

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«МОЛОДЕЧНЕНСКИЙ ТОРГОВО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ»  
БЕЛКООПСОЮЗА

**ПЕРЕЧЕНЬ ВОПРОСОВ  
ДЛЯ ПОДГОТОВКИ К ОБЯЗАТЕЛЬНОЙ КОНТРОЛЬНОЙ РАБОТЕ №2**

**Предмет:** «Тестирование программного обеспечения»

**Темы:** «Структурное и функциональное тестирование»,  
«Организация тестирования ПО»

**Специальность:** 2-40 01 01 «Программное обеспечение информационных технологий»

**Группы:** СП305, ПО309

Составил преподаватель:

О.Н. Апетенок

Рассмотрено и одобрено на заседании  
методической (цикловой) комиссии  
программного обеспечения  
информационных технологий и  
информатики  
Протокол № \_\_\_\_\_  
« \_\_\_\_\_ » \_\_\_\_\_ 2022 года  
Председатель цикловой комиссии  
\_\_\_\_\_ С.С. Лаптик

**ПЕРЕЧЕНЬ ВОПРОСОВ**  
**ДЛЯ ПОДГОТОВКИ К ОБЯЗАТЕЛЬНОЙ КОНТРОЛЬНОЙ РАБОТЕ**

1. Виды рабочей тестовой документации и их характеристика.
2. Ошибки, дефекты, сбои, отказы и т. д.
3. Отчёт о дефекте и его жизненный цикл. Атрибуты (поля) отчёта о дефекте
4. Документация, создаваемая на различных этапах жизненного цикла.
5. Документация, сопровождающая процессы верификации и тестирования.
6. Отчеты о прохождении тестов. Отчеты о покрытии программного кода. Отчеты о проблемах. Трассировочные таблицы
7. Цели и задачи модульного тестирования.
8. Понятие о модуле и его границах.
9. Подходы к проектированию тестового окружения.
10. Организация модульного тестирования
11. Цели и задачи интеграционного тестирования. Организация интеграционного тестирования.
12. Структурная классификация методов интеграционного тестирования. Временная классификация методов интеграционного тестирования.
13. Планирование интеграционного тестирования

**ГЛОССАРИЙ**

<b>Термин (по-русски)</b>	<b>Термин (по-английски)</b>	<b>Определение</b>
Автоматизированное тестирование	Automated testing	Набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования.
Альфа-тестирование	Alpha testing	Тестирование, которое выполняется внутри организации-разработчика с возможным частичным привлечением конечных пользователей. Может являться формой внутреннего приёмочного тестирования.
Анализ первопричин	Root cause analysis	Процесс исследования и классификации первопричин возникновения событий, негативно влияющих на безопасность, здоровье, окружающую среду, качество, надёжность и производственный процесс.
Бета-тестирование	Beta testing	Тестирование, которое выполняется вне организации-разработчика с активным привлечением конечных пользователей/заказчиков.
Граничное условие	Border condition, boundary condition	Значение, находящееся на границе классов эквивалентности.
Дефект	Defect, anomaly	Отклонение фактического результата от ожиданий наблюдателя, сформированных на основе требований, спецификаций, иной документации или опыта и здравого смысла.

