

**Автономная некоммерческая организация профессионального образования  
«ПЕРМСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»  
(АНО ПО «ПГТК»)**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ  
ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ  
МДК 02.03. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ  
ПРОГРАММНЫХ МОДУЛЕЙ**

для специальности

**09.02.11 Разработка и управление программным обеспечением  
(код и наименование специальности)**

Квалификация выпускника  
**Программист**

Форма обучения  
**Очная**

Пермь 2026

Методические рекомендации по выполнению практических работ МДК 02.03. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ составлен в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением (утвержден приказом Министерства Просвещения Российской Федерации от 24 февраля 2025 г. N 138).

Данные методические рекомендации помогут организовать самостоятельную деятельность студентов на основе деятельного и компетентного подходов к обучению, что соответствует ФГОС СПО по специальности 09.02.11 Разработка и управление программным обеспечением.

Автор – составитель: Могильникова Н.С., старший преподаватель.

Методические рекомендации по выполнению практических работ рассмотрена и одобрена на заседании кафедры математических и естественно-научных дисциплин, протокол, № 01 от 04.02.2026.

Методические рекомендации по выполнению практических работ предназначен для оценивания достижений запланированных результатов по дисциплине МДК 02.03. ПОДДЕРЖКА И ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ. Методические рекомендации по выполнению практических работ представляет собой комплект материалов для проведения практических занятий (в форме практической подготовке) и осуществления контроля за выполнением работ.

Методические рекомендации по выполнению практических работ позволяет оценивать:

Код ОК, ПК	Уметь	Знать	Владеть навыками
ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;	распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части определять этапы решения задачи, составлять план действия, реализовывать составленный план, определять необходимые ресурсы выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы владеть актуальными методами работы в профессиональной и смежных сферах оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)	актуальный профессиональный и социальный контекст, в котором приходится работать и жить структура плана для решения задач, алгоритмы выполнения работ в профессиональной и смежных областях основные источники информации и ресурсы для решения задач и/или проблем в профессиональном и/или социальном контексте методы работы в профессиональной и смежных сферах порядок оценки результатов решения задач профессиональной деятельности	
ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;	определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации выделять наиболее значимое в перечне информации, структурировать получаемую информацию, оформлять результаты поиска оценивать практическую значимость результатов поиска	номенклатура информационных источников, применяемых в профессиональной деятельности приемы структурирования информации формат оформления результатов поиска информации современные средства и устройства информатизации, порядок их применения программное обеспечение в профессиональной деятельности, в том числе цифровые средства	

	применять средства информационных технологий для решения профессиональных задач использовать современное программное обеспечение в профессиональной деятельности использовать различные цифровые средства для решения профессиональных задач	психологические основы деятельности коллектива	
ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;	грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке проявлять толерантность в рабочем коллективе	правила построения устных сообщений особенности социального и культурного контекста	
ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.	понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы участвовать в диалогах на знакомые общие и профессиональные темы строить простые высказывания о себе и о своей профессиональной деятельности кратко обосновывать и объяснять свои действия (текущие и планируемые) писать простые связные сообщения на знакомые или интересующие профессиональные темы	правила построения простых и сложных предложений на профессиональные темы основные общеупотребительные глаголы (бытовая и профессиональная лексика) лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности особенности произношения правила чтения текстов профессиональной направленности	
ПК. 2.1 Проектировать модули программного обеспечения.	проектировать модули, соответствующие бизнес-задачам. создавать архитектурные	основные принципы проектирования модулей программного обеспечения 3	проектирования модулей ПО с учетом требований заказчика.

	<p>диаграммы и документацию.</p> <p>определять структуру и интерфейсы модулей</p> <p>анализировать требования к модулю и определять его функциональность</p> <p>проектировать архитектуру модуля, включая выбор подходящих паттернов проектирования и структуры данных</p> <p>создавать диаграммы классов, последовательностей и прочих диаграмм для визуализации проектируемого модуля</p> <p>выбирать подходящие языки программирования и технологии для реализации модуля</p> <p>проектировать интерфейсы программного обеспечения для взаимодействия с другими модулями и системами</p> <p>учитывать требования к масштабируемости, производительности и безопасности при проектировании модуля</p> <p>проводить анализ и оптимизацию проектируемого модуля для повышения его эффективности и качества</p>	<p>языки программирования и технологии для реализации модулей</p> <p>паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей</p> <p>методы анализа требований и способов определения функциональности модуля</p> <p>принципы создания интерфейсов для взаимодействия с другими модулями и системами</p> <p>принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей</p> <p>методы анализа и оптимизации проектируемых модулей для повышения их эффективности и качества</p>	<p>создания архитектурных диаграмм и спецификаций модулей.</p> <p>определения интерфейсов и взаимодействия модулей в системе.</p>
<p>ПК. 2.2</p> <p>Разрабатывать модули программного обеспечения.</p>	<p>разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий</p> <p>применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей</p> <p>анализировать требования и</p>	<p>язык программирования, основные конструкции, синтаксис</p> <p>паттерны проектирования структуры данных</p> <p>принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP</p> <p>работа с инструментальным программным обеспечением</p>	<p>создание модулей программного обеспечения на различных языках программирования</p> <p>отладки и тестирования разработанных модулей</p> <p>применение структурного и объектно-ориентированного программирования</p>

	<p>определять функциональность модуля</p> <p>создавать интерфейсы для взаимодействия с другими модулями и системами</p> <p>обеспечивать безопасность,</p> <p>производительность и масштабируемость при разработке модулей</p> <p>оптимизировать проектируемые модули для повышения их эффективности и качества</p> <p>работать с системой контроля версий</p> <p>улучшать производительность модулей, выявляя и устраняя узкие места</p> <p>проводить анализ и мониторинг производительности приложений</p> <p>применять инструменты для рефакторинга и оптимизации программного кода</p>	<p>методы оптимизации кода и алгоритмов</p> <p>эффективные алгоритмы и структуры данных для повышения производительности</p> <p>многопоточность в программных модулях</p> <p>методы оптимизации сетевых протоколов для ускорения обмена данными</p> <p>кэширование данных</p> <p>управление памятью</p> <p>техники повышения производительности программного обеспечения</p>	<p>оптимизации кода и алгоритмов</p> <p>программных модулей для увеличения производительности мониторинга и анализа</p> <p>производительности приложений</p>
<p>ПК. 2.3</p> <p>Выполнять интеграцию модулей и компонентов программного обеспечения.</p>	<p>интегрировать модули и компоненты,</p> <p>обеспечивая их взаимодействие</p> <p>работать с API и устанавливать соединения между компонентами</p> <p>отслеживать и устранять конфликты и ошибки интеграции</p> <p>анализировать и определять зависимости между модулями и компонентами</p> <p>работать с различными форматами данных и протоколами передачи данных</p>	<p>общих принципов функционирования аппаратных, программных и программно-аппаратных средств администрируемой информационно-коммуникационной системы</p> <p>международных стандартов локальных вычислительных сетей</p> <p>методы и подходы к интеграции модулей и компонентов</p> <p>принципы версионирования и управления изменениями при интеграции</p> <p>принципы безопасности при интеграции модулей и компонентов</p>	<p>интеграции программных модулей и компонентов в единое программное решение</p> <p>работы с API и веб-сервисами для взаимодействия между модулями</p> <p>работы с интеграционными платформами и инструментами обеспечения совместимости и стабильности системы</p>
<p>ПК. 2.4</p> <p>Выполнять тестирование и отладку программного обеспечения.</p>	<p>анализировать требования к программному обеспечению и составлять планы тестирования.</p>	<p>принципы и методы тестирования программного обеспечения.</p> <p>основы программирования и архитектуры программного обеспечения.</p>	<p>отладки программного обеспечения на уровне программных модулей</p>

	<p>создавать тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям.</p> <p>выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования.</p> <p>анализировать результаты тестирования и документировать найденные ошибки.</p> <p>разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении.</p> <p>выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования</p> <p>использовать системы контроля дефектов ПО</p> <p>составлять отчет о выполнении тестирования ПО</p>	<p>основы баз данных и SQL-запросов.</p> <p>инструменты для автоматизации тестирования</p> <p>основы разработки и отладки программного обеспечения на разных языках программирования</p> <p>понятие дефекта программного обеспечения</p> <p>критерии качества ПО</p> <p>виды и типы тестирования ПО</p> <p>техники ручного тестирования</p> <p>техники автоматизированного тестирования</p> <p>жизненный цикл дефекта ПО</p> <p>принципы работы в системе контроля дефектов</p> <p>основные понятия о качестве ПО</p>	<p>тестирования программного обеспечения</p> <p>формирования тестовых сценариев</p> <p>подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости)</p> <p>оценки объема тестирования ПО с целью определения необходимых ресурсов для его выполнения</p> <p>настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции</p> <p>формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами</p> <p>выполнения тестовых процедур на тестовых данных</p>
<p>ПК. 2.5</p> <p>Осуществлять документирование программных модулей программного обеспечения.</p>	<p>описывать функциональность модулей в документации</p> <p>создавать диаграммы для иллюстрации работы модулей</p> <p>программировать с использованием комментариев для документирования кода</p> <p>использовать специальные метки/теги для отметки важных частей кода в документации</p> <p>вести журнал изменений и</p>	<p>стандарты технической документации</p> <p>принципы документирования программного обеспечения</p> <p>инструменты для создания технической документации и комментирования кода</p>	<p>создания технической документации для модулей документирования кода, API и интерфейсов работы со специализированным ПО по документированию программного кода</p>

	фиксировать обновления программных модулей разбивать модули на логические блоки и описывать каждый блок отдельно включать в документацию особенности модулей, такие как ограничения, уязвимости или оптимальные настройки проводить регулярное обновление документации при изменении модулей или добавлении нового функционала.		
--	---	--	--

В результате текущей аттестации по МДК 02.03. Поддержка и тестирование программных модулей осуществляется проверка сформированности умений и знаний, направленных на формирование соответствующих ФГОС СПО общих и профессиональных компетенций.

Перечень практических работ.

**Перечень практических занятий.**

### **Практическое занятие «Разработка тестового сценария проекта»**

Цель: Научиться разрабатывать простейшие тестовые сценарии (test case)

Задание: Написать тестовый сценарий из не менее 10 шагов, соответствующий полученному варианту задания. Сценарий должен включать в себя не только основной вариант использования функционала, но и ошибочный (например: ввод пустого/неверного пароля в примере). Обратите внимание, что все предварительные действия, необходимые для прохождения шага, должны быть явно описаны. Например, нельзя требовать от тестирующего банкомата ввести ПИН код до того, как он вставил карту.

**ВАРИАНТЫ:**

Оплата мобильного телефона, [через платежный терминал](#)

1. Снятие наличных денег в банкомате
2. Проезд в автобусе с кондуктором
3. Использование [будильника мобильного телефона](#)
4. Ксерокопирование



5. Проход в метро (по смарт-карте и/или с жетоном)
6. Закрывание двери ключом
7. Поездка в лифте
8. Звонок в службу поддержки Интернет-провайдера/мобильного оператора

#### СОДЕРЖАНИЕ ОТЧЕТА:

1. Титульный лист с название группы, номером и темой практического задания, вариантом, ФИО.
2. Цель и задание, соответствующие полученному варианту.
3. Результаты работы: [тестовый сценарий в виде таблицы](#), включающий в себя номер шага, описание действия, необходимые на данном шаге тестовые данные и ожидаемый результат выполнения шага.
4. Выводы: достигли ли цель работы.

#### ПРИМЕР:

Название	IS-login-1
Дата создания	29.10.2021
Автор	Ivan Ivanov
Дата последнего изменения	30.10.2021
Описание	Проверка функционирования подсистемы «Вход в IS» некой информационной системы на соответствие требованиям при вводе корректных и некорректных значений.

Шаг №	Описание	Тестовые данные	Ожидаемый результат
1	Введите имя пользователя. Нажмите кнопку «Войти»	Имя пользователя = Test	Основное окно программы не

			открывается. Должно быть выведено сообщение «Введите пароль»
2	Введите пароль. Нажмите кнопку «Войти»	Пароль = Test	Основное окно программы не открывается. Должно быть выведено сообщение «Введите имя пользователя»
3	Введите имя пользователя и пароль. Нажмите кнопку «Войти»	Имя пользователя = Test  Пароль = XXX	Основное окно программы не открывается. Должно быть выведено сообщение «Введите имя пользователя и/или пароль неверные. Пожалуйста введите правильные данные»
4	Введите имя пользователя и пароль. Нажмите кнопку «Войти»	Имя пользователя = XXX  Пароль = Test	Основное окно программы не открывается. Должно быть выведено сообщение «Введите имя пользователя и/или пароль неверные. Пожалуйста введите правильные данные»
5	Введите имя пользователя и пароль. Нажмите кнопку «Войти»	Имя пользователя = XXX  Пароль = XXX	Основное окно программы не открывается. Должно быть выведено сообщение «Введите имя пользователя и/или пароль неверные. Пожалуйста введите правильные данные»
6	Введите имя пользователя и пароль. Нажмите кнопку «Войти»	Имя пользователя = « »  Пароль « »	Основное окно программы не открывается. Должно быть

			выведено сообщение «Введите имя пользователя и/или пароль неверные. Пожалуйста введите правильные данные»
7	Введите имя пользователя и пароль. Нажмите кнопку «Войти»	Имя пользователя = Test  Пароль = Test	Должно открыться основное окно приложения.
8	Введите имя пользователя и пароль. Нажмите кнопку «Войти»	USER = ADMIN  Пароль = ADMIN	Должно открыться окно приложения с ад

## Практическое занятие «Разработка тестовых пакетов»

**Цель:** получить навыки разработки тестовых пакетов.

### Теоретические сведения:

В ходе выполнения лабораторной работы провести тестирование по принципу «белого ящика».

Критерии покрытия кода:

- покрытие операторов — каждая ли строка исходного кода была выполнена и протестирована;
- покрытие условий — каждая ли точка решения (вычисления истинно ли или ложно выражение) была выполнена и протестирована;
- покрытие путей — все ли возможные пути через заданную часть кода были выполнены и протестированы;
- покрытие функций — каждая ли функция программы была выполнена;
- покрытие вход/выход — все ли вызовы функций и возвраты из них были выполнены;
- покрытие значений параметров — все ли типовые и граничные значения параметров были проверены.

### Метод покрытия операторов

Целью этого метода тестирования является выполнение каждого оператора программы хотя бы один раз.

Пример.

Если для тестирования задать значения переменных  $A = 2$ ,  $B = 0$ ,  $X = 3$ , будет реализован путь *ace*, т. е. каждый оператор программы выполнится один раз (рис. Л5.1, *a*). Но если внести в алгоритм ошибки — заменить в первом условии *and* на *or*, а во втором  $X > 1$  на  $X < 1$  (рис. Л5.1, *b*), ни одна ошибка не будет обнаружена (табл. Л5.1). Кроме того, путь *abd* вообще не будет охвачен тестом, и если в нем есть ошибка, она также не будет обнаружена. В табл. Л5.1 ожидаемый результат определяется по блок-схеме на рис. Л5.1, *a*, а фактический — по рис. Л5.1, *b*.

Как видно из этой таблицы, ни одна из внесенных в алгоритм ошибок не будет обнаружена.

Таблица Л5.1. Результат тестирования методом покрытия операторов

Тест	Ожидаемый результат	Фактический результат	Результат тестирования
$A = 2, B = 0, X = 3$	$X = 2,5$	$X = 2,5$	Неуспешно

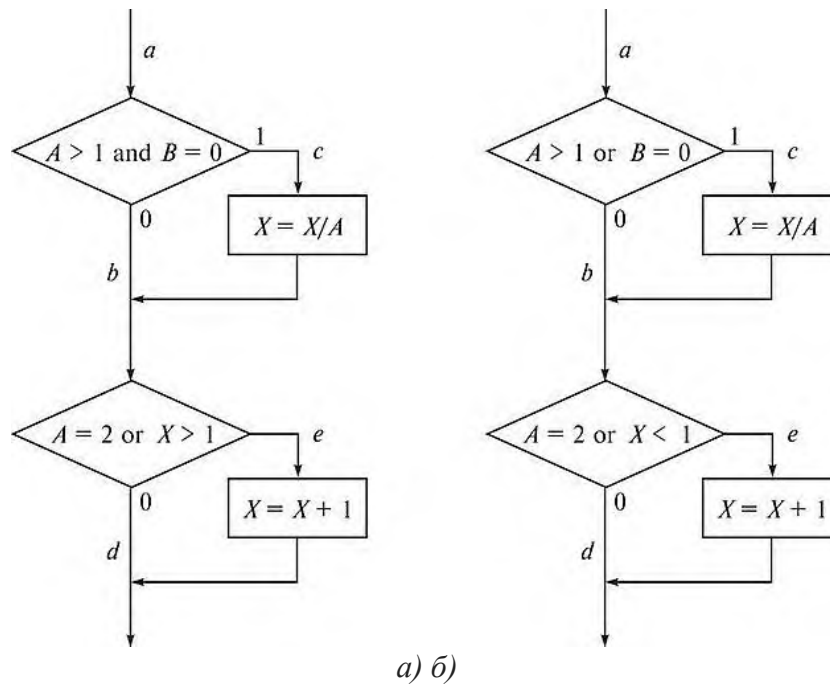


Рис. Л5.1. Пример алгоритма программы: *a* — правильный; *б* — с ошибкой

#### Метод покрытия решений (покрытия переходов)

Согласно методу покрытия решений каждое направление перехода должно быть реализовано, по крайней мере, один раз. Этот метод включает в себя критерий покрытия операторов, так как при выполнении всех направлений переходов выполнятся все операторы, находящиеся на этих направлениях.

Для программы, приведенной на рис. Л5.1, покрытие решений может быть выполнено двумя тестами, покрывающими пути  $\{ace, abc!\}$ , либо  $\{acc1, abe\}$ . Для этого выберем следующие исходные данные;  $\{A = 3, B = 0, X = 3\}$  — в первом случае и  $\{A = 2, B = 1, X = 1\}$  — во втором. Однако путь, где  $X$  не меняется, будет проверен с вероятностью 50 %: если во втором условии вместо условия  $X > 1$  записано  $X < 1$ , то ошибка не будет обнаружена двумя тестами.

Результаты тестирования приведены в табл. Л5.2.

Таблица Л5.2. Результат тестирования методом покрытия решений

Тест	Ожидаемый результат	Фактический результат	Результат тестирования
$A = 3, B = 0, X = 2$	$X = 1$	$X = 1$	Неуспешно
$A = 2, B = 1, X = 1$	$X = 2$	* П	Успешно

#### Метод покрытия условий

Этот метод может дать лучшие результаты по сравнению с предыдущими. В соответствии с методом покрытия условий записывается число тестов, достаточное для того, чтобы все возможные результаты каждого условия в решении выполнялись, по крайней мере, один раз.

В рассматриваемом примере имеем четыре условия:  $\{A > 1, 5=0\}$ ,  $\{A = 2, X > 1\}$ . Следовательно, требуется достаточное число тестов, такое, чтобы реализовать ситуации, где  $A > 1$ ,  $A < 1$ ,  $5 = 0$  и  $5 \neq 0$  в точке  $a$  и  $5 = 2$ ,  $A * 2$ ,  $X >$  и  $T < 1$  в точке  $b$ . Тесты, удовлетворяющие критерию покрытия условий (табл. Л5.3), и соответствующие им пути:

- а)  $A = 2, 5 = 0, X=4 ace$ ;
- б)  $A = 1, 5 = 1, X=0 abc!$ .

Таблица Л5.3. Результаты тестирования методом покрытия условий

Тест	Ожидаемый	Фактический	Результат
	результат	результат	тестирования
$5 = 2, B = 0, X = 4$	$* = 3$	$X = 3$	Неуспешно
$A = 1, B = 1, X = 0$	$* = 0$	$X = 1$	Успешно

#### Метод покрытия решений/условий

Критерий покрытия решений/условий требует такого достаточного набора тестов, чтобы все возможные результаты каждого условия выполнялись по крайней мере один раз, все результаты каждого решения выполнялись по крайней мере один раз и, кроме того, каждой точке входа передавалось управление по крайней мере один раз.

Недостатки метода:

- не всегда можно проверить все условия;
- невозможно проверить условия, которые скрыты другими условиями;
- недостаточная чувствительность к ошибкам в логических выражениях.

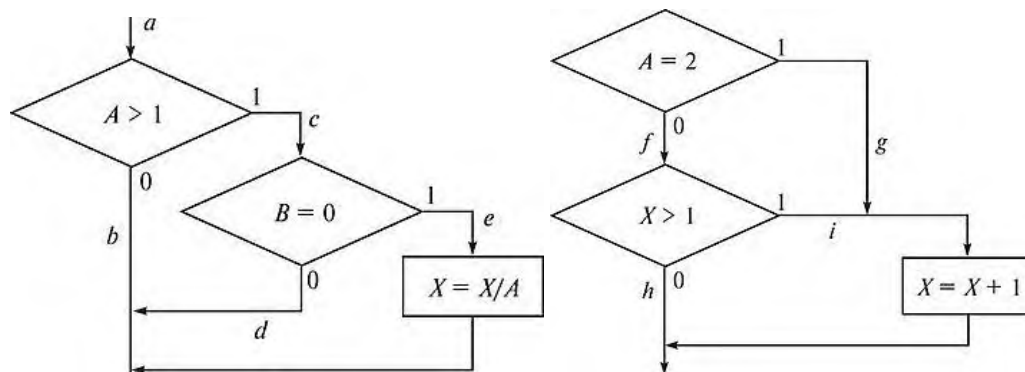
Так, в рассматриваемом примере два теста метода покрытия условий

а)  $A = 2, B = 0, X = 4 ace$ ;

б)  $A = 1, B = 1, X = 0 abc!$

отвечают и критерию покрытия решений/условий. Это является следствием того, что одни условия приведенных решений скрывают другие условия в этих решениях. Так, если условие  $A > 1$  будет ложным, транслятор может не проверять условия  $B = 0$ , поскольку при любом результате условия  $B = 0$  результат решения  $((A >) \& (B = 0))$  примет значение *ложь*. То есть в варианте на рис. Л5.1 не все результаты всех условий выполняются в процессе тестирования.

Рассмотрим реализацию того же примера на рис. Л5.2. Наиболее полное покрытие тестами в этом случае осуществляется



так, чтобы выполнялись все возможные результаты каждого простого решения. Для этого нужно покрыть пути  $acseg$  (тест  $A = 2, B = 0, X = 4$ ),  $ac\bar{e}/k$  (тест  $A = 3, B = 1, X = 0$ ),  $ab/k$  (тест  $A = 0, B = 0, X = 0$ ),  $abp$  (тест  $A = 0, B = 0, X = 2$ ).

Протестировав алгоритм на рис. Л5.2, нетрудно убедиться в том, что критерии покрытия условий и критерии покрытия решений/условий недостаточно чувствительны к ошибкам в логических выражениях.

#### Метод комбинаторного покрытия условий

Критерий комбинаторного покрытия условий удовлетворяет также и критериям покрытия решений, покрытия условий и покрытия решений/условий.

Этот метод требует создания такого числа тестов, чтобы все возможные комбинации результатов условия в каждом решении выполнялись по крайней мере один раз. По этому критерию в рассматриваемом примере должны быть покрыты тестами следующие восемь комбинаций:

- 1.  $A > 1, B = 0, 5. A = 2, X > 1$ .
- 2.  $A > 1, B \neq 0$ .
- 3.  $A < 1, B = 0$ .
- 4.  $A < 1, B \neq 0$ .
- 6.  $A = 2, 1$ .
- 7.  $A \neq 2, X > 1$ .
- 8.  $A \neq 2, X < 1$ .

Для того чтобы протестировать эти комбинации, необязательно использовать все 8 тестов. Фактически они могут быть покрыты четырьмя тестами (табл. Л5.4):

- •  $A = 2, B = 0, X = 4$  (покрывает 1,5);
- •  $A = 2, B = 1, X = 1$  (покрывает 2,6);
- •  $A = 0,5, B = 0, X = 2$  (покрывает 3, 7);
- •  $A = 1, B = 0, X =$  (покрывает 4, 8).

Таблица Л5.4. Результаты тестирования методом комбинаторного покрытия условий

Тест	Ожидаемый результат	Фактический результат	Результат тестирования
$A=2, B=0, X=4$	$\neq 3$	$\neq 3$	Неуспешно
$A=2, B=1, X=1$	$X=2$	1 Г) СЧ П	Успешно
$A=0,5, B=0, X=2$	$X=3$	$X=4$	Успешно
$A=1, B=0, X=1$	$X=1$	$X=1$	Неуспешно

**Задание № 1.** В Древней Греции (II в. до н.э.) был известен шифр, называемый "квадрат Полибия". Шифровальная таблица представляла собой квадрат с пятью столбцами и пятью строками, которые нумеровались цифрами от 1 до 5. В каждую клетку такого квадрата записывалась одна буква. В результате каждой букве соответствовала пара чисел, и шифрование сводилось к замене буквы парой чисел. Для латинского алфавита квадрат Полибия имеет вид:

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I, J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Пользуясь вышеизложенным способом написать программу, которая:

- зашифрует введенный текст и выведет на экран;
- считает зашифрованный текст и расшифрует данный текст (пункт б выполнить по желанию и возможностям).

**Задание № 2. Выбрать один из методов.** Спроектировать тесты по принципу «белого ящика» для программы, разработанной в задании № 1. Выбрать алгоритм для тестирования, обозначить буквами или цифрами ветви этих алгоритмов. Выписать пути алгоритма, которые должны быть проверены тестами для выбранного метода тестирования. Записать тесты, которые позволят пройти по путям алгоритма. Протестировать разработанную вами программу. Результаты оформить в виде таблицы:

Тест	Ожидаемый результат	Фактический результат	Результат тестирования
...	...	...	...

**Задание № 3.** Проверить все виды тестов и сделать выводы об их эффективности.



## Практическое занятие «Использование инструментария анализа качества»

**Цель:** получить навыки использования инструментария анализа качества.

### Теоретические вопросы

Общие требования к качеству функционирования сложных программных комплексов.

Требования к характеристикам качества сложных программных комплексов.

Требования к эффективности использования ресурсов ЭВМ программным комплексом в реальном времени.

Проверка корректности функциональных требований к сложным комплексам программ.

**Задание № 1.** Написать программу, генерирующую массив вещественных чисел в диапазоне от  $-10$  до  $10$  и определяющую все минимальные положительные элементы.

**Задание № 2.** Оценить эффективность разработанной программы:

	Исходная программа		Улучшенная программа	
	Недостатки	Количественная оценка	Улучшения	Количественная оценка
Время выполнения				
Оперативная память				
Внешняя память				

**Задание № 3.** Оценить качество разработанной программы:

	Правильность	Универсальность	Проверяемость	Точность результатов
Недостатки				
Оценка				

**Задание № 4.** Оформить отчет.

## Практическое занятие «Анализ и обеспечение обработки исключительных ситуаций»

**Цели:** получение навыков анализа и обеспечения обработки исключительных ситуаций.

**Задание № 1.** Написать программу, в которой обрабатываются следующие исключительные ситуации: "отрицательное значение возраста" и "год рождения, больше текущего". То есть не может быть осуществлен ввод отрицательного возраста и не может быть осуществлен ввод года рождения больше 2020.

**Задание № 2.** Составить программу циклического вычисления значений функций, определенных из таблицы вариантов заданий. Значения  $R$  должны вводиться с клавиатуры.  $R_1$  и  $R_2$  – вещественные,  $R_3$  – комплексное. Предусмотреть вывод подсказок в виде (например):

Funkciya sin(x)  
 Q –Vyihod iz programmyi  
 Vvedite chislo ili Q:

Для вычисления значений функции написать функцию, вычисляющую требуемые по заданию значения.

Предусмотреть анализ всей введенной информации на ошибки. Вывод на экран и чтение с клавиатуры организовать при помощи стандартных потоков ввода/вывода/ошибки.

Вывести исходные данные и результат в виде(например): Sin(R) = rez,  
 где rez –результаты вычисления (вещественный).

Вариант	Функция	Обработчики
1	$\sin(R1) \cdot \pi / R2 - R3$	Потеря разряда Деление на 0
2	$\sin(R2) / \pi \cdot R1 + R3$	Потеря разряда Переполнение
3	$\tan(R1) / R3 + \text{Cmod}(R3)$	Потеря разряда Прерывание
4	$\arctan(R1) \cdot R2 + R3$	Потеря разряда Переполнение
5	$\ln(R1 - R2) \cdot R2 - R3$	Обл.опр.арг. Исчезновение порядка

Задание № 3. Оформите отчет.

## Практическое занятие «Функциональное тестирование»

*Цель работы:* изучение структуры плана тестирования и получение практических навыков создания тестовых наборов.

### Методические указания

Тест план (Test Plan) – это документ, описывающий весь объем работ по тестированию, начиная с описания тестируемых объектов, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения ([http:// wiki.software- testing.ru/Тест- план](http://wiki.software-testing.ru/Тест-план)). Тест-кейс (Test Case) – набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства. Под тест-кейсом также может пониматься соответствующий документ, представляющий формальную запись тест-кейса [Куликов]. Тест-кейс всегда содержит набор входных значений, предусловий, пошаговое описание и постусловия для каждой проверки, ожидаемый результат.

Набор тест-кейсов называется тестовым набором (Test Suite).

Цели создания тест-плана.

- Согласование объёмов и стратегии тестирования различных составляющих тестируемого ПО с другими участниками проектной команды.

- Расстановка приоритетов задач тестирования.
- Своевременное планирование ресурсозатрат на тестирование.
- Учёт требуемых ресурсов (ПО, оборудование), необходимых для тестирования.

- Заблаговременный учёт рисков, которые могут возникнуть в процессе реализации плана, и внедрение предупреждающей стратегии

Содержание тест-плана отвечает на вопросы:

1. что надо тестировать (объект тестирования: система, приложение, оборудование);
2. что будете тестировать (список функций и компонент тестируемой системы);
3. как будете тестировать (стратегия тестирования – виды тестирования и их применение по отношению к тестируемому объекту);
4. тестовые окружения, на которых необходимо проверять программный продукт;
5. последовательность проведения работ: подготовка, тестирование, анализ результатов, учёт зависимостей тестовых активностей от задач разработки и смежных групп;
6. риски и стратегии по их разрешению;
7. перечень согласовывающих лиц;
8. принятые стандарты и шаблоны;
9. критерии начала и окончания тестирования.

Порядок выполнения:

1. Разработать базовую структуру тестового плана в соответствии с разделами, представленными в примере выполнения работы.
2. Создать набор тест-кейсов по тестированию графического интерфейса веб-приложения.
3. Обобщить полученную информацию и сделать выводы.

### **Пример выполнения практической работы**

I. Тест-план.

Версия 1.1 от 25.04.2019 Разработан

Иванова

История исправлений

Дата

Версия

Описание

Автор

24.04.2020 1.0 создан

Иванова Виктория

Содержание.

Введение.

1. Цель и объект тестирования.

Объект тестирования.

Цель тестирования.

2. Виды и техники тестирования.

2.1. Функциональное тестирование.

2.2. Конфигурационное тестирование.

2.3. Тестирование производительности.

- 2.4. Стресс тестирование.
- 2.5. Юзабилити тестирование.
- 2.6. Тестирование графического интерфейса.
- 2.7. Тестирование совместимости.
- 3. Требования к численности и квалификации персонала.
- 3.1 Оценка объема работ.
- 3.2 Распределение по ролям и квалификации.
- 4. Программные средства.
- 5. Комплекс документации для проведения тестирования.
- 6. Ответственные.
- 7. Риски и дополнительные обстоятельства.

Введение.

#### 1. Цель и объект тестирования

Объектом тестирования является сайт [туристической компании Русский город](http://ruscity.ru/) (<http://ruscity.ru/>)

*Рис. 9.5. Фрагмент главной страницы веб-приложения туристической фирмы Русский город*

В туристической отрасли большое значение имеет знание потенциальными клиентами, в какие туристические фирмы можно обратиться и какую информацию можно получить с сайта компании. Часто по внешнему виду ресурса посетители составляют мнение о туроператоре и сервисе, который он предоставляет. Поэтому требуется, чтобы сайт туроператора был функционально наполненным, имел современный дизайн и собственный стиль.

На основании присланных запросов потенциальных экскурсантов администрация веб-системы может составить базу данных телефонных номеров, которую можно использовать для новостной рассылки.

#### 1.2. Цель тестирования.

Обеспечение эффективной работы сайта компании туристической компании Русский город (<http://ruscity.ru/>)

#### 2. Виды и техники тестирования

##### 2.1. Функциональное тестирование

2.1.1. Ручное тестирование методом черного ящика при тестировании графического интерфейса: [переходов по разделам верхнего и левого меню](#), нажатие на кнопки Далее.

2.1.2. Автоматизированное и регрессионное тестирование формы Обратного звонка методом черного ящика.

##### 2.2. Конфигурационное тестирование

Ручное тестирование работоспособности приложения для различных разрешениях монитора рабочего места, мобильных телефонов и планшетов.

##### 2.3. Юзабилити тестирование

Ручное тестирование интерфейса сайта с позиций удобства его использования клиентом.

##### 2.4. Стрессовое тестирование.

Автоматизированное тестирование проводится для обеспечения работоспособности сайта при превышении возможного количества одновременно находящихся на сайте посетителей.

##### 2.5. Нагрузочное тестирование

Автоматизированное тестирование проводится для обеспечения работоспособности сайта при большом количестве посетителей.

#### 2.6. Тестирование графического интерфейса.

Провести функциональное и нефункциональное тестирование графического интерфейса веб-приложения ручным способом.

#### 2.7. Тестирование совместимости.

Провести тестирование совместимости с последними версиями браузеров Google Chrome, Internet Explorer, Safari, Mozilla Firefox с помощью специализированных онлайн-сервисов.

### 3. Требования к численности и квалификации персонала

Тестовый менеджер 1 чел.

Тест-дизайнер – 1 чел.

Тестирующий – 1 чел.

Общие трудозатраты на тестирование составляют 50 человеко-часов.

### 4. Программные средства

4.1. Управление тестированием BTS Selenium IDE.

4.2. Планирование последовательности выполнения работ в среде Trelo (<https://trello.com/user89366602/boards>).

4.3. Браузер Google Chrome.

4.4. Операционная система Windows 7, 8 или 10.

### 5. Комплекс документации для проведения тестирования:

1) Тест-план (Test plan);

2) Тест-кейсы (Test cases);

3) Тестовый набор (Test Suite).

4) Чек-лист (Check-list);

5) Отчет о тестировании (Test report);

6) Отчет об инцидентах (Incident report);

7) Правила и методика испытаний;

### 6. Ответственные.

Тест-менеджер Петрова – за процесс тестирования.

Тест-дизайнер Иванова – за составление тест-кейсов и метрик тестирования.

Тестирующий Горбачева – за проведение тестирования по разработанным тест-кейсам.

### 7. Риски и дополнительные обстоятельства.

При болезни членов тест-группы на его замену назначается свободный от тестирования сотрудник. В случае возникновения форс-мажорных обстоятельств, связанных с технической поддержкой процесса тестирования, тест-менеджером составляется акт о причинах произошедшего. Сроки тестирования согласовываются с заказчиком.

### **Набор тест-кейсов для тестирования проекта**

Таблица

Тест-кейсы для тестирования графического интерфейса

Номер кейса

Название сценария

Последовательность действий

Ожидаемый результат

1.

Открытие сайта через браузер  
Google Chrome

1.

Набрать в адресной строке Google Chrome адрес сайта <http://ruscity.ru/>

2. Перейти на сайт

Открытие сайта турфирмы Русский город

2.

Заккрытие приложения при нажатии на системную кнопку  
«Заккрыть»

1. Проверить закрытие веб- приложения при нажатии на системную кнопка  
«Заккрыть»

Веб-приложение закроется

3.

Расположение текста внутри текстового поля

1. Перейти на сайт <http://ruscity.ru/>

2. Открыть вкладку

Программы

Текстовая информация находится внутри текстового поля

4.

Переход по вкладкам верхнего меню.

3. Открыть на сайте <http://ruscity.ru/>

4. Вкладку О компании

5. Нажать на логотип компании.

1.Осуществляется переход на страницу О компании.

Название страницы соответствует.

2. Возврат на главную страницу

5.

Кликабельность кнопки

Online бронирования номеров

1. Перейти на вкладку

Сборные туры

2. Сделать одинарный щелчок на кнопке

Online бронирования

1.

Переход на страницу Сборные туры осуществлен

2.

Переход на страницу раннего бронирования номеров

6.

Кликабельность кнопки Поиск по дате

1. Перейти на вкладку

Сборные туры

2. Сделать одинарный щелчок на кнопке

Поиск по дате

3. Ввести диапазон дат с 25.02.2020 по

31.03.2020 1.Переход на страницу

Сборные туры осуществлен

2. Открытие веб-форм для ввода дат начала и конца запрашиваемого диапазона

3. Заполнение полей

7.

Запрос по диапазону дат

1. Перейти на вкладку

Сборные туры.

2. Ввести диапазон дат для запроса экскурсий с

25.02.2019 по 31.03.2019 1. Переход на страницу Сборные туры осуществлен

2. Открытие веб- форм для ввода дат начала и конца запрашиваемого диапазона

3.

Вывод список экскурсий, проводимых в указанный диапазон

8.

Работоспособность ссылки

Петербург классический 2 дня

1. Перейти на страницу

Группы для взрослых

2. Нажать на ссылку

Петербург классический 2 дня

1. Переход на страницу

Группы для взрослых

2. Открытие страницы с информацией об экскурсии

Петербург классический 2 дня

9.

Подписка на новости сайта через форму

Подписка на рассылку

1.Найти в футуре сайта

Подписку на рассылку

2.Заполнить ее данными:

Ваше имя: Антонина;

Ваш e-mail: guar@mail.ru

Ввести капчу.

Нажать на кнопку

Подписаться

1. Подписка на рассылку найдена.

2. Веб-формы заполнены.

3. Капча введена.

4. 4.Кнопка нажата и подписка оформлена.

10.

Отображение приложения на мобильном устройстве

Samsung Galaxy S2 1. Запустить эмулятор <http://quirktools.com>>Scree nfly.

2.

Ввести <http://ruscity.ru/>

3.

Перейти на вкладку с пиктограммой мобильного устройства.

4.

Выбрать Samsung  
Galaxy S2.

5.

Сравнить изображение на эмуляторе мобильного устройства и на персональном компьютере.

1. Отображение главной страницы совпадает с изображением на персональном компьютере.

### **Практическое занятие «Тестирование безопасности»**

1 Порядок моделирования угроз безопасности информации и разработки моделей угроз безопасности информации.

2 Определение возможных негативных последствий от реализации угроз безопасности информации.

3 Оценка условий реализации угроз безопасности информации.

4 Источники угроз безопасности информации и оценка возможностей нарушителей.

5 Определение сценариев реализации угроз безопасности информации.

6 Оценка уровней опасности угроз безопасности информации.

### **«Тестирование безопасности операционной системы Windows»**

1 Исследование надежности пароля администратора.

2 Очистка списков недавних мест и программ.

3 Очистка списка USB-накопителей.

4 Очистка кэша и истории браузеров.

5 Удаление записи DNS.

6 Удаление списка последних документов MS Office.

### **Практическое занятие «Нагрузочное тестирование, стрессовое тестирование»**

1. Что такое нагрузочное тестирование?

2. Суть нагрузочного тестирования?

3. Цель нагрузочного тестирования?

4. Основные характеристики нагрузочного тестирования?

5. Цель стресс-тестирования?

6. Основные характеристики стресс-тестирования?

7. Различия нагрузочного и стресс теста?

#### **Задание № 1**

Разработать Компилятор простых арифметических выражений, например  $2 + +(-5)*(7-8)$ .

Вход и выход осуществляются в виде строк.

#### **Задание № 2**

Разработать тестовый сценарий нагрузочного тестирования. Ответить на вопрос – сколько запросов в секунду может обработать приложение при условии, что они идут последовательно.

Построить график зависимости времени ответа от количества параллельных запросов

(рассматривать логарифмическую шкалу по основанию два, т.е. 1, 2, 4, 8, 16, 32 и т.д. запроса)

Ответить на вопрос – какое максимальное количество параллельных запросов может обработать приложение без сбоев.

#### **Задание № 3**

Оформить отчет.

### **Практическое занятие «Тестирование интеграции»**

Цель: получение навыков тестирования интеграции.



Теоретические вопросы

1. Особенности тестирования интеграции.
2. Методы интеграционного тестирования.
3. Нисходящее тестирование интеграции.
4. Восходящее тестирование интеграции.
5. Сравнение нисходящего и восходящего тестирования интеграции

#### **Задание № 1**

Разработать приложение, состоящее из трех модулей:

- 1) главный модуль, считывающий из текстового файла координаты точек на плоскости;
- 2) модуль, содержащий функции расчета расстояния между двумя точками;
- 3) модуль, содержащий функцию, определяющую треугольник с максимальной площадью.

#### **Задание № 2**

Описать этапы нисходящего проектирования разработанного приложения.

#### **Задание № 3**

Описать этапы восходящего проектирования разработанного приложений.

#### **Задание № 4**

Оформить отчет.

### **Практическое занятие «Конфигурационное тестирование»**

Цели: получение навыков проведения конфигурационного тестирования.

Теоретические вопросы

1. Особенности конфигурационного тестирования.
2. Конфигурационное тестирование (Configuration testing).

#### **Задание № 1**

Дана структура с именем ZNAK, состоящая из полей:

- фамилия, имя;
- знак Зодиака;
- дата рождения (массив из трех чисел).

Написать программу, которая выполняет следующие действия:

- ввод с клавиатуры данных в массив, состоящий из 8 элементов типа ZNAK, и занесение их в файл данных;
- чтение данных из файла и вывод их на экран;
- вывод на экран информации о людях, родившихся в месяц, значение которого введено с клавиатуры (если таких нет – вывести об этом сообщение);
- список должен быть упорядочен по знакам Зодиака.

#### **Задание № 2**

Описать и обосновать итоги тестирования работы разработанного приложения на различных платформах: различных вариантах аппаратной конфигурации, версиях операционной системы и окружения.

### **Практическое занятие «Тестирование установки»**

**Цель:** получение навыков тестирования установки.

**Теоретические вопросы**

1. Комплексное тестирование приложения.

#### **Задание №1**

Разработать приложение, интерфейс которого представлен на рисунке 1.

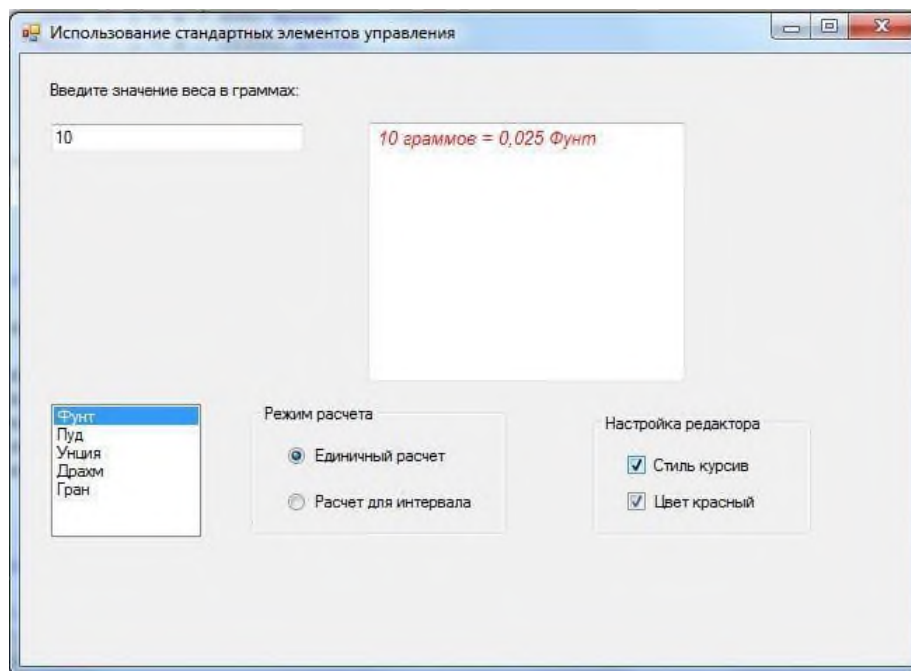


Рисунок 1. Интерфейс приложения

### Задание №2

Провести комплексное тестирование разработанного приложения.

### Задание №3

Оформить отчет.

## Контрольные вопросы.

1. Эскизное проектирование. Основные задачи.
2. техническое проектирование. Состав проектной документации.
3. Рабочее проектирование. Основные задачи.
4. Состав проектной документации стадии рабочего проектирования.
5. Постановка задачи.
6. Стадия ввода в эксплуатацию.
7. Основные особенности внедрения ЭИС.
8. Распределение обязанностей на стадии ввода в эксплуатацию.
9. Виды испытаний информационных систем на стадии ввода в эксплуатацию. Основные понятия и определения ИС. Жизненный цикл информационных систем
10. Организация и методы сбора информации. Анализ предметной области. Основные понятия системного и структурного анализа.
11. Постановка задачи обработки информации. Основные виды, алгоритмы и процедуры обработки информации, модели и методы решения задач обработки информации.
12. Основные модели построения информационных систем, их структура, особенности и области применения.
13. Сервисно - ориентированные архитектуры. Анализ интересов клиента. Выбор вариантов решений
14. Методы и средства проектирования информационных систем. Case-средства для моделирования деловых процессов (бизнес-процессов). Инструментальная среда - структура, интерфейс, элементы управления.
15. Принципы построения модели IDEF0: контекстная диаграмма, субъект моделирования, цель и точка зрения.
16. Диаграммы IDEF0: диаграммы декомпозиции, диаграммы дерева узлов, диаграммы только для экспозиции (FEO).
17. Работы (Activity). Стрелки (Аpow). Туннелирование стрелок. Нумерация работ и диаграмм. Каркас диаграммы.
18. Слияние и расщепление моделей.
19. Особенности информационного, программного и технического обеспечения различных видов информационных систем. Экспертные системы. Системы реального времени
20. Оценка экономической эффективности информационной системы. Стоимостная оценка

проекта. Классификация типов оценок стоимости: оценка порядка величины, концептуальная оценка, предварительная оценка, окончательная оценка, контрольная оценка.

21. Основные процессы управления проектом. Средства управления проектами
22. Основные понятия качества информационной системы. Национальный стандарт обеспечения качества автоматизированных информационных систем.
23. Международная система стандартизации и сертификации качества продукции. Стандарты группы ISO.
24. Методы контроля качества в информационных системах. Особенности контроля в различных видах систем

Решение задач по темам:

1. Определение конфигурации информационной системы. Выбор технических средств.
2. Формирование репозитория проекта, определение уровня доступа в системе контроля версий. Распределение ролей
3. Настройки среды разработки
4. Мониторинг разработки проекта. Сохранение версий проекта
5. Требования к интерфейсу пользователя. Принципы создания графического пользовательского интерфейса (GUI).
6. Понятие спецификации языка программирования. Синтаксис языка программирования. Стил программирования
7. Основные конструкции выбранного языка программирования. Описание переменных, организация ввода-вывода, реализация типовых алгоритмов
8. Создание сетевого сервера и сетевого клиента.
9. Разработка графического интерфейса пользователя.
10. Отладка приложений. Организация обработки исключений.
11. Виды, цели и уровни интеграции программных модулей.
12. Выбор источников и приемников данных, сопоставление объектов данных.
13. Транспортные протоколы. Стандарты форматирования сообщений.
14. Организация файлового ввода-вывода.
15. Процесс отладки. Отладочные классы.
16. Спецификация настроек типовой ИС.
17. Понятие «тестирования информационных систем».
18. Типы ошибок и ручные методы тестирования

19. Критерии тестирования.
20. Принципы тестирования.
21. Классификация тестирования.
22. Виды и методы тестирования (в том числе автоматизированные).
23. Тестирование «белого ящика».
24. Тестирование «черного ящика».
25. 25.Функциональное тестирование.
26. Нефункциональное тестирование.
27. Тесты в процессе разработки ИС.
28. Проектирование тестирования.
29. Организация тестирования в команде разработчиков.
30. Тестовые сценарии, тестовые варианты. Оформление результатов тестирования
31. Инструментарии анализа качества программных продуктов в средеразработке.
32. Обработка исключительных ситуаций. Методы и способы идентификацииисбоев и ошибок.
33. Выявление ошибок системных компонентов.
34. Реинжиниринг бизнес-процессов в информационных системах.

## ТЕСТ

Первый вариант языка SQL назывался

+ **SEQUEL**

-QUEL

-DDL

-DML

ANSI SQL- это

+ **стандарт на язык**

- детальное описание языка

- новейший язык манипулирования данными

- расширение языка SQL

Команды языка SQL подразделяются на команды языка

+ **определения данных**

+ **манипулирования данными**

- преобразования данных

- хранения данных

Команды языка SQL подразделяются на команды языка

+ **DDL**

+ **DML**

-DNL

-DBL

Команды языка SQL подразделяются на команды языка

+ **DCL**

+ **DQL**

-DPL

-DSL

Команды языка SQL подразделяются на команды

+ **администрирования базы данных**

+ **управления транзакциями**

- нормализации базы данных

Модернизации базы данных Значение NULL эквивалентно

+ **отсутствию информации**

- цифре ноль

- пробелу

прочерку Представление

- ничем не отличается от таблицы

- постоянно хранит какие-либо данные

- отличается от таблицы только форматированием

+ **большую часть времени не содержит данных** Хранимые процедуры представляют собой

+ **группы связанных SQL - операторов**

- подпрограммы

- правила хранения данных

- процедуры резервного копирования Триггеры представляют собой

**+ разновидность хранимых процедур**

- способ хранения данных
- процедуры резервного копирования
  - функции защиты данных от несанкционированного доступа

Разграничение доступа к информации, хранящейся в базе данных, регулируется с помощью привилегии

**+ на создание таблицы**

- SELECT
- INSERT
- UPDATE

Разграничение доступа к информации, хранящейся в базе данных, регулируется с помощью привилегии

**+ на создание хранимой процедуры**

- REFERENCE
- INSERT (имя\_поля)
- UPDATE (имя\_поля)

Объектными привилегиями являются привилегии

**+ SELECT**

- на создание таблицы
- на создание хранимой процедуры
- на создание представления

Объектными привилегиями являются привилегии

**+UPDATE**

- на удаление таблицы
- на удаление представления
  - на удаление хранимой процедуры Привилегия REFERENCE разрешает

**+ ссылаться на все поля указанной таблицы**

- создавать и удалять таблицы, представления и хранимые процедуры
- передавать права доступа другим пользователям
- изменять информацию в базе данных

Для управления доступом пользователей к базеданным в языке SQL существует оператор

**+GRANT**

**+REVOKE**

- REFERENCE
- SELECT

Оператор GRANT служит для

**+ предоставления пользователю как системных, так и объектных привилегий**

- отмены предоставленных пользователю привилегий
- предоставления пользователю системных привилегий
  - предоставление пользователю объектных привилегий Оператор REVOKE служит для
- + отмены предоставленных привилегий**
- предоставление пользователю системных привилегий
  - предоставление пользователю как системных, так и объектных привилегий
  - предоставление пользователю объектных привилегий Power Designer это

**+ система моделирования данных**

-СУБД

- язык программирования высокого уровня

- программа для быстрой разработки сайтов CASE средства могут осуществлять

**+ генерацию документации**

**+ верификацию проекта**

- помощь в принятии решений

- выбор языка программирования или СУБД CASE средства могут осуществлять

**+ автоматическую генерацию программного кода**

**+ сопровождение и реинжиниринг**

- согласование этапов разработки с заказчиком

- оценку стоимости проекта

Возможность определения единственного имени для процедуры или функции, которые применяются ко всем объектам иерархии наследования, является следствием

**+ полиморфизма**

- инкапсуляции

- наследования

- внедрения

Комбинирование данных с процедурами и функциями, манипулирующими этими данными, это следствие

**+ инкапсуляции**

- наследования

- полиморфизма

- связывания

Возможность использования уже определённых классов для построения иерархии классов, производных от них, это -

**+ наследование**

- согласованность классов

- приемственность

- инкапсуляция



## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

<b>№ п.п.</b>	<b>Содержание изменения</b>	<b>Дата, номер протокола заседания кафедры, подпись зав.кафедрой</b>
1	2	3
1		
2		
3		
4		