 1. «Разработка сценария внедрения программного продукта для рабочего места»	2
	2	Практическое занятие № 2. «Разработка руководства оператора»	2
	3	Практическое занятие № 3. «Разработка (подготовка) документации и отчетных форм для внедрения программных средств»	2
	4	Практическое занятие № 4 Организация процесса обновления в информационной системе. Регламенты обновления	2
	5	Практическое занятие № 5 Тестирование программного обеспечения в процессе внедрения и эксплуатации	6
	6	Практическое занятие № 6 Эксплуатационная документация	6
<b>Тема 1.2</b> <b>Загрузка и установка программного обеспечения</b>	7	Практическое занятие № 7. «Измерение эксплуатационных характеристик качества программного обеспечения».	2
	8	Практическое занятие № 8 Анализ эксплуатационных характеристик качества программного обеспечения».	2
	9	Практическое занятие № 9 «Выявление проблем установки программного обеспечения»	2

Номер и тема раздела	Номер практического занятия	Наименование темы занятия	Кол-во часов (ауд.)
	10	Практическое занятие №10 Документирование проблем установки программного обеспечения»	2
	11	Практическое занятие №11 «Устранение проблем совместимости программного обеспечения»	2
	12	Практическое занятие №12 «Конфигурирование программных и аппаратных средств»	2
	13	Практическое занятие № 13 «Настройки системы и обновлений»	2
	14	Практическое занятие № 14 «Создание образа системы».	2
	15	Практическое занятие № 15 «Восстановление системы»	2
	16	Практическое занятие № 16 «Разработка модулей программного средства»	2
	17	Практическое занятие № 17 «Настройка сетевого доступа»	2
	18	Практическое занятие № 18 «Установка, адаптация и сопровождение клиентского программного обеспечения.»	2
<b>Раздел 2 Обеспечение качества компьютерных систем в процессе эксплуатации МДК. 03.02 Обеспечение качества функционирования компьютерных систем Тема 2.1</b>	19	Практическое занятие № 19 «Тестирование программных продуктов»	6
	20	Практическое занятие № 20 «Сравнение результатов тестирования с требованиями технического задания и/или спецификацией».	6
	21	Практическое занятие № 21 «Анализ рисков»	6
	22	Практическое занятие № 22 «Выявление первичных ошибок»	6
	23	Практическое занятие № 23	6

Номер и тема раздела	Номер практического занятия	Наименование темы занятия	Кол-во часов (ауд.)
Основные методы обеспечения качества функционирования		«Выявление вторичных ошибок»	
Тема 2.2 Методы и средства защиты компьютерных систем	24	Практическое занятие № 24 «Обнаружение вируса и устранение последствий его влияния»	6
	25	Практическое занятие № 25 «Установка и настройка антивируса.»	6
	26	Практическое занятие № 26 «Настройка обновлений с помощью зеркала»	6
	27	Практическое занятие № 27 «Настройка политики безопасности»	6
	28	Практическое занятие № 28 «Настройка браузера»	6
	29	Практическое занятие № 29 «Работа с реестром»	8
	30	Практическое занятие № 30 «Работа с программой восстановления файлов и очистки дисков»	9
			121

## Практическое занятие № 1

**Тема** Разработка сценария внедрения программного продукта для рабочего места

**Тема практического занятия:** Разработка сценария внедрения программного продукта для рабочего места

**Цели:** научиться определять цели и задачи внедрения программного продукта, разбиению команды разработчиков на рабочие группы.

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1.** Изучить теоретические сведения.

Теоретические сведения

*Внедрение программного обеспечения* — процесс настройки программного обеспечения под определенные условия использования, а также обучения пользователей работе с программным продуктом.

Цели внедрения программного продукта:

1. Обеспечение стабильной повторяемости результатов с заданным качеством.
2. Снижение квалификационных требований к участникам IT-проектов.
3. Сокращение сроков выполнения IT-проектов.

После определения целей, следует сконцентрироваться на действиях, на каждом из этапов работы, преимущественно на процессе исполнения

*Коллективная разработка.* Одним из основных вопросов коллективной разработки является разделение труда.

Модели разбиения коллектива на рабочие группы:

- иерархическая модель;
- матричная модель (равноправные соисполнители);
- бригада главного программиста;
- программирование в парах;
- ядерная модель;
- общинная модель разработки.

**Задание 2.** Сформулируйте цели и задачи внедрения вашего программного продукта.

**Задание 3.** Сформулируйте этапы внедрения программного продукта.

**Задание 4.** Организуйте разбиение вашего коллектива (всей группы целиком) в соответствии с классификацией разбиения коллектива на рабочие группы.



**Задание 5.** По группам обсудите и распределите обязанности и задания конкретно для каждого участника.

**Задание 6.** Обсудите, разработайте и оформите в электронном виде техническое задание разработки вашего программного продукта.

**Задание 7.** В ранее созданных вами группах распределите функции управления.

**Задание 8.** Определите ответственных и исполнителей управления.

**Задание 9.** Продумайте и опишите все процессы управления.

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 2**

**Тема** Разработка сценария внедрения программного продукта для рабочего места

**Тема практического занятия:** Разработка руководства оператора

**Цели:**

1. ознакомиться с различными ГОСТами по разработке руководства оператора;
2. научиться составлять руководство оператора.

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1.** Изучить теоретические сведения

Теоретические сведения

В соответствии с государственными стандартами, руководство оператора входит в состав комплекта эксплуатационной документации на программное обеспечение.

Роль оператора по своим функциям ближе всего к роли пользователя, однако, отличается от нее тем, что перед оператором не ставятся прикладные задачи, которые он может решить с помощью программы тем или иным способом, в том или ином порядке.

Его работа заключается в выполнении отдельных операций (согласно инструкции), то есть конкретных последовательностей действий, приводящих к конкретному результату (например, ежедневный запуск вспомогательных программ).

Наличие Руководства оператора регламентируется ГОСТ 19.101, а структура и содержание - ГОСТ 19.505.

Однако, в зависимости от сложности, назначения и области применения программного обеспечения, различные руководства оператора могут отличаться друг от друга по способу, методике и стилю изложения.

**Задание 2.** Составить руководство оператора, которое должно состоять из следующих пунктов:

- а. Титульный лист.
- б. Аннотация.
- в. *Назначение программы.* Указать сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

г. *Условия выполнения программы.* Указать условия, необходимые для выполнения программы (минимальный и (или) максимальный состав аппаратных и программных средств и т.п.).

д. *Выполнение программы.* Указать последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузки и управляет выполнением программы, а также ответы программы на эти команды.

е. *Сообщения оператору.* Привести тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия оператора (действия оператора в случае сбоя, возможности повторного запуска программы и т.п.).

### **Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

### **Практическое занятие № 3**

**Тема раздела** Разработка сценария внедрения программного продукта для рабочего места

**Тема** практического занятия : Подготовка документации и отчетных форм для внедрения программных средств

**Цель:** научиться разрабатывать документацию для внедрения программного продукта

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

#### **Последовательность выполнения работы**

Задание 1. Изучить теоретические сведения

Теоретические сведения

При разработке программных средств (ПС) создается и используется большой объем разнообразной документации.

Она необходима как средство передачи информации между разработчиками программного продукта, как средство управления разработкой, и как средство передачи пользователям информации, необходимой для применения и сопровождения.

На создание этой документации приходится большая доля стоимости программного средства.

Эту документацию можно разбить на две группы:

1. документы управления разработкой программного средства;
2. документы, входящие в состав программного средства.

Документы управления разработкой ПС *управляют* и протоколируют процессы разработки и сопровождения ПС, обеспечивая связи внутри коллектива разработчиков ПС и между коллективом разработчиков и менеджерами ПС - лицами, управляющими разработкой ПС.

Эти документы могут быть следующих типов:

- планы, оценки, расписания;
- отчеты об использовании ресурсов в процессе разработки;
- стандарты;
- рабочие документы;
- заметки и переписка.

Документы, входящие в состав ПС, описывают программы ПС как точки зрения их применения пользователями, так и точки зрения их разработчиков и сопроводителей. Эти документы будут использоваться не только на стадии эксплуатации ПС, но и на стадии разработки для управления процессом разработки (вместе с рабочими документами).

Эти документы образуют два комплекта с разным назначением:

- пользовательская документация ПС (П-документация).

- документация по сопровождению ПС (С-документация).

Пользовательская документация программных средств

Пользовательская документация объясняет пользователям, как они должны действовать, чтобы применить разрабатываемое ПС. Она необходима, если ПС предполагает какое-либо взаимодействие с пользователями.

К такой документации относятся документы, которыми должен руководствоваться пользователь при инсталляции, при применении для решения своих задач и при управлении.

Эти документы частично затрагивают вопросы сопровождения, но не касаются вопросов, связанных с модификацией программ.

Можно считать типовым составом следующий состав пользовательской документации для достаточно больших ПС:

- общее функциональное описание;
- руководство по инсталляции;
- инструкция по применению;
- справочник по применению;
- руководство по управлению.

Документация по сопровождению программных средств

Документация по сопровождению описывает ПС с точки зрения ее разработки. Эта документация необходима, если ПС предполагает изучение того, как оно устроено (сконструировано), и модернизацию его программ.

Документация по сопровождению ПС можно разбить на две группы:

1. документацию, определяющую строение программ и структур данных ПС и технологию их разработки;
2. документацию, помогающую вносить изменения в ПС.

Документация первой группы содержит итоговые документы каждого технологического этапа разработки ПС.

Она включает следующие документы:

- внешнее описание ПС;
- описание архитектуры ПС, включая внешнюю спецификацию каждой ее программы (подсистемы).
- для каждой программы ПС описание ее модульной структуры, включая внешнюю спецификацию каждого включенного в нее модуля;
- для каждого модуля спецификацию и описание его строения;
- тексты модулей на выбранном языке программирования;
- документы установления достоверности ПС.

Документация второй группы содержит *руководство по сопровождению ПС*, которое описывает особенности реализации ПС.

В нем также фиксируются, какие части ПС являются аппаратно- и программно-зависимыми.