Динамическое тестирование	Dynamic testing	Тестирование с запуском кода на исполнение.
Дымовое тестирование	Smoke test	Тестирование, которое направлено на проверку самой главной, самой важной, самой ключевой функциональности, неработоспособность которой делает бессмысленной саму идею использования приложения (или иного объекта, подвергаемого дымовому тестированию).
Интеграционное тестирование	Integration testing	Тестирование, которое направлено на проверку взаимодействия между несколькими частями приложения (каждая из которых, в свою очередь, проверена отдельно на стадии модульного тестирования).
Класс эквивалентности	Equivalence class	Набор данных, обрабатываемых одинаковым образом и приводящих к одинаковому результату.
Метод белого ящика	White box testing	Метод тестирования, в рамках которого у тестировщика есть доступ к внутренней структуре и коду приложения, а также есть достаточно знаний для понимания увиденного.
Метод серого ящика	Gray box testing	Комбинация методов белого ящика и чёрного ящика, состоящая в том, что к части кода и архитектуры у тестировщика доступ есть, а к части — нет.
Метод чёрного ящика	Black box testing	Метод тестирования, в рамках которого у тестировщика либо нет доступа к внутренней структуре и коду приложения, либо недостаточно знаний для их понимания, либо он сознательно не обращается к этим данным в процессе тестирования.
Метрика	Metric	Числовая характеристика показателя качества. Может включать описание способов оценки и анализа результата.
Модель разработки ПО	Software Development Model	Структура, систематизирующая различные виды проектной деятельности, их взаимодействие и последовательность в процессе разработки ПО.
Модульное (компонентное) тестирование	Unit testing, component testing	Тестирование, направленное на проверку отдельных частей приложения, которые (как правило) можно исследовать изолированно от других подобных частей.
Набор тест кейсов	Test case suite, test suite, test set	Совокупность тест-кейсов, выбранных с некоторой общей целью или по некоторому общему признаку.
Негативное тестирование	Negative testing	Тестирование, направленное на исследование работы приложения в ситуациях, когда с ним выполняются (некорректные) операции и/или используются данные, потенциально приводящие к ошибкам.
Нефункциональное тестирование	Non-functional testing	Тестирование, направленное на проверку нефункциональных особенностей приложения (корректность реализации нефункциональных требований), таких как удобство использования, совместимость, производительность, безопасность и т.д.
Нефункциональные требования	Non-functional requirements	Требования, описывающие свойства системы (удобство использования, безопасность, надёжность, расширяемость и т.д.), которыми она должна обладать

		при реализации своего поведения.
Отчёт о дефекте	Defect report	Документ, описывающий и приоритизирующий обнаруженный дефект, а также содействующий его устранению
Отчёт о результатах тестирования	Test progress report, test summary report	Документ, обобщающий результаты работ по тестированию и содержащий информацию, достаточную для соотнесения текущей ситуации с тест-планом и принятия необходимых управленческих решений.
Позитивное тестирование	Positive testing	Тестирование, направленное на исследование приложения в ситуации, когда все действия выполняются строго по инструкции без каких бы то ни было ошибок, отклонений, ввода неверных данных и т.д.
Приёмочное тестирование	Acceptance testing	Формализованное тестирование, направленное на проверку приложения с точки зрения конечного пользователя/заказчика и вынесения решения о том, принимает ли заказчик работу у исполнителя (проектной команды).
Расширенное тестирование	Extended test	Тестирование, направленное на исследование всей заявленной в требованиях функциональности — даже той, которая низко приоритизирована по степени важности.
Регрессионное тестирование	Regression testing	Тестирование, направленное на проверку того факта, что в ранее работоспособной функциональности не появились ошибки, вызванные изменениями в приложении или среде его функционирования.
Ручное тестирование	Manual testing	Тестирование, в котором тест-кейсы выполняются человеком вручную без использования средств автоматизации.
Системное тестирование	System testing	Тестирование, направленное на проверку всего приложения как единого целого, собранного из частей, проверенных на стадиях модульного и интеграционного тестирования.
Статическое тестирование	Static testing	Тестирование без запуска кода на исполнение.
Структурная декомпозиция	Work break down structure, WBS	Иерархическая декомпозиция объёмных задач на всё более и более малые подзадачи с целью упрощения оценки, планирования и мониторинга выполнения работы.
Тест	Test	Набор из одного или нескольких тест-кейсов.
Тестирование критического пути	Critical path test	Тестирование, направленное на исследование функциональности, используемой типичными пользователями в типичной повседневной деятельности.
Тестирование под управлением данными	Data-driven testing	Способ разработки автоматизированных тест-кейсов, в котором входные данные и ожидаемые результаты выносятся за пределы тест-кейса и хранятся вне его — в файле, базе данных и т.д.
Тестирование под управлением ключевыми словами	Keyword-driven testing	Способ разработки автоматизированных тест-кейсов, в котором за пределы тест-кейса выносятся не только набор входных данных и ожидаемых результатов, но и логика поведения тест-кейса, которая описывается ключевыми словами (командами).

Тестирование под управлением поведением	Behavior-driven testing	Способ разработки автоматизированных тест-кейсов, в котором основное внимание уделяется корректности работы бизнес-сценариев, а не отдельным деталям функционирования приложения.
Тестирование программного обеспечения	Software testing	Процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта.
Тестирование производительности	Performance testing	Исследование показателей скорости реакции приложения на внешние воздействия при различной по характеру и интенсивности нагрузке
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.
Требование	Requirement	Описание того, какие функции и с соблюдением каких условий должно выполнять приложение в процессе решения полезной для пользователя задачи.
Трудозатраты	Man-hours	Количество рабочего времени, необходимого для выполнения работы (выражается в чело веко-часах).
Функциональная декомпозиция	Functional decomposition	Процесс определения функции через её разделение на несколько низкоуровневых под функций.
Функциональное тестирование	Functional testing	Тестирование, направленное на проверку корректности работы функциональности приложения

## Примерные задачи

---

### ЗАДАЧА

Определить классы эквивалентности. Выбрать представителя каждого класса. Выполнить тесты.

#### УСЛОВИЕ:

На рейсах авиакомпании при оформлении предварительной оплаты за провоз сверхнормативного багажа ранее, чем за 24 часа до вылета рейса по расписанию действует скидка в размере 50% от общего тарифа. При оплате за провоз сверхнормативного багажа во время посадки в самолет действует тариф на 20% выше общего тарифа. Регистрация на рейс начинается за 3 часа до вылета.

---

### ЗАДАЧА

Определить классы эквивалентности. Определить граничные значения для каждого класса (важно понимать к какому классу относится значение). Провести тесты по проверке значения до границы, на границе и сразу после границы.

#### УСЛОВИЕ:

На рейсах авиакомпании при оформлении предварительной оплаты за провоз сверхнормативного багажа ранее, чем за 24 часа до вылета рейса по расписанию действует скидка в размере 50% от общего тарифа. При оплате за

провоз сверхнормативного багажа во время посадки в самолет действует тариф на 20% выше общего тарифа. Регистрация на рейс начинается за 3 часа до вылета.

Предложить вариант выделения классов эквивалентности входных данных для подпрограммы открытия файла по его имени. Тестовые требования к подпрограмме выделены и имеют следующий состав:

- проверить, что присутствия в имени файла символов, не являющимися буквами латинского алфавита и цифрами, система выводит сообщение об ошибке;

- проверить, что в случае, когда длина имени файла превышает 11 символов, система выдает сообщение об ошибке;

- проверить, что система не различает регистр символов имени при открытии файла;

- проверить, что в случае, когда имя файла не противоречит требованиям 1-3, система открывает файл.

---

#### ЗАДАЧА

Разработать unit-тест. Проверить правильно ли вычисляет программа площадь прямоугольника со сторонами 3 и 5. Ожидаемый результат (правильное решение) в данном случае это число 15.

---

#### ЗАДАЧА

В проекте **Calc** содержатся 2 метода, которые необходимо протестировать на корректность работы. Для этого создайте unit-тест, который будет проверять операцию деления двух чисел.

---

#### ЗАДАЧА

В проекте **Calc** содержатся 2 метода, которые необходимо протестировать на корректность работы. Для этого создайте unit-тест, который будет проверять операцию деления на ноль.

---

#### ЗАДАЧА

В проекте **Calc** содержатся 2 метода, которые необходимо протестировать на корректность работы. Для этого создайте unit-тест, который будет проверять операцию сложения двух чисел и инкрементацию полученной суммы.

---

#### ЗАДАЧА

Разработать программу для подсчета объема цилиндра и создать модульный тест.

## Рекомендуемая литература

1. **Бейзер, Б.** Тестирование «черного ящика». Технологии функционального тестирования программного обеспечения и систем / Б. Бейзер. СПб., 2004.
2. **Гленфорд М.**, Искусство тестирования программ, 3-е издание / Гленфