

	Государственное бюджетное профессиональное образовательное учреждение «Южно-Уральский государственный колледж»
	Учебно-методический отдел
	Учебно-методические материалы
	Методические указания

**Методические указания по выполнению практических работ
 обучающихся по учебной дисциплине
 МДК 04.02 Обеспечение качества функционирования компьютерных
 систем**

Специальность 09.02.07 Информационные системы и программирование

Челябинск, 2019

	Должность	Фамилия/Подпись	Дата
Разработал	Преподаватель	Чераева О.А.	
Проверил	Зам. директора по учебно-методической работе	Манапова О.Н.	
Согласовал	Зам. директора по учебной работе	Калиновская Т.С.	
Версия: 01	Без подписи документ действителен 3 суток после распечатки. Дата и время распечатки:	Экземпляр № _____	с. 1 из 37

	ГБПОУ «ЮУГК»
	Учебно-методический отдел
	Учебно-методические материалы
	Методические указания

ББК 74.57

Чераева О.А. Методические указания по выполнению практических работ обучающихся по учебной дисциплине МДК 04.02 Обеспечение качества функционирования компьютерных систем: методические указания для студентов. - Издательский центр ЮУГК, 2019. – 37 с.

Рассмотрено и одобрено на заседании ПЦК *Информационных технологий*
 Протокол от 19 февраля 2019 г. №6
 Председатель ПЦК _____ Н.А. Назарова

Данные методические рекомендации способствуют эффективному выполнению практических работ обучающимися. Содержат требования к знаниям, умениям, общие компетенции, на формирование которых направлено выполнение самостоятельных работ; перечень видов и описание практических работ; формы контроля и критерии оценки выполнения практических работ.

Версия: 01	Без подписи документ действителен 3 суток после распечатки. Дата и время распечатки: _____	Экземпляр № 01	с. 2 из 37
------------	---	----------------	------------

Содержание

Пояснительная записка	4
Практическая работа № 1	7
Практическая работа № 2	9
Практическая работа № 3	11
Практическая работа № 4	14
Практическая работа № 5	20
Практическая работа № 6	23
Практическая работа № 7	25
Практическая работа № 8	27
Практическая работа № 9	29
Практическая работа № 10	32
ПРИЛОЖЕНИЕ 1	35
ПРИЛОЖЕНИЕ 2	36

Пояснительная записка

Методические указания по выполнению практических работ студентов по дисциплине МДК 04.02 Обеспечение качества функционирования компьютерных систем предназначены для студентов по специальности 09.02.07 Информационные системы и программирование.

Цель методических указаний: оказание помощи студентам в выполнении практической работы по дисциплине МДК 04.02 Обеспечение качества функционирования компьютерных систем.

Настоящие методические указания содержат работы, которые позволят студентам овладеть фундаментальными знаниями, профессиональными умениями и навыками деятельности по специальности, опытом творческой и исследовательской деятельности и направлены на формирование следующих компетенций:

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

ПК 4.4. Обеспечивать защиту программного обеспечения компьютерных систем программными средствами

ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности

ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста

ОК 9. Использовать информационные технологии в профессиональной деятельности

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках

В результате освоения учебной дисциплины МДК 04.02 Обеспечение качества функционирования компьютерных систем обучающийся должен уметь:

- производить настройку отдельных компонентов программного обеспечения компьютерных систем;
- использовать методы защиты программного обеспечения компьютерных систем.

В результате освоения учебной дисциплины МДК 04.02 Обеспечение качества функционирования компьютерных систем обучающийся должен знать:

- 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.

Описание каждой практической работы содержит: тему, цели работы, задания, формы контроля, требования к выполнению и оформлению заданий.

Для получения дополнительной, более подробной информации по изучаемым вопросам, приведено учебно-методическое и информационное обеспечение.

Перечень практических работ представлен в таблице 1.

Таблица 1

№ и название темы	Кол-во часов	Вид практической работы	Форма контроля
Тема 1. Основные методы обеспечения качества функционирования	10		
1. Тестирование программных продуктов	2	Аудиторная	Представление выполненных работ
2. Сравнение результатов тестирования с требованиями технического задания и/или спецификацией	2	Аудиторная	Представление выполненных работ
3. Анализ рисков	2	Аудиторная	Представление выполненных работ
4. Выявление первичных и вторичных ошибок	4	Аудиторная	Представление выполненных работ
Тема 2. Методы и средства защиты компьютерных систем	14		
5. Установка и настройка антивируса. Настройка обновлений с помощью зеркала	2	Аудиторная	Представление выполненных работ
6. Обнаружение вируса и устранение последствий его влияния	2	Аудиторная	Представление выполненных работ
7. Настройка политики безопасности	4	Аудиторная	Представление выполненных работ
8. Настройка браузера	2	Аудиторная	Представление выполненных работ
9. Работа с реестром	2	Аудиторная	Представление выполненных работ
10. Работа с программой восстановления файлов и очистки дисков	2	Аудиторная	Представление выполненных работ
Итого	24		

Практическая работа № 1

Тема Тестирование программных продуктов

Цели: ознакомление с методами тестирования программного продукта

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 сандвича – 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.

Методические указания

1. Выбрать методы тестирования программного продукта.
2. Разработать план тестирования программного продукта.
3. Провести тестирование программы и представить результаты в виде таблицы 2

Таблица 2

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.

Контрольные вопросы

1. Что такое тестирование программного обеспечения?
2. Чем тестирование отличается от отладки?
3. Для чего проводится функциональное тестирование?
4. Что такое комплексное тестирование?
5. Каковы правила тестирования программы «как черного ящика»?
6. Как проводится тестирования программы по принципу «белого ящика»?
7. Что такое модульное тестирование?
8. Как осуществляется сборка программы при модульно тестировании?

Практическая работа № 2

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.

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

Методические указания

1. Составьте техническое задание, которое будет содержать необходимые требования к программному продукту
2. Составьте спецификацию на разработку программного продукта
3. Сравните результаты тестирования из предыдущей работы с требованиями технического задания
4. Сравните результаты тестирования из предыдущей работы с требованиями спецификации

Контрольные вопросы

1. Подтверждает ли тестирование правильность программы?
2. Что можно сказать о программе, если она на значительном количестве тестов ведет себя правильно?
3. Может ли повысить надежность программы процесс тестирования?
4. Типы ошибок, обнаруживаемые при тестировании.
5. Основные принципы тестирования.

Практическая работа № 3

Тема Анализ рисков

Цель: научиться производить анализ рисков программного обеспечения

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.

Методические указания

1. Определите какие риски могут быть при разработке вашего программного продукта
2. Заполните таблицу 3

Таблица 3

Риски разработки программного обеспечения

Риск	Способ устранения	Последствия связанные с не устранением риска

3. Каким образом можно минимизировать риски.

Контрольные вопросы

1. Как влияет на разработку программного продукта текучка кадров и низкая производительность кадров?
2. Перечислите основные риски при разработке программного обеспечения.
3. Перечислите общие методы оценки рисков.

Практическая работа № 4

Тема Выявление первичных и вторичных ошибок

Цель: научиться выявлять первичные и вторичные ошибки программного обеспечения

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.



Рисунок 1 – Распределение выявленных ошибок

Искажения в тексте программ (первичные ошибки) являются элементами, подлежащими корректировке. Однако непосредственно наличие ошибки обнаруживаются по ее вторичным проявлениям.

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 удастся 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 $a + b$), 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.

Методические указания

1. Заполните таблицу 4

Таблица 4

Категории тяжести ошибки в программном обеспечении

Номер категории ошибки	Наименование категории тяжести ошибки	Описание последствий проявления ошибки

2. Разделите ошибки на первичные и вторичные
3. Сделайте вывод по проделанной работе

Контрольные вопросы

1. Приведите классификацию ошибок программного обеспечения
2. Какие ошибки могут возникнуть при тестировании программного обеспечения?
3. Перечислите основные пути борьбы с ошибками.

Практическая работа № 5

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».

Теоретические сведения

Антивирусом называют специализированную программу, в функции которой входит обнаружение и удаление вредоносных приложений на компьютере. Кроме этого антивирусы выполняют функцию восстановления зараженных файлов.

Большинство компаний, разрабатывающих антивирусное программное обеспечение, выпускают свою продукцию под операционные системы Windows.

Связано это с тем, что большая часть вирусных программ разрабатывается именно под эту платформу, просто потому что Windows, это наиболее популярная и наиболее распространенная система.

Методические указания

I. Установка антивирусного программного обеспечения

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**

Контрольные вопросы

1. Для чего используется функция «Зеркала» в антивирусном программном обеспечении?
2. Перечислите типы обновлений антивирусного программного обеспечения и их характеристики.
3. Опишите принцип работы сервера зеркала.

Практическая работа № 6

Тема Обнаружение вируса и устранение последствий его влияния

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

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 рабов 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.

Методические указания

1. Открыть ранее установленное антивирусное программное обеспечение
2. Проверить ПК на наличие вирусного программного обеспечения
3. Устранить последствия заражения вирусом

Контрольные вопросы

1. Что называют «вредоносным программным обеспечением»?
2. Какое наказание предусмотрено в УК РФ за распространение вредоносного программного обеспечения?
3. Перечислите законы аналогичные статье 273 УК РФ, действующие за пределами РФ.

Практическая работа № 7

Тема Настройка политики безопасности

Цель: научиться настраивать параметры политики безопасности в ОС Windows

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.

Методические указания

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 gpvc в данной директории реестра и измените ключ Start, введя значение 4 («Отключено»).

8. Закройте все окна, нажав ОК, перезапустите компьютер.

Контрольные вопросы

1. На какие группы делятся локальные политики?
2. Что нужно сделать чтобы управлять конфигурациями безопасностями для нескольких устройств?
3. От каких технологий зависят политики настройки безопасности?

Практическая работа № 8

Тема Настройка браузера

Цель: научиться настраивать браузер и устанавливать расширения

Теоретические сведения

Браузер - прикладное программное обеспечение для просмотра страниц, содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями.

В глобальной сети браузеры используют для запроса, обработки, манипулирования и отображения содержания веб-сайтов. Многие современные браузеры также могут использоваться для обмена файлами с серверами FTP, а также для непосредственного просмотра содержания файлов многих графических форматов (GIF, JPEG, PNG, SVG), аудио-видео форматов (MP3, MPEG), текстовых форматов (PDF, DJVU) и других файлов.

Функциональные возможности браузеров постоянно расширяются и улучшаются благодаря конкуренции между их разработчиками и высоким темпом развития и внедрения информационных технологий. Браузеры распространяются, как правило, бесплатно. Потребителям браузер может быть поставлен в форме самостоятельного (автономного) приложения или в составе комплектного программного обеспечения.

В настоящее время наиболее популярными являются: Internet Explorer; Mozilla Firefox; Opera; Google Chrome; Safari.

Методические указания

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.).

10. Вкладка Программы позволяет определить программы, которые будут по умолчанию использоваться службами Интернета (почтовые программы, HTML редакторы и т.п.).

Контрольные вопросы

1. Что такое «Браузером»?
2. Перечислите современные браузеры?
3. Как настроить безопасную работу браузера?

Практическая работа № 9

Тема Работа с реестром

Цель: научиться работать с реестром в операционной системой Windows

Теоретические сведения

Реестр – особая часть операционной системы Windows, которая представляет собой базу данных из всех параметров или настроек операционной системы.

Все опции Панели управления и других мест системы, где есть возможность изменения каких-либо параметров, зафиксированы в реестре.

Там же хранятся данные о путях к файлам, о расположении установленных программ и других моментах, связанных с функционированием Windows.

Физически реестр состоит из группы файлов, которые хранятся в папке System32\config. При загрузке операционной системы из этих файлов собирается база данных с текущими параметрами, которые и применяются в работе Windows. Эта база состоит из пяти главных веток.

Редактировать файлы из указанного каталога напрямую не рекомендуется, поскольку для того, чтобы войти в реестр Windows 10, существует специальный инструмент, зовущийся regedit.

Реестр Windows – мощный инструмент для управления операционной системы, предоставляющий доступ ко многим параметрам, не реализованным в стандартном визуальном интерфейсе Windows.

Для доступа к таким параметрам часто применяют программы-твикеры, которые могут настроить размер оконных рамок проводника, время проигрывания анимаций и много других параметров.

В любом случае, изменять параметры реестра не рекомендуется без особой причины, а перед исправлением ошибочных значений всегда следует создавать резервную копию.

Методические указания

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.
3. Подтвердить импорт значений.

Контрольные вопросы

1. Что такое реестр в ОС Windows?
2. Из чего состоит структура реестра?
3. Где хранятся файлы реестра?

Практическая работа № 10

Тема Работа с программой восстановления файлов и очистки дисков

Цель: научиться работа с программой восстановления файлов

Теоретические сведения

Операционная система «Windows» отслеживает месторасположение файлов на жестком диске через определенные метки. Каждый файл и папка, расположенные на диске, содержат собственный указатель, который уведомляет операционную систему, где начинаются и заканчиваются данные каждого конкретного файла.

Когда пользователь запускает процесс удаления файла, тем или иным способом, система «Windows» удаляет лишь конкретный указатель и отмечает, что сектора, содержащие данные удаленного файла, теперь свободны, и доступны для записи новых файлов. С точки зрения файловой системы, удаленный файл больше не присутствует на вашем жестком диске, а сектора, содержащие его данные, считаются незанятыми.

Однако до тех пор, пока операционная система «Windows» фактически не будет записывать новые данные в сектора, содержащие содержимое файла, который был удален пользователем, такой файл по-прежнему годен к последующему восстановлению. Программа восстановления файлов может выполнить сканирование жесткого диска на предмет наличия удаленных файлов и способна восстановить их.

Если файл был частично перезаписан, программа восстановления файлов может восстановить только часть данных. Такое происходит, когда часть секторов, хранящих содержимое удаленного файла, была использована для записи новых данных.

В некоторых случаях, программа восстановления может восстановить файл полностью, исправив перезаписанные сектора, используя глубокие методы анализа диска.

Описанный процесс удаления в полной мере относится к файлам, хранящимся на жестких магнитных дисках «HDD». Для файлов, удаленных с

твердотельных накопителей «SSD», данный процесс не подходит, и в следующем разделе мы остановимся на описании особенностей удаления файлов с накопителей «SSD» более подробно.

Использование автоматических программ для удаления неиспользуемых файлов позволяет освобождать дисковое пространство и ускорять общую производительность системы, но повышает риск случайного удаления важных данных.

Методические указания

I. Восстановление файлов

1. Запустите программу восстановления файлов
2. В выпадающем меню выберите носитель, на котором следует восстановить файлы
3. Если вы помните название файла, то нажмите на чёрный треугольник в правой части кнопки Scan в появившемся меню и выберите Scan contents.
4. Запускаем сканирование удалённых файлов с помощью Scan files.
5. По окончании процесса поиска будет выведен его результат в главном окне программы
6. Для восстановления нужных файлов требуется поставить галочки напротив нужных или, если требуется восстановить все, то поставить галочку в самом верхнем чек-боксе.
7. Затем нужно нажать кнопку Recover и выбрать, куда сохранять информацию.
8. 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

Контрольные вопросы

1. Что происходит при удалении файлов?
2. Почему удаленные файлы не стираются сразу?
3. Как можно восстановить удаленный файл?

ПРИЛОЖЕНИЕ 1

Общие требования к выполнению и оформлению практических работ

Ход работы:

- изучить теоретический материал;
- выполнить задания;
- описать ход выполнения заданий;
- ответить на контрольные вопросы.

Выполнение практических работ должно быть оформлено в MS WORD.

Отчёт по практической работе должен содержать:

- номер практической работы;
- тему и цель практической работы;
- схемы и структуры;
- заполненные таблицы;
- ответы на контрольные вопросы;
- выводы по работе.

ПРИЛОЖЕНИЕ 2

Критерии оценки работы студента на практическом занятии

1. Критерии оценки выполнения практических заданий.

Оценка «отлично» ставится:

- если студент выполнил работу в полном объеме с соблюдением необходимой последовательности действий;
- в отчете правильно выполнена последовательность заданий;
- правильно выполняет анализ ошибок.

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

Оценка «удовлетворительно» ставится,

- если студент выполнил работу не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы;
- в ходе проведения работы были допущены ошибки.

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

2. Оценивание защиты контрольных вопросов.

Оценка «отлично» ставится в том случае, если студент

- правильно понимает сущность вопроса, дает точное определение и истолкование основных понятий;
- строит ответ по собственному плану, сопровождает ответ новыми примерами, умеет применить знания в новой ситуации;
- может установить связь между изучаемым и ранее изученным материалом курса, а также с материалом, усвоенным при изучении других дисциплин.

Оценка «хорошо» ставится, если

- ответ студента удовлетворяет основным требованиям к ответу на оценку 5, но дан без использования собственного плана, новых примеров, 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

— 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

— не овладел основными знаниями и умениями в соответствии с требованиями программы и допустил больше ошибок и недочетов, чем необходимо для оценки 3.

— не может ответить ни на один из поставленных вопросов.