## Документирование ППП

Создание и использование пакета прикладных программ (ППП) от формирования концепции и требований к первой версии до изъятия его из эксплуатации сопровождается документированием объектов и процессов жизненного цикла ППП.

По своему назначению документацию ППП можно классифицировать как:

1. технологическую документацию процесса разработки, включающую подробные технические описания для специалистов, ведущих проектирование, разработку и сопровождение ППП, обеспечивающую возможность отчуждения, детального освоения, развития и корректировки ими программ и баз данных на всем жизненном цикле ППП;

2. эксплуатационную (пользовательскую) документацию программного продукта, создаваемую для конечных пользователей пакета и позволяющую им осваивать и квалифицированно применять его для решения конкретных прикладных задач.

Технологическая документация включает:

- проектную документацию;
- документацию тестирования компонентов и комплексов программ;
- документацию испытаний ППП;
- документацию сопровождения и управления конфигурацией ППП.

В состав проектной документации входят:

- отчет по обследованию предметной области, для которой предназначен разрабатываемый ППП, с описанием комплекса задач;
- описание концепции проектирования;
- техническое задание на проектирование;
- план-график работ;
- спецификации эскизного и технического проекта;
- документация на разработанные программные модули пакета;
- общее описание программного обеспечения, используемого при разработке и функционировании пакета.

В состав документации тестирования входят:

- исходные данные для проведения тестирования (методы тестирования, тестовые наборы, эталонные значения, реальные ресурсы тестирования - временные, аппаратно-программные, людские, критерии полноты и качества тестирования);
- программа (сценарии) тестирования;
- журнал тестирования;
- итоговый отчет о результатах тестирования.

В состав документации испытаний входят:

- программа испытаний;
- описание методов и методик испытаний;
- протоколы испытаний;
- акт завершения работ;
- акт приемки ППП в эксплуатацию.

В состав документации сопровождения управления конфигурацией входят:

- отчеты пользователей о выявленных дефектах и предложения по корректировке программ;
- журнал выявленных дефектов и предложений по совершенствованию и развитию версии ППП;
- журнал подготовленных и утвержденных корректировок, а также реализованных изменений в новой версии пакета;
- отчет о результатах эксплуатации снятой с сопровождения версии пакета;
- журнал тиражирования и характеристик базовых версий, поддерживаемых сопровождением.

Пользовательская документация включает в себя:

- паспорт на программное средство;
- общее описание информационной системы (ИС), в составе которой будет использоваться ППП;
- руководство администратора программного средства,
- руководства оперативных пользователей с требованиями к уровню подготовки пользователя, описание функций.

**Задание 2.** Разработайте регламент внедрения программного продукта, который состоит из следующих пунктов:

1. Титульный лист
2. Общие положения, в которых перечислены основные задачи разработки регламента:
  - 2.1. Определения
  - 2.2. Участники деятельности
  - 2.3. Этапы разработки и внедрения
3. Постановка задачи и запуск проекта
4. Техническое задание должно содержать в себе, следующую информацию:
  - 4.1. цель автоматизации;
  - 4.2. наименование и краткую характеристику системы;
  - 4.3. назначение и функции предмета разработки;
  - 4.4. требования к предмету разработки, в том числе к функциональным характеристикам, надежности, справочной информации и др.;
  - 4.5. требования к видам обеспечения, требования к информационному обеспечению, условия работы;

- 4.6. порядок выполнения работ по созданию системы с указанием содержания работ;
- 4.7. особые требования к проведению приемки работ;
- 4.8. условия взаимодействия с другими проектами;
- 4.9. другая необходимая информация.
- 5. Порядок выполнения работ и внедрения программных продуктов

#### **Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

#### **Практическое занятие № 4, 5**

**Тема** Организация процесса обновления в информационной системе.  
Регламенты обновления

**Тема** Тестирование программного обеспечения в процессе внедрения и эксплуатации

**Цель:** научиться определять качество программного продукта по его характеристикам

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 180 минут

**Форма отчетности по занятию:** вывод о проделанной работе

#### **Последовательность выполнения работы**

Задание 1. Изучить теоретические сведения

Теоретические сведения

*Качество программного обеспечения* — способность программного продукта при заданных условиях удовлетворять установленным или предполагаемым потребностям.

Рекомендуется следующая общая схема процессов оценки характеристик качества программ:



- установка исходных требований для оценки - определение целей испытаний, идентификация типа метрик программного средства, выделение адекватных показателей и требуемых значений атрибутов качества;
- селекция метрик качества, установление рейтингов и уровней приоритета метрик субхарактеристик и атрибутов, выделение критериев для проведения экспертиз и измерений;
- планирование и проектирование процессов оценки характеристик и атрибутов качества в жизненном цикле программного средства;
- выполнение измерений для оценки, сравнение результатов с критериями и требованиями, обобщение и оценка результатов.

Для каждой характеристики качества рекомендуется формировать меры и шкалу измерений с выделением требуемых, допустимых и неудовлетворительных значений.

Реализация процессов оценки должна коррелировать с этапами жизненного цикла конкретного проекта программного средства в соответствии с применяемой, адаптированной версией стандарта ISO 12207.

*Функциональная пригодность* - наиболее неопределенная и объективно трудно оцениваемая субхарактеристика программного средства. Области применения, номенклатура и функции комплексов программ охватывают столь разнообразные сферы деятельности человека, что невозможно выделить и унифицировать небольшое число атрибутов для оценки и сравнения этой субхарактеристики в различных комплексах программ.

*Оценка корректности программных средств* состоит в формальном определении степени соответствия комплекса реализованных программ исходным требованиям контракта, технического задания и спецификаций на программное средство и его компоненты. Путем верификации должно быть определено соответствие исходным требованиям всей совокупности к компонентам комплекса программ, вплоть до модулей и текстов программ и описаний данных.

*Оценка способности к взаимодействию* состоит в определении качества совместной работы компонентов программных средств и баз данных с другими прикладными системами и компонентами на различных вычислительных платформах, а также взаимодействия с пользователями в стиле, удобном для перехода от одной вычислительной системы к другой с подобными функциями.

*Оценка защищенности программных средств* включает определение полноты использования доступных методов и средств защиты программного средства от потенциальных угроз и достигнутой при этом безопасности функционирования информационной системы.

Наиболее широко и детально методологические и системные задачи оценки комплексной защиты информационных систем изложены в трех частях стандарта ISO 15408:1999-1--3 «Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий».

*Оценка надежности* - измерение количественных метрик атрибутов субхарактеристик в использовании: завершенности, устойчивости к дефектам, восстанавливаемости и доступности/готовности.

*Потребность в ресурсах памяти и производительности* компьютера в процессе решения задач значительно изменяется в зависимости от состава и объема исходных данных.

Для корректного определения предельной пропускной способности информационной системы с данным программным средством нужно измерить экстремальные и средние значения длительностей исполнения функциональных групп программ и маршруты, на которых они достигаются.

Если предварительно в процессе проектирования производительность компьютера не оценивалась, то, скорее всего, понадобится большая доработка или даже замена компьютера на более быстродействующий.

*Оценка практичности* программных средств проводится экспертами и включает определение понятности, простоты использования, изучаемости и привлекательности программного средства.

В основном это качественная (и субъективная) оценка в баллах, однако некоторые атрибуты можно оценить количественно по трудоемкости и длительности выполнения операций при использовании программного средства, а также по объему документации, необходимой для их изучения.

*Сопровождаемость* можно оценивать полнотой и достоверностью документации о состояниях программного средства и его компонентов, всех предполагаемых и выполненных изменениях, позволяющей установить текущее состояние версий программ в любой момент времени и историю их развития.

Она должна определять стратегию, стандарты, процедуры, распределение ресурсов и планы создания, изменения и применения документов на программы и данные.

*Оценка мобильности* - качественное определение экспертами адаптируемости, простоты установки, совместимости и замещаемости программ, выражаемое в баллах.

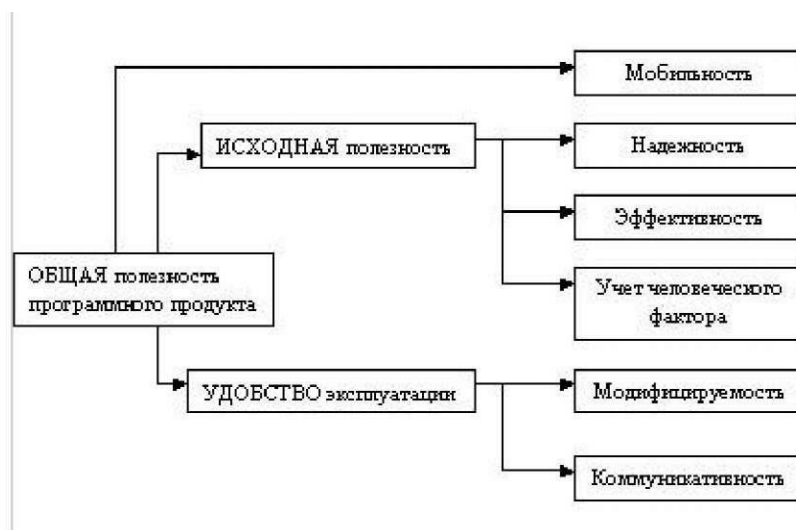
Количественно эту характеристику программного средства и совокупность ее атрибутов можно (и целесообразно) оценить в экономических показателях: стоимости, трудоемкости и длительности реализации процедур переноса на иные платформы определенной совокупности программ и данных.

## **Задание 2. Выбрать вариант задания для своей группы**

### **Варианты заданий**

№ вар ианта	Наименование
1.	Программный продукт «Аптечная система Апрель»
2.	Программный продукт «Лаборатория Касперского »
3.	Программный продукт «Юрайт»
4.	Программный продукт «Book.ru»

**Задание 3. Согласно выбранному варианту охарактеризуйте полезность программного продукта (рисунок 1)**



**Задание 4. Опишите программное обеспечение по следующим признакам:**

- 3.1. алгоритмическая сложность (логика алгоритмов обработки информации);
  - 3.2. состав и глубина проработки реализованных функций обработки;
  - 3.3. полнота и системность функций обработки;
  - 3.4. объем файлов программ;
  - 3.5. требования к операционной системе и техническим средствам обработки со стороны программного средства;
  - 3.6. объем дисковой памяти;
  - 3.7. размер оперативной памяти для запуска программ;
  - 3.8. тип процессора;
  - 3.9. версия операционной системы.
4. Сделайте вывод о качестве программного продукта для пользователя

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 6**

**Тема** Эксплуатационная документация

**Тема** Измерение эксплуатационных характеристик качества программного обеспечения.

**Цель:** научиться выявлять ошибки программного обеспечения

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретические сведения**

Теоретические сведения

Источниками ошибок в программном обеспечении являются специалисты - конкретные люди с их индивидуальными особенностями, квалификацией, талантом и опытом.

В общем случае отказ программного обеспечения можно определить, как:

- прекращение функционирования программы на время превышающее заданный порог;
- прекращение функционирования программы на время не превышающее заданный порог, но с потерей всех или части обрабатываемых данных;
- прекращение функционирования программы, потребовавшее перезагрузки ЭВМ, на которой функционирует программное обеспечение.

**Задание 2. Проанализируйте примеры, масштабных инцидентов, произошедших из-за ошибки программного обеспечения.**

1. В июне 1996 года специалисты Европейского космического агентства осуществляли запуск ракеты Ariane 5. Ошибка в программном обеспечении для модуля управления привела к старту процесса самоуничтожения - через 37 секунд полета ракета взорвалась.

2. В 2014 году крупный сбой в программном обеспечении чуть не обанкротил корпорацию Knight Capital. Фирма менее чем за час потеряла полмиллиарда долларов - система начала несанкционированно покупать и продавать большое количество акций. В итоге за два дня акции упали в цене на 75%.

3. В 1980-годы несколько пациентов погибли после получения слишком большой дозы облучения рентгеновским аппаратом Therac-25 (лучевая терапия).

4. Летом 2013 года произошло отключение серверов американской компании Amazon (самая известная компания в мире по продаже различных товаров и услуг через Интернет). Это привело к потере файлов пользователей, хранившихся в сетевом хранилище. Авария, первоначально вызванная грозой, неожиданно усугубилась ошибками в используемом программном обеспечении. Это быстро привело к каскадному сбою.

5. Массовое отключение электричества в 2003 году в северо-восточной части США произошло из-за локальной аварии, которая не была зафиксирована программным обеспечением General Electric Energy. Отсутствие реакции на локальный сбой привело к каскадному отключению электроэнергии.

6. В 2014 году из-за ошибки в программе была заблокирована работа всех самолетов авиакомпании American Airlines. Сбой возник в системе бронирования билетов - проводилась работа по объединению программных платформ нескольких компаний.

**Задание 3 Охарактеризуйте их по категории тяжести и последствиям, оформите в виде таблицы**

Категории тяжести ошибки в программном обеспечении

Номер категории ошибки	Наименование категории тяжести ошибки	Описание последствий проявления ошибки

**Задание 4. Подумайте, как можно было избежать подобных ошибок**

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 7, 8**

**Тема** Измерение и анализ эксплуатационных характеристик качества программного обеспечения.

**Цель:** выработка практических навыков измерения и анализа эксплуатационных характеристик качества программного обеспечения.

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 180 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретические сведения теоретические сведения.**

Процесс оценки качества программного обеспечения осуществляется для каждой фазы его жизненного цикла и включает:

- выбор совокупности (номенклатуры) показателей качества оцениваемого программного средства;
- определение значений этих показателей;
- сравнение полученных значений с базовыми значениями показателей.

Под жизненным циклом программного обеспечения понимается период времени с момента начала предпроектного обследования до момента полного выхода программы из употребления пользователями.

Весь период жизненного цикла программного обеспечения делится на следующие временные промежутки или фазы.

1. Анализ — этап определения требований к программному обеспечению, спецификация требований и формирования технического задания на проектирование программы.
2. Проектирование — этап разработки технического проекта.
3. Реализация — этап разработки программного обеспечения, средств тестирования и документации.
4. Тестирование — этап испытания программного обеспечения и устранение недостатков.
5. Изготовление — этап преобразования программного обеспечения в форму, готовую для поставки; завершение формирования документации.
6. Внедрение — этап подтверждения стабильной работы программного обеспечения; ввод в стадию активного применения.
7. Эксплуатация — этап применения программного обеспечения по назначению.

8. Сопровождение — этап устранения дефектов в процессе эксплуатации; усовершенствование, оптимизация и модификация используемого программного обеспечения при условии сохранности целостности программного продукта.

Оценка качества программного обеспечения на всех фазах жизненного цикла осуществляется на основе четырёхуровневой системы показателей.

Показатели первого уровня (факторы качества) характеризуют потребителски-ориентированные свойства программных средств, которые соответствуют потребностям пользователей. Факторы качества, собственно, определяют наиболее значимые (с точки зрения использования) свойства программ. Для оценки качества программного обеспечения используют следующие факторы:

- надёжность;
- сопровождаемость;
- удобство применения;
- эффективность;
- универсальность;
- корректность.

Каждый фактор представляет собой интегральную оценку, которой соответствует несколько критериев качества (комплексных показателей второго уровня).

#### **Задание**

В соответствии с приведенными ниже критериями оценить характеристики качества программного продукта.

#### **Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## Практическое занятие № 9, 10

**Тема** Выявление и документирование проблем установки программного обеспечения.

**Цель:** Научиться устанавливать ПО и выявлять проблемы установки.

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретические сведения**

Установка программного обеспечения, инсталляция — процесс установки программного обеспечения на компьютер конечного пользователя. Выполняется особой программой (пакетным менеджером), присутствующей в операционной системе (например, RPM, APT или dpkg в Linux, Установщик Windows в Microsoft Windows), или же входящим в состав самого программного обеспечения средством установки.

Программа установки, установщик или инсталлятор (англ. installer) — это программа, которая устанавливает программное обеспечение на компьютер. Некоторые установщики специально сделаны для установки содержащихся в них файлов, другие же являются универсальными и работают, считывая содержимое комплекта программного обеспечения, которое необходимо установить.

Распространенные программы установки семейства Windows

Наиболее популярным форматом для семейства Windows NT является установочный пакет MSI, который устанавливается посредством Установщика Windows. Компании, производящие средства для создания инсталляторов: InstallShield (InstallShield Wizard), Macrovision (InstallAnywhere), Wise Solutions, Inc., SetupBuilder, InstallBuilders (Smart Install Maker). Большинство из этих средств могут создавать как пакеты MSI, так и свои собственные пакеты.

Бесплатными альтернативами являются NSIS, Actual Installer Free, Clickteam Install Creator, Inno Setup, Install Simple, а также инструментальные средства от Microsoft (WiX).

**Задание 2:**

Изучите и выполните (по возможности) следующие инструкции:

Создание пользовательского пакета MSI

Чтобы во время этого процесса не возникало проблем, следует начинать работу с «чистым» компьютером.

- Запустите «чистый» компьютер или компьютер, представляющий компьютеры в вашей сети.



- Запустите программу Discover, чтобы зафиксировать конфигурацию программного обеспечения этого компьютера (снимок состояния «до установки»).
- Установите программу на компьютере, для которого была создана копия состояния «до установки».
- Перезагрузите компьютер.
- Запустите новую программу, чтобы убедиться в ее работоспособности.
- Закройте программу.
- Запустите программу Discover и создайте копию состояния «после установки» для новой конфигурации компьютера. Программа Discover сравнивает две копии состояния и фиксирует изменения. После этого она создает пакет установщика Майкрософт с инструкциями по установке данной программы на подобном компьютере.
- (Не обязательно) С помощью консоли Veritas Software Console измените настройки пакета установщика Майкрософт.
- Очистите контрольный компьютер, чтобы подготовиться к новому запуску программы Discover.
- (Не обязательно) Выполните пробную установку программы на рабочих станциях, не задействованных в рабочем процессе.

#### **Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## Практическое занятие № 11

**Тема** Устранение проблем совместимости программного обеспечения

**Продолжительность проведения:** 2 часа.

**Цель:** изучить решение проблем совместимости программного обеспечения

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретические сведения**

Теоретические сведения

*Программное обеспечение* - совокупность программ, выполняемых вычислительной системой, так же относят все области деятельности по проектированию и разработке программного обеспечения.

Совместимость - способность различных объектов, аппаратных или программных компонентов взаимодействовать друг с другом.

По отношению к персональному компьютеру можно выделить совместимости:

1. Аппаратная (техническая) - способность одного устройства работать с узлами другого устройства

2. Информационная - способность 2-х или более систем адекватно воспринимать одинаково представленные данные

3. Программная - способность выполнения одинаковых программ с получением одних и тех же результатов

При наличии всех трех совместимостей устройств без ограничения, для конечных пользователей, говорят о полной совместимости этих устройств.

**Задание. Используя руководство пользователя по программе варианта исследуйте совместимость данной программы с аппаратными и программными платформами.**

№ вар ианта	Наименование
5.	Программный продукт «Аптечная система Апрель»
6.	Программный продукт «Лаборатория Касперского »
7.	Программный продукт «Юрайт»
8.	Программный продукт «Book.ru»

**Задание 2. Заполните таблицу**

Аппаратная платформа	Программная платформа	Совместимость программы (работоспособность, функционал)

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 не существенных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 12**

**Тема** Конфигурирование программных и аппаратных средств

**Цель:** научиться управлять конфигурацией программного продукта

**Материально-техническое и комплексно-методическое**

**обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретические сведения**

Теоретические сведения

Управление конфигурацией — это один из вспомогательных процессов, поддерживающих основные процессы жизненного цикла программного обеспечения, прежде всего процессы разработки и сопровождения программного обеспечения.

Под конфигурацией программного обеспечения понимается совокупность его функциональных и физических характеристик, установленных в технической документации и реализованных в программном обеспечении.

В крупном проекте большие объёмы информации меняются очень быстро и неконтролируемые изменения могут быстро ввергнуть проект в хаос.

Работая над программным проектом, группа программистов, тестеров и менеджеров сталкивается с проблемой отслеживания версий программ, внесения в них изменений.

Чем больше проект, тем больше времени разработчики тратят на согласования изменений в исходных текстах и получения работающих версий программного продукта.

Управление конфигурацией позволяет организовать, систематически учитывать и контролировать внесение изменений на всех стадиях жизненного цикла.

При групповой разработке сложных программного обеспечения, состоящих из многих компонентов, каждый из которых может иметь разновидности или версии, необходимо:

— выделить набор объектов, определяющих структуру будущей системы, чтобы затем контролировать их состояния и ход работ по каждому из них. Такими объектами могут быть функционально-логическая модель системы, реляционная модель базы данных, модули прототипов системы (экраны, меню, отчеты, тексты процедур или классов), системные и программные спецификации, документация, планы проведения тестирования, спецификации тестовых процедур;

— контролировать запросы на доработку модуля, сообщения о найденной ошибке или неисправности оборудования, запросы на модификацию оборудования или программного обеспечения, задания на установку рабочего места, задания разработчику, аналитику и т.п., так как эти объекты влияют на состояние текущих версий других объектов и относятся к сфере управления изменениями;

— вести журнал всех изменений, внесенных в систему в процессе разработки или сопровождения;

— вести полный и достоверный архив всех версий всех объектов системы;

— контролировать состояние и развитие коллективно используемых компонентов программного обеспечения и их версий, учитывая связи компонентов системы для согласования между собой измененных частей;

— обеспечивать адекватность реально изменяющихся компонентов и их комплектной документации;

— проводить оценку конфигурации — оценивать функциональную полноту компонентов программного обеспечения, а также соответствие их физического состояния текущему техническому описанию;

— изготавливать эталонные копии программного обеспечения и документации, хранить и поставлять их пользователям в соответствии с порядком, принятым в организации. Это упрощает выпуск и поставку программного обеспечения;

— обеспечивать развитие всей системы, ограничивая усложнение проекта.

Общие принципы и рекомендации конфигурационного учета, планирования и управления конфигурациями программного обеспечения отражены в проекте стандарта ISO 12207-2.

**Задание 1. Используя руководство пользователя по программе варианта исследуйте конфигурацию данного программного обеспечения.**

№ вар ианта	Наименование
1.	Программный продукт «Аптечная система Апрель»
2.	Программный продукт «Лаборатория Касперского »
3.	Программный продукт «Юрайт»
4.	Программный продукт «Book.ru»

**Задание 2. Исследовать конфигурацию рабочего ПК для установки выбранного программного обеспечения и заполнить таблицу**

Элементы системного блока

Название элемента	Фирма производитель модель	Основные и характеристики	Примечание
Процессор			
Блок питания			
Материнская плата			

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 13**

### **Тема Настройка системы обновлений**

**Цель:** научиться производить настройку системы обновлений в операционной системе Windows

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

#### **Последовательность выполнения работы**

##### **Задание 1. Изучите теоретические сведения**

Теоретические сведения

Когда разработчикам становится известно о критических уязвимостях в продуктах, брешах в безопасности, когда они вносят незначительные усовершенствования в программы, добавляют новые функции или решают проблемы совместимости, они выпускают обновления.

Своевременное обновление программного обеспечения гарантирует, что вы используете наиболее актуальную версию без багов и уязвимостей.

##### **Задание. Выполните следующий набор действий**

1. Открыть диалоговое окно «Выполнить», используя комбинацию
  2. В окно ввести следующий текст: Rundll32 newdev.dll, Device Internet Setting Ui после чего щелкнуть на кнопку Enter на клавиатуре
  3. В появившемся окне сперва следует выбрать пункт «Нет, предоставить возможность выбора», после чего — «Никогда не устанавливать драйверы из Центра обновления Windows».
  4. Сохранить изменения.
- Выполнив эту последовательность действий, вы запретите системе искать драйверы в Интернете, а предпишете устанавливать их исключительно из памяти персонального компьютера.
5. Есть еще один метод ограничения обновлений. Для этого необходимо настроить интернет-подключение через «лимитный» Wi-Fi канал.
  6. Открыть меню «Параметры» и зайти в подраздел «Сеть и интернет».
  7. Затем в подразделе беспроводного подключения зайти в «Дополнительные параметры» и включить опцию «Задать как лимитное подключение».
  8. Использование реестра Windows 10. С помощью реестра можно полностью отключить скачивание обновляющих пакетов.
  9. Введите regedit в диалоговом окне «Выполнить» (вход + R) или в поисковой строке и щелкнуть по клавише Enter через
- Откроется реестровый редактор.

10. Зайти в следующий подраздел:  
HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Microsoft\Windows

11. После чего необходимо создать новый подраздел в системном каталоге Windows, назвав его Windows Update. В нем следует создать ещё один раздел, назвав его AU.

Затем следует выбрать подраздел AU и сформировать там параметр DWORD, назвав его noautoupdate и присвоив ему значение «1».

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя



## **Практическое занятие № 14, 15**

**Тема:** Создание образа системы. Восстановление системы

**Продолжительность проведения:** 4 часа.

**Цель:** научиться создавать образ системы в операционной системе Windows

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретический материал**

Теоретические сведения

Если компьютер с операционной системой Windows вдруг перестал работать, в нем что-то сломалось или «полетели» программы, то возможно потребуется восстановление системы. Бывает и так, что операционная система настолько сбилась, «испортилась», что компьютер даже перестал загружаться.

Для загрузки компьютера, у которого операционная система перестала загружаться, служит диск восстановления системы Windows. Но этого диска недостаточно для полного «ремонта» операционной системы. С его помощью можно лишь загрузить компьютер с тем, чтобы начать процедуру восстановления системы.

Поэтому существует комплект программных средств, позволяющий восстановить работоспособность операционной системы, который состоит из:

1. диска восстановления системы;
2. образа системы;
3. архивной копии системы и файлов пользователей.

**Задание. Создайте образ диска, используя следующую схему**

Создание образа системы

1. Подготовьте носитель в виде DVD-диска, на котором будет храниться образ системы.
2. Запустите «Панель управления» «Система и безопасность» «Архивация и восстановление».
3. В открывшемся окне «Резервное копирование и восстановление» Нажимаем «Создать диск восстановления системы»
4. Выбираем где будет храниться образ системы и запускаем процедуру создания диска восстановления системы нажав «Далее»
5. Программа создания образа системы сообщает нам о размере создаваемого образа системы и предлагает подтвердить параметры архивации образа системы. Нажимаем «Архивировать»
6. Если на диске закончилось место, следует сменить его на другой

диск.

7. Далее появится вопрос о возможности стереть все данные с DVD-диска, нажмите «Форматировать».

8. После завершения процесса система запрашивает создание диска восстановления системы. Нажимаем «Нет»

## **II. Восстановление системы**

1. Подготовьте носитель для восстановления системы

2. Откройте «Панель управления» «Система и безопасность» «Архивация и восстановление» «Резервное копирование и восстановление» «Создать диск восстановления системы»

3. Выбираем устройство для записи и нажимаем «Создать диск»

4. После того, как на диск запишется информация нажимаем «Закрыть»

В конце нажимаем «Ок», чтобы окончательно закрыть окно программы для создания диска восстановления системы

### **Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 16**

**Тема** Разработка модулей программного средства

**Цель:** научиться разрабатывать модули программного средства

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

### **Задание 1. Изучите теоретические сведения**

Теоретические сведения

Практически все сложное программное обеспечение имеет модульную структуру, то есть состоит из программных модулей, при этом разработка программных модулей программного обеспечения может быть выполнена:

1. Как составляющая часть процесса разработки программного обеспечения. В данном случае разработка модулей, в зависимости от проекта, может вестись последовательно и (или) параллельно, но всегда завершается их объединением в единую систему.

2. В процессе доработки, усовершенствования или модернизации ранее созданного программного обеспечения. Модульная конструкция программного обеспечения позволяет вносить изменения в отдельные модули, удалять некоторые из них или дополнять систему новыми модулями.

При этом разработка программных модулей программного обеспечения и их внедрение, как и изменение или удаление, не заставляет переделывать всю систему в целом.

В программном комплексе созданные модули могут представлять собой:

- законченные компоненты;
- функциональные группы;
- пакеты программ;
- единый программный комплекс - законченный программный

продукт конкретного целевого назначения.

Каждый программный модуль может быть спроектирован и разработан отдельно от других модулей. Более того, он может представлять собой самостоятельный программный продукт и быть включенным в структуру разных программ, не обязательно тех, для которых разрабатывался.

**Задание 2. Создайте простейший модуль своего программного продукта на формах .**

**Задание 3. Проведите отладку и тестирование модуля.**

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 17**

### **Тема Настройка сетевого доступа**

**Цель:** научиться настраивать сетевой доступ к дискам (папкам) в системе Windows

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

#### **Последовательность выполнения работы**

##### **Задание 1. Изучите теоретические сведения**

Теоретические сведения

При работе с локальной сетью или с компьютерами интрасети организации приходится настраивать общие папки, для того чтобы пользователи могли просматривать, изменять и создавать файлы и папки для компьютеров, которые расположены в одной локальной группе или в одном домене.

В настройке общего доступа к файлам и папкам нет ничего сложного, но в связи с тем, что для открытия общего доступа нужны права администратора, не всем пользователям вашей сети будет предоставлена такая возможность.

Для того чтобы пользователи могли просматривать содержимое локальной сети и иметь доступ к компьютерам и устройствам можно включить сетевое обнаружение.

Если к каждому компьютеру сети не подключен локальный принтер, то придется открывать общий доступ к принтерам для того чтобы пользователи могли распечатывать свою документацию.

Поэтому можно предоставить общий доступ к ресурсам компьютера, как для всех пользователей, так и для тех пользователей, учетные данные которых имеются на компьютере, предоставляющем общий доступ к файлам и папкам.

##### **Задание 2. Согласно следующей инструкции настройте сетевой доступ к папке**

1. Кликните правой кнопкой мыши на иконке диска (папки) и в выпадающем меню выберите «Свойства»;

2. Перейдите на вкладку «Доступ» и нажмите на кнопку «Расширенная настройка...»; в открывшемся окне, установите галочку напротив надписи: «Открыть доступ к этой папке», в поле «Имя общего ресурса» введите сетевое имя (необходимо для удобства идентификации ресурса в домашней локальной сети), после чего кликните «Применить».

3. На следующем этапе настройки, нужно определить, кто будет иметь доступ к диску (папке) и какими правами будет наделен. Кликните кнопку «Разрешения» и в открывшемся окне в списке «Группы или

пользователи» выделите «Все» и установите галочку «Полный доступ», затем кликните «Применить» и «ОК».

4. Если группа «Все» отсутствует, ее необходимо включить в список. Кликаем «Добавить» и в открывшемся окне, в поле «Введите имена выбираемых объектов» вбиваем «Все» (именно так, как указано в кавычках), затем «ОК».

5. Далее, действуем по уже описанному алгоритму.

6. Если, вы желаете ограничить права пользователей при работе с сетевым ресурсом, к примеру, запретить удаление файлов и папок — необходимо воспользоваться вкладкой «Безопасность». Выделите группу «Все» и нажмите кнопку «Дополнительно».

7. В открывшемся окне также отметьте пункт «Все» и кликните «Изменить», после чего справа вверху щелкните надпись: «Отображение дополнительных разрешений».

8. Отключите ненужные разрешения, путем снятия галочек и подтвердите действия нажав «ОК».

9. При первом обращении к сетевому ресурсу, система может предложить ввести логин/пароль, просто введите имя компьютера (на сетевой диск которого пытаетесь зайти) и пароль (если он отсутствует, оставьте поле пустым), не забудьте поставить галочку «Запомнить учетные данные». Если пароля нет, а войти не удастся — отключите доступ по паролю (смотрите пункт «Устранение неполадок при настройке общего доступа»).

10. Если система выдает предупреждение о недоступности сетевого ресурса и «ругается» на настройки, нужно проверить параметры общего доступа для различных сетевых профилей на том компьютере, на сетевой ресурс которого пытаетесь зайти. Пройдите по цепочке: «Пуск» — «Параметры» — «Сеть и интернет» — «Ethernet» или «Wi-Fi» (зависит от протокола подключения к локальной сети) — «Изменения расширенных параметров общего доступа».

11. Выставьте параметры

- а. «Включить сетевое обнаружение»
- б. «Включить общий доступ к файлам и принтерам»
- в. «Разрешить Windows управлять подключениями домашней группы»
- г. «Включить общий доступ, чтобы сетевые пользователи могли читать и записывать файлы в общих папках»
- д. «Использовать 128 - битное шифрование для защиты подключений общего доступа»
- е. «Отключить общий доступ с парольной защитой»

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 18**

**Тема** Установка, адаптация и сопровождение клиентского программного обеспечения

**Цель:** научиться настраивать клиентское программное обеспечения

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретические сведения**

Теоретические сведения

**Виды клиентского программного обеспечения. Установка, адаптация и сопровождение клиентского программного обеспечения**

**Программное обеспечение** – это совокупность программных и документальных средств для создания и эксплуатации систем обработки данных средствами вычислительной техники.

**Клиентское ПО** – это ПО, осуществляющее доступ к ресурсам, которые предоставляет сервер.

**Установка и сопровождение клиентского программного обеспечения**

**Подготовка компьютера перед установкой**

Убедится, что конфигурация компьютера отвечает минимальным требованиям ПО.

Проверить наличия свободного места на жёстком диске, если его не хватает, освободить.

Отключить антивирусные программы.

Остановить выполнение посторонних программ.

Установка различного ПО может существенно различаться, но чаще всего установка приложения сводится к стандартному набору действий в которых вам помогает мастер установки.

В случае если программа уже была когда то проинсталирована на ПК мастер установки спросит, что делать со старой версией удалить, заменить или восстановить.

После этого можно будет указать путь установки программы и завершить установку выполнив все действия указанные мастером.

Завершающим этапом установки, чаще всего, становится регистрация программного продукта.



## **Поддержка программного обеспечения**

### ***Пакеты исправлений и заплатки***

Время от времени производители программного обеспечения выпускают исправления к своим программным продуктам. Как правило, исправления корректируют выявленную проблему, а затраты на их тестирование могут варьироваться в зависимости от программного продукта.

Протестировав исправление и приняв решение о его развертывании, администратор копирует необходимые файлы в точку распространения ПО, производя замену старых файлов. Производители программного обеспечения распространяют обновления либо в виде нового пакета установщика, либо в качестве исправления установщика. В случае поставки нового пакета установщика администратор просто заменяет пакет, размещенный в точке распространения ПО. Если же поставляется исправление установщика, администратор следует указаниям производителя программного продукта по применению MSP-файла к существующему пакету, содержащему приложение.

После обновления файлов в точке распространения ПО администратор открывает расширение Установка программ (Software Installation) в объекте групповой политики, который управляет существующим программным обеспечением, и щелкает правой кнопкой мыши по исправляемому приложению для вызова контекстного меню. Он выбирает команду Развернуть приложение заново (Redeploy application), обеспечивающую копирование исправленных файлов при следующем применении групповой политики к тем пользователям, которые уже установили данное приложение.

### ***Пакеты обновления***

Между пакетами обновления и исправлениями нет больших отличий. Как правило, в состав пакета обновления входят несколько уже протестированных исправлений. Пакеты обновления распространяются реже, чем исправления, но чаще, чем полные обновления программного продукта.

Рекомендуется распространять пакет обновления в качестве исправления, если он содержит небольшое количество файлов. Распространение пакета с большим количеством файлов и управление им рекомендуется рассматривать в качестве полного обновления программного продукта.

В любом случае, следуйте инструкциям производителя заплатки и тестируйте ее в лаборатории или на небольшой группе пользователей, прежде чем осуществлять распространение заплатки для всех пользователей и компьютеров, управляемых заданным объектом групповой политики.

Пакеты обновлений подготовлены с учетом возможности их управления при помощи компонента установки и поддержки ПО, который является частью технологии IntelliMirror.

### **Обновления**

Обновление подразумевает замену значительного числа файлов. Поскольку производятся большие изменения в программном продукте, обновление получает другой номер версии.

Производитель предоставляет новую версию в пакете установщика ПО, с заложенной в него информацией о том, какие более ранние версии он может обновлять. В пакете также должен содержаться сценарий процесса обновления, в котором описано, какие файлы должны быть заменены, удалены и добавлены.

Процесс обновления начинается с размещения файлов программного продукта (пакетов установщика ПО и файлов преобразования для них) в точке распространения ПО. Затем администратор назначает или публикует новую версию в расширении Установка программ (Software Installation). При этом администратор задает преемственную связь между версиями, если это необходимо. Если пакету установщика ПО с новой версией известно об обновляемом приложении, расширение Установка программ автоматически устанавливает преемственную связь между версиями.

Администраторам необходимо определиться с тем, будет обновление обязательным (немедленно вступит в силу для всех пользователей, работающих с текущей версией) или необязательным (пользователи смогут установить новую версию, когда сочтут это нужным).

### **Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 19**

**Тема** Тестирование программных продуктов

**Цели:** ознакомление с методами тестирования программного продукта

**Материально-техническое и комплексно-методическое**

**обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретические сведения**

Теоретические сведения

Тестирование является завершающим этапом разработки программного продукта. В узком смысле цель тестирования состоит в обнаружении ошибок, цель же отладки - не только в обнаружении, но и в устранении ошибок. Однако ограничиться только отладкой программы, если есть уверенность в том, что все ошибки в ней устранены, нельзя. Цели у отладки и испытания разные.

Полностью отлаженная программа может не обладать определенными потребительскими свойствами и тем самым быть непригодной к использованию по своему назначению.

Не может служить альтернативой испытанию и проверка работоспособности программы на контрольном примере, так как программа, работоспособная в условиях контрольного примера, может оказаться неработоспособной в других условиях применения.

Попытки охватить контрольным примером все предполагаемые условия функционирования сводятся в конечном счете к тем же испытаниям.

В соответствии с ГОСТ 19.004 - 80 под испытанием программ понимают установление соответствия программы заданным требованиям и программным документам. Это определение построено на предположении, что в техническом задании на разработку программы определены все требования, обеспечение которых гарантирует пригодность программы к использованию по своему назначению.

Методы тестирования:

*Восходящее тестирование* - программа собирается и тестируется снизу-вверх.

*Нисходящее тестирование* - программа собирается и тестируется сверху вниз. Изолировано тестируется только головной модуль.

*Метод большого скачка* - каждый модуль тестируется автономно. По окончании тестирования модулей они интегрируются в систему все сразу.

*Метод сэндвича* - представляет собой компромисс между восходящим и нисходящим подходами. При использовании этого метода одновременно начинают восходящее и нисходящее тестирование, собирая программу как снизу, так и сверху и встречаясь, в конце концов, где-то в середине. Точка

встречи зависит от конкретной тестируемой программы и должна быть заранее определена при изучении ее структуры.

**Задание 2. Выбрать методы тестирования программного продукта.**

**Задание 3. Разработать план тестирования программного продукта.**

**Задание 4. Провести тестирование программы и представить результаты в виде таблицы**

Результаты тестирования

Тест (значения для входных данных)	Ожидаемый результат (значения для выходных данных)	Фактический результат (полученные значения выходных данных)	Результат тестирования (успешно/неуспешно)

4. Выработать рекомендации для корректировки тестируемой программы.

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 20**

**Тема** Сравнение результатов тестирования с требованиями технического задания и/или спецификацией

### **Цели:**

1. ознакомиться документацией на разработку программного обеспечения;
2. сравнить результаты тестирования с требованиями к программным продуктам.

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

### **Последовательность выполнения работы**

#### **Задание 1. Изучите теоретические сведения**

Теоретические сведения

*Техническое задание* - это основной исходный документ для разработки программного обеспечения, в котором формулируются основные цели разработки, список принципиальных требований к продукту, определяются сроки и этапы разработки и регламентируется процесс приемно-сдаточных испытаний.

Этот документ содержит основные требования заказчика, исходные данные для разработки, указываются назначение продукта, область его применения, стадии разработки различной документации, её состав, сроки исполнения и т. д., а также особые требования, обусловленные спецификой проекта либо условиями его эксплуатации.

Основой технического задания, как документа определяющего характеристики разрабатываемого программного обеспечения и функциональную составляющую программного обеспечения являются:

исходные данные и требуемые результаты, которые определяют функции разрабатываемого программного обеспечения;

среда (программная и аппаратная), в которой разрабатываемое программное обеспечение будет функционировать, может быть задана, а может выбираться для обеспечения требуемых параметров;

возможное взаимодействие с другим программным обеспечением и/или конкретными техническими средствами - также может быть определено, а может выбираться исходя из набора выполняемых функций.

*Спецификация требований программного обеспечения* - это документ, который содержит законченное описание поведения программы, которую требуется разработать.

Включает ряд пользовательских сценариев, которые описывают все варианты взаимодействия между пользователями и программным обеспечением.

Пользовательские сценарии являются средством представления функциональных требований. В дополнение к пользовательским сценариям, спецификация также содержит нефункциональные требования, которые налагают ограничения на дизайн или реализацию (такие как требования производительности, стандарты качества, или проектные ограничения).

Состоит из следующих разделов: введение, общее описание, функции системы, требования к данным, требования к внешним интерфейсам, атрибуты качества

**Задание 2. Составьте техническое задание, которое будет содержать необходимые требования к программному продукту**

**Задание 3. Составьте спецификацию на разработку программного продукта**

**Задание 4. Сравните результаты тестирования из предыдущей работы с требованиями технического задания**

**Задание 5. Сравните результаты тестирования из предыдущей работы с требованиями спецификации**

**Критерии оценивания:**

2. **Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

3. **Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

4. **Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

5. **Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 21**

### **Тема Анализ рисков**

**Цель:** научиться производить анализ рисков программного обеспечения

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

#### **Последовательность выполнения работы**

##### **Задание 1. Изучите теоретические сведения**

Теоретические сведения

*Управление рисками* - это одна из самых динамично развивающихся видов деятельности и большинство крупных компаний имеют в штате специалистов по управлению ими. Существует развитый инструментарий для минимизации потерь от воздействия неблагоприятных факторов как внутренней, так и внешней среды.

Полученные в работе результаты по выявлению, описанию и классификации рисков разработки программного обеспечения являются начальным этапом управления этими рисками. В дальнейшем необходимо оценить риски выбранными измерителями и выбрать управляющие стратегии, после чего оценить их воздействие.

*Риски плохого взаимодействия между заказчиком и исполнителем* - это риски связанные с отсутствием коммуникации между исполнителем и заказчиком или их представителями. Недостаточное обсуждение задач или архитектуры может негативно сказаться на разрабатываемом программном обеспечении.

*Риски управления проектом* - это риски, связанные с отсутствием навыков проектного менеджмента у менеджера проекта, а также с отсутствием интереса или мотивации у него. Сама по себе уже хорошо отлаженная система управления рисками может являться эффективным средством для того чтобы определить такого рода риски, так как позволяет идентифицировать проблему и выработать решение.

*Риски, связанные с недостаточной осведомлённостью управляющего проектом о точном состоянии проекта* - это вид рисков, связанных с отсутствием обратной связи. Он возникает, когда проектный менеджер не выстроил рабочий процесс таким образом, чтобы контролировать ход выполнения проекта на всех его этапах.

*Риски планирования* - это риски, которые могут быть связаны с отсутствием навыков планирования по проекту как менеджером, так и исполнителями, если они готовят информацию о сроках выполнения работ.

*Риски отсутствия системы контроля* - обусловлены большим количеством аспектов в области проектного менеджмента при разработке программного обеспечения, когда сложно учесть все возможные ситуации.

*Риск появления новых требований* возникает в процессе разработки программного обеспечения, когда появляются всё новые и новые требования, которые отодвигают сроки и оценку конкретных задач.

*Риск противоречивости в требованиях (декомпозиция спецификации)* - это риски связанные с выявлением противоречивости в требованиях заказчика на этапе программирования или интеграции проекта.

*Риски неправильно определённых системных требований* - это риски, когда в самом начале проекта были некорректно сформулированы характеристики целевой системы, для которой разрабатывается программное обеспечение: программное окружение или требования к аппаратной части

*Риски использования нестабильных технологий* - это риски, связанные с использованием новых технологий, которые ещё не прошли апробацию в производстве или других проектах.

*Риски, связанные с неспособностью справиться со сложностью проекта* - иногда проект может быть настолько сложным, что команда попросту может с ним не справиться.

*Риск низкой продуктивности* обусловлен длительностью реализации проекта. Это в самом начале проекта создаёт большую потерю времени, которую сложно будет наверстать. При этом приходится либо переносить сроки, либо работать в более динамичном режиме на более поздних этапах проекта

*Риск смены сотрудников*, когда проект покидают ключевые сотрудники, которые максимально владеют информацией.

*Риски хищения исходного кода* возникают, когда разработчики, уходя из компании, забирают с собой разрабатываемый ими проект и немного модифицировав исходный код, могут продать его или использовать в других проектах, например, у конкурентов.

*Риски нарушение закона об авторском праве* могут возникнуть при использовании разработчиками без ведома проектного менеджера чужого исходного кода, алгоритма или библиотеки, которые защищены законом об авторском праве, но не приобретены или их использование не согласовано с автором.

**Задание 2. Определите какие риски могут быть при разработке вашего программного продукта**

**Задание 3. Заполните таблицу**

Риск	Способ устранения	Последствия
		связанные с не устранением риска



**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 22, 23**

**Тема** Выявление первичных и вторичных ошибок

**Цель:** научиться выявлять первичные и вторичные ошибки программного обеспечения

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретические сведения**

Теоретические сведения

Под ошибкой в широком смысле слова понимается неправильность, погрешность или неумышленное, невольное искажение объекта или процесса. При этом подразумевается, что известно правильное или неискаженное эталонное состояние объекта, к которому относится ошибка.

Считается, что если программа не выполняет того, что пользователь от нее ожидает, то в ней имеется ошибка.

Важной особенностью процесса выявления ошибок в сложных программах является отсутствие полностью определенной правильной программы-эталона, которому должен соответствовать проверяемый текст.

Поэтому нельзя гарантированно утверждать, что возможно написать программу без ошибок.

Искажение выходных результатов исполнения программ (вторичная ошибка) вызывает необходимость выполнения ряда операций по локализации устранению первичной ошибки (отладка программ).

На этапе отладки программ выявляются и исправляются много ошибок, но не все.

После отладки в течение некоторого времени интенсивность обнаружения ошибок при самом активном тестировании снижается настолько, что разработчики попадают в зону нечувствительности к программным ошибкам и отказам.

При такой интенсивности отказов программ трудно прогнозировать затраты времени, необходимые для обнаружения очередной ошибки.

Создается представление о полном отсутствии ошибок в программе, о невозможности и бесцельности их поиска. Поэтому усилия на отладку сокращаются, и интенсивность обнаружения ошибок еще больше снижается.

Этой предельно низкой интенсивности обнаружения отказов соответствует наработка на обнаруженную ошибку, при которой прекращается улучшение характеристик программного обеспечения на этапах его отладки и испытаний у заказчика.

Ошибку можно отнести к одному из ниже перечисленных классов:

- системные ошибки;
- ошибки в выборе алгоритма;
- алгоритмические ошибки;
- технологические ошибки;
- программные ошибки.

**Системные ошибки** в большом (сложном) программном обеспечении определяются, прежде всего неполной информацией о реальных процессах, происходящих в источниках и потребителях информации.

На начальных стадиях проектирования программного обеспечения не всегда удается точно сформулировать целевую задачу всей системы и требования к ней. В процессе проектирования целевая функция системы уточняется и выявляются отклонения от уточненных требований, которые могут квалифицироваться как системные ошибки.

Некачественное определение требований к программе приводит к созданию программы, которая будет правильно решать неверно сформулированную задачу. В таких случаях, как правило, требуется полное перепрограммирование.

Признаком того, что создаваемая для заказчика программа может оказаться не соответствующей его истинным потребностям, служит ощущение неясности задачи. Письменная регистрация требований к программе заставляет заказчика собраться с мыслями и дать достаточно точное определение требований. Всякие устные указания являются заведомо ненадежными и часто приводят к взаимному недопониманию.

При автономной и в начале комплексной отладки программного обеспечения доля найденных системных ошибок в нем невелика (примерно 10%), но она существенно возрастает (до 35—40%) на завершающих этапах комплексной отладки.

В процессе эксплуатации преобладающими являются системные ошибки (примерно 80% всех ошибок).

**Ошибки в выборе алгоритма.** Часто плохой выбор алгоритма становится очевидным лишь после его опробования. Поэтому все же следует уделять внимание и время выбору алгоритма, с тем, чтобы впоследствии не приходилось переделывать каждую программу. Во избежание выбора некорректных алгоритмов, необходимо хорошо ознакомиться с литературой по своей специальности.

К алгоритмическим ошибкам следует отнести, прежде всего, ошибки, обусловленные некорректной постановкой функциональных задач, когда в спецификациях не полностью оговорены все условия, необходимые для получения правильного результата.

Эти условия формируются и уточняются в значительной части в процессе тестирования и выявления ошибок в результатах функционирования программ.

Также следует отнести ошибки связей модулей и функциональных групп программ. Их можно квалифицировать как ошибки некорректной постановки задачи.

Алгоритмические ошибки проявляются в неполном учете диапазонов изменения переменных, в неправильной оценке точности используемых и получаемых величин, в неправильном учете связи между различными переменными, в неадекватном представлении формализованных условий решения задачи в спецификациях или схемах, подлежащих программированию и т.д.

Эти обстоятельства являются причиной того, что для исправления каждой алгоритмической ошибки приходится изменять иногда целые ветви программного обеспечения, т.е. пока еще существенно больше операторов, чем при исправлении программных ошибок.

Алгоритмические ошибки значительно труднее поддаются обнаружению методами формализованного автоматического контроля. Вот почему необходимо тщательным образом продумывать алгоритм прежде, чем транслировать его в программу.

Некоторые программисты проверяют алгоритм следующим образом. Через несколько дней после составления алгоритма они повторно обращаются к описанию задачи и составляют алгоритм заново. Затем сличают оба варианта. Такой шаг на первый взгляд может показаться пустой тратой времени, однако всякая ошибка на уровне алгоритма может в дальнейшем обернуться катастрофой и повлечь основательный пересмотр программы.

**Технологические ошибки**— это ошибки документации и фиксирования программ в памяти ЭВМ. Они составляют 5—10 % от общего числа ошибок, обнаруживаемых при отладке. Большинство технологических ошибок выявляются автоматически формализованными методами (например, транслятором).

**Программные ошибки.** Языки программирования - это искусственные языки, созданные человеком для описания алгоритмов. Все предложения таких языков строятся по строгим синтаксическим правилам, обеспечивающим однозначное их понимание, что позволяет поручать расшифровку алгоритма ЭВМ, построенного по правилам семантики.

**Синтаксис** - это набор правил построения из символов алфавита специальных конструкций, с помощью которых можно составлять различные алгоритмы (программы). Эти правила требуют их неукоснительного соблюдения. В противном случае будет нарушен основной принцип - четкая и строгая однозначность в понимании алгоритма.

**Семантика языка** — это система правил истолкования построений конструкций. Правила семантики конструкций обычно вполне естественны и понятны, но в некоторых случаях их надо специально оговаривать, комментировать.

Таким образом, программы, позволяющие однозначно производить процесс переработки данных, составляются с помощью соединения символов из алфавита в предложения в соответствии с синтаксическими правилами, определяющими язык, с учетом правил семантики.

Выделяют синтаксические и семантические ошибки.

Под **синтаксическими ошибками** понимается нарушение правил записи программ на данном языке программирования. Они выявляются самой машиной, точнее транслятором, вовремя перевода записи алгоритма на язык машины. Исправление их осуществляется просто — достаточно сравнить формат исправляемой конструкции с синтаксисом в справочнике и исправить его.

**Семантические (смысловые) ошибки** — это применение операторов, которые не дают нужного эффекта (например,  $a$ —вместо,  $a+v$ ), ошибка в структуре алгоритма, в логической взаимосвязи его частей, в применении алгоритма к тем данным, к которым он неприменим и т.д. Правила семантики не формализуемы. Поэтому поиск и устранение семантической ошибки и составляет основу отладки.

Каждая программная ошибка влечет за собой необходимость изменения команд существенно меньше, чем при алгоритмических и системных ошибках.

На этапах комплексной отладки программного обеспечения и эксплуатации удельный вес программных ошибок падает и составляет примерно 15 и 30 % соответственно от общего количества ошибок, выявляемых в единицу времени.

**Задание 2** Заполните таблицу

Категории тяжести ошибки в программном обеспечении

Номер категории ошибки	Наименование категории тяжести ошибки	Описание последствий проявления ошибки

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 24**

**Тема** Обнаружение вируса и устранение последствий его влияния

**Цель:** научиться с помощью антивирусного программного обеспечения обнаруживать и устранять последствия вирусного заражения

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретические сведения**

Теоретические сведения

Вредоносными программами уголовное законодательство Российской Федерации понимает программы, специально созданные для нарушения нормального функционирования компьютерных программ.

Под нормальным функционированием понимается выполнение определенных в документации на программу операций.

В Российской Федерации отношения производителей и распространителей вирусов с обществом регулируются статьей 273 Уголовного кодекса, гласящей следующее: «Создание программ для ЭВМ или внесение изменений в уже существующие программы, заведомо приводящих к несанкционированному уничтожению, блокированию, модификации или копированию информации, нарушению работы ЭВМ, систем ЭВМ или их сети, а равно использование либо распространение таких программ или машинных носителей с такими программами наказывается лишением свободы на срок до 3-х лет со штрафом от 200 до 500 минимальных размеров оплаты труда...». Аналогичные законы приняты и в других странах.

*Компьютерные вирусы* - это целая группа компьютерных программ, способных размножаться путем самокопирования и внедряться в код других приложений. Кроме этого они способны по различным каналам рассылать свои копии. Внедряясь в код программ вирусы, приводят к ошибкам и нарушению работы компьютера, а также к уничтожению файлов.

Несмотря на значительные усилия по созданию противовирусного программного обеспечения, полной гарантии безопасности и защиты от попадания вредоносных программ на стационарное или мобильное устройство не существует.

Поэтому, наряду с использованием самых современных антивирусов нужно научиться придерживаться некоторых установок:

- не устанавливать и не запускать на компьютере незнакомые приложения из источников, не имеющих серьезной репутации;
- не открывать подозрительные сайты, а при входе на популярные сервисы в интернете обращать внимание на написание доменного имени в адресной строке;

— регулярно обновлять программы, обеспечивающие безопасность работы системы.

**Задание 2. Открыть ранее установленное антивирусное программное обеспечение**

**Задание 3. Проверить ПК на наличие вирусного программного обеспечения**

**Задание 4. Устранить последствия заражения вирусом**

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 25, 26**

**Тема** Установка и настройка антивируса. Настройка обновлений с помощью зеркала

### **Цели:**

1. научиться устанавливать и настраивать антивирусное ПО;
2. научиться настраивать обновления с помощью функции «Зеркало».

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

### **Последовательность выполнения работы**

**Задание . Используя следующую схему настроить антивирус**

1. Скачать пробную 30-ти дневную версию антивируса можно с сайта разработчика <https://www.esetnod32.ru/>
2. Установить антивирус на свое рабочее место
3. На последнем этапе нажимаем «Установить пробную версию»
4. Откройте вкладку «Дополнительные настройки»
5. Изменить переключатель на «Включить обнаружение потенциально опасных приложений»
6. Изменить переключатель на «Защита файловой системы в режиме реально времени»

### **II. Включение и настройка «Зеркала»**

1. **В окне «Дополнительные настройки» (F5) выберите команду «Разное»> «Лицензии».**
2. Нажмите кнопку **«Добавить...»**, укажите путь к файлу \*.lic и нажмите кнопку **«Открыть»**. Это позволит выбрать лицензию и настроить функцию зеркала.
3. В разделе **«Обновление»** нажмите кнопку **«Настройка»** и щелкните вкладку **«Зеркало»**.
4. **Установите флажки «Создать зеркало обновления» и «Передавать файлы обновления с помощью внутреннего HTTP-сервера».**
5. Введите полный путь к папке (**«Папка для дублируемых файлов»**), где будут храниться файлы обновления.
6. Параметры **«Имя пользователя»** и **«Пароль»** служат для аутентификации клиентских рабочих станций, пытающихся получить доступ к папке зеркального отображения. В большинстве случаев заполнять эти поля не требуется.
7. Установите для параметра «Аутентификация» значение **«НЕТ»**.



8. Выберите компоненты для загрузки (компоненты всех языковых версий, которые будут использоваться в данной сети). Компоненты отображаются только в случае, если они доступны на серверах обновления ESET.

9. Если зеркало является частью ERA, этот параметр можно настроить в консоли ERAC

10. В меню «Сервис» «Параметры сервера»» вкладка «Дополнительно»» «Изменить дополнительные параметры»» «ESET Remote Administrator»» «ERA Server»> «Настройка» > «Зеркало».

11. Включите все языковые версии программы, присутствующие в сети.

12. Если необходимо настроить зеркало таким образом, чтобы для обновления клиентов использовался протокол HTTPS, выберите **ERAC > Служебные программы > Параметры сервера...** > вкладка **Дополнительно > Изменить дополнительные**  
параметры. > ESET Remote Administrator > ERA  
Server > Настройка > Зеркало > Протокол > HTTPS

#### **Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 27**

**Тема** Настройка политики безопасности

**Цель:** научиться настраивать параметры политики безопасности в ОС Windows

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание 1. Изучите теоретические сведения**

Теоретические сведения

Безопасность операционной системы основана на правилах, регулирующих разные аспекты ее работы. Вместе эти правила составляют единую политику безопасности.

Политика безопасности состоит из набора правил, объединенных в следующие группы:

- Политики (правила, эти термины тождественны) учетных записей.
- Локальные политики.
- Правила журнала регистрации.
- Правила групп с ограниченным членством.
- Правила системных служб.
- Правила реестра.
- Правила файловой системы.
- Политики открытого ключа.
- Политики ограниченного использования программ.
- Политики безопасности IP.

**Задание 2. Исполуя следующую схему настроить систему**

1. Дайте команду «Пуск — Выполнить» и введите в открывшейся строке команду gpedit.msc.
2. Запустить редактор также можно, введя «gpedit» в поиске главного меню Windows 8/10.
3. Перейдите в директорию \HKLM\SYSTEM\CurrentControlSet\Services\gpsvc (процесс gpsvc.exe, видимый в диспетчере задач при работе «Клиента групповой политики», запускается реестром Windows из этого места).
4. Правый щелчок мышью откроет меню папки «gpsvc» — выберите «Разрешения»
5. Выберите другого владельца, зайдя в дополнительные параметры безопасности
6. Отметьте все разрешения. Желательно удалить лишних пользователей из настройки неограниченного доступа к службе.

7. Вернитесь к папке gpсvc в данной директории реестра и измените ключ Start, введя значение 4 («Отключено»).
8. Закройте все окна, нажав ОК, перезапустите компьютер.

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **Практическое занятие № 28**

### **Тема Настройка браузера**

**Цель:** научиться настраивать браузер и устанавливать расширения

**Материально-техническое и комплексно-методическое**

**обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

#### **Последовательность выполнения работы**

**Задание . Используя следующую схему настроить браузер**

1. Выбираем меню «Настройки» ^ «Свойства браузера»:
  2. Вкладка «Общие» позволяет задать адрес домашней страницы, которая будет автоматически загружаться в окно браузера при его запуске, цвета гиперссылок по умолчанию, название шрифта по умолчанию.
  3. Здесь же определяется сколько дней будет храниться ссылка посещенных страниц в журнале.
  4. Кроме того, для ускорения просмотра. Все посещенные страницы помещаются в специальную папку, и с помощью кнопки «Параметры» можно задать разные способы обновления таких страниц.
  5. С помощью вкладки «Безопасность» можно создать списки надежных узлов и узлов с ограниченными функциями. Зона Интернет будет при этом включать все остальные узлы, не вошедшие в эти две папки. Для каждой из них с помощью кнопки «Другой» можно изменить параметры безопасности, установленные для них по умолчанию. Здесь можно запретить выполнение сценариев, отображение всплывающих окон, загрузку файлов и т.д.
  6. Вкладка «Конфиденциальность» дает возможность настроить работу с файлами cookie, с помощью которых информация о пользователе автоматически передается на сервер.
  7. Вкладка «Содержание» позволяет ограничить доступ к некоторой информации (насилие, ненормативная лексика и т.д.).
  8. Вкладка «Подключения» позволяет установить подключение к Интернету.
  9. На вкладке «Дополнительно» можно задать некоторые дополнительные параметры работы (отключить загрузку графических изображений, отменить подчеркивание ссылок, запретить отладку сценариев и т.д.).
- Вкладка Программы позволяет определить программы, которые будут по умолчанию использоваться службами Интернета (почтовые программы, HTML редакторы и т.п.).

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

10.

## **Практическое занятие № 29**

### **Тема Работа с реестром**

**Цель:** научиться работать с реестром в операционной системой Windows

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

#### **Последовательность выполнения работы**

**Задание. Используя следующую схему настроить реестр**

1. Щелкнуть по поисковой строке на панели задач или воспользоваться сочетанием клавиш Win + S.

2. Вписать в текстовое поле команду «regedit».

3. При появлении результатов поиска нажать на кнопку «выполнить команду».

4. Процесс внесения правок в содержимое реестра представляет собой поиск нужной строки в определенном каталоге и указание нового значения для неё. Поиск нужной строки проще всего осуществить через соответствующую опцию, которая вызывается сочетанием клавиш Ctrl + F или кнопкой F3.

5. Прежде чем вносить изменения в какой-либо параметр, следует поискать в интернете информацию о нём.

Почистить реестр на Windows 10 можно утилитой CCleaner.

Скачав и установив данный продукт, можно приступить к процедуре поиска неисправностей и их устранения.

Для этого нужно:

1. Запустите программу CCleaner.

2. Перейти на вкладку «Реестр».

3. Нажать на кнопку «Поиск проблем».

4. Дождаться завершения процедуры поиска.

5. Нажать на кнопку «Исправить».

6. Подтвердить создание резервной копии и указать файл для сохранения текущей версии значений реестра.

7. Щелкнуть по кнопке «Исправить отмеченные».

8. Закрыть окно.

В том случае, если после ручного редактирования значений реестра или автоматической чистки с применением стороннего программного обеспечения наблюдаются проблемы в работе операционной системы, нужно восстановить реестр Windows 10 до прежнего состояния. Делается это путём импортирования созданного на этапе редактирования файла.

Для того, чтобы записать значения из файла в реестр, потребуется:

1. С помощью Проводника Windows найти папку, хранящую \*.reg - файл с резервной копией.

2. Дважды кликнуть по файлу.

Подтвердить импорт значений.

**Критерии оценивания:**

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

3.

### **Практическое занятие № 30**

**Тема** Работа с программой восстановления файлов и очистки дисков

**Цель:** научиться работа с программой восстановления файлов

**Материально-техническое и комплексно-методическое обеспечение:** Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

**Время выполнения:** 90 минут

**Форма отчетности по занятию:** вывод о проделанной работе

**Последовательность выполнения работы**

**Задание. Используя следующую схему восстановите файлы**

Запустите программу восстановления файлов

В выпадающем меню выберите носитель, на котором следует восстановить файлы

Если вы помните название файла, то нажмите на чёрный треугольник в правой части кнопки Scan в появившемся меню и выберите Scan contents.

Запускаем сканирование удалённых файлов с помощью Scan files.

По окончании процесса поиска будет выведен его результат в главном окне программы

Для восстановления нужных файлов требуется поставить галочки напротив нужных или, если требуется восстановить все, то поставить галочку в самом верхнем чек-боксе.

Затем нужно нажать кнопку Recover и выбрать, куда сохранять информацию.

Не сохраняйте данные на тот же диск с которого Вы их восстанавливаете. Иначе восстановленные файлы будут записываться на место восстанавливаемых файлов, что приведёт к их необратимому повреждению.



## II. Очистка диска

1. Наберите Win + E, чтобы открыть Проводник. Найдите системный диск (по особой иконке).

2. Кликните правой кнопкой и выберите Свойства.

3. Нажмите «Очистка диска»

4. Система предложит свои варианты

5. **Выберите какие файлы следует удалить**

III. Сделайте вывод по проделанной работе

## Критерии оценивания:

**Оценка 5 «отлично»** работа выполнена полностью и правильно, сделаны правильные выводы

**Оценка 4 «хорошо»** работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

**Оценка 3 «удовлетворительно»** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

**Оценка 2 «неудовлетворительно»** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

## **2. ОБЩИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ**

1. Внимательно прочитайте задание, при необходимости повторите лекционный материал по конспектам и другим источникам, относящийся к теме практической/ лабораторной работы.

2. Ответьте на контрольные вопросы, если они предложены.

1. Подготовьте все необходимое для выполнения задания, рационально подготовьте рабочее место.

2. Продумайте ход выполнения работы.

3. Если ваша работа связана с использованием ИКТ, проверьте наличие и работоспособность программного обеспечения, необходимого для выполнения задания.

4. Если при выполнении практической работы применяется групповое или коллективное выполнение задания, старайтесь поддерживать в коллективе нормальный психологический климат, грамотно распределить роли и обязанности. Вместе проводите анализ организации и промежуточные результаты практической работы микрогруппы.

5. При выполнении практического задания соблюдайте правила техники безопасности и охраны труда.

6. В процессе выполнения практической работы обращайтесь за консультациями к преподавателю, чтобы вовремя скорректировать свою деятельность, проверить правильность выполнения задания.

7. По окончании выполнения практической работы составьте письменный или устный отчет в соответствии с теми методическими указаниями по оформлению отчета, которые вы получили от преподавателя или в методических указаниях.

8. Сдайте готовую работу преподавателю для проверки.

9. Участвуйте в обсуждении и оценке полученных результатов практической работы (общегрупповом или в микрогруппах).

Студент должен:

- строго выполнять весь объем домашней подготовки, указанный в описаниях соответствующих практических работ;

- знать, что выполнению каждой работы предшествует проверка готовности студента, которая производится преподавателем;

- знать, что после выполнения работы бригада, которая назначается преподавателем на весь период работы, должна представить отчет о проделанной работе с обсуждением полученных результатов и выводов.

Процедура выставления окончательной оценки студенту по работе и порядок выполнения пропущенных работ по уважительным и неуважительным причинам следующая:

Объем работы исчисляется количеством полно и правильно выполненных заданий вне зависимости от порядка следования. Если задание не предполагает сохранения результатов в файле, то оценка выполнения производится по принципу вопрос-ответ или устное задание - выполненное

действие. Процент выполнения рассчитывается для каждой работы исходя из общего числа заданий.

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

Для допуска студента к итоговой аттестации по МДК необходимо выполнение не менее 95% заданий.

### **3. КРИТЕРИИ ОЦЕНИВАНИЯ ВЫПОЛНЕННЫХ РАБОТ**

Оценка за практическую работу складывается из оценки за выполнение работы и оценки за защиту.

**Оценка «отлично»** ставится, если студент выполнил практическую работу в полном объеме с соблюдением необходимой последовательности действий; в отчете правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

**На защите студент при ответе на вопросы** правильно понимает сущность вопроса, дает точное определение и истолкование основных понятий; сопровождает ответ новыми примерами, умеет применить знания в новой ситуации; может установить связь между изучаемым и ранее изученным материалом, а также с материалом, усвоенным при изучении других дисциплин.

**Оценка «хорошо»** ставится, если студент выполнил требования к оценке "5", но допущены 2-3 недочета.

**На защите студент при ответе на вопросы** ответ студента удовлетворяет основным требованиям к ответу на оценку 5, но дан без применения знаний в новой ситуации, без использования связей с ранее изученным материалом и материалом, усвоенным при изучении других дисциплин; студент допустил одну ошибку или не более двух недочетов и может их исправить самостоятельно или с небольшой помощью преподавателя.

**Оценка «удовлетворительно»** ставится, если студент выполнил работу не полностью, но не менее 50% объема практической работы, что позволяет получить правильные результаты и выводы; в ходе проведения работы были допущены ошибки.

**На защите студент при ответе на вопросы** правильно понимает сущность вопроса, но в ответе имеются отдельные пробелы в усвоении вопросов курса, не препятствующие дальнейшему усвоению программного материала; допустил не более одной грубой ошибки и двух недочетов.

**Оценка «неудовлетворительно»** ставится, если студент выполнил работу не полностью или объем выполненной части работы не позволяет сделать правильных выводов;

**На защите студент при ответе на вопросы** не овладел основными знаниями и умениями в соответствии с требованиями программы и допустил больше ошибок и недочетов, чем необходимо для оценки 3 или не может ответить ни на один из поставленных вопросов.

## **4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКИХ РАБОТ**

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

1. Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2021. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/472502>

2. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для среднего профессионального образования / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2021. — 147 с. — (Профессиональное образование). — ISBN 978-5-534-09823-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/473307>

3. Сергеев, А. Г. Стандартизация и сертификация : учебник и практикум для среднего профессионального образования / А. Г. Сергеев, В. В. Терегеря. — Москва : Издательство Юрайт, 2021. — 323 с. — (Профессиональное образование). — ISBN 978-5-534-04315-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/469819>

4. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2021. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/473347>

5. Илюшечкин, В. М. Основы использования и проектирования баз данных : учебник для среднего профессионального образования / В. М. Илюшечкин. — испр. и доп. — Москва : Издательство Юрайт, 2021. — 213 с. — (Профессиональное образование). — ISBN 978-5-534-01283-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/471698>

### **Интернет-ресурсы**

1. Образовательный портал INTUIT.RU <http://www.intuit.ru>
2. METANIT.COM. Сайт о программировании <https://metanit.com>
3. Журнал «Успехи современной науки» - Режим доступа: <https://www.elibrary.ru/item.asp?id=27712534>

4. Журнал «Известия Высших учебных заведений. Поволжский регион. Технические науки – Режим доступа: <https://www.elibrary.ru/item.asp?id=44291082>

**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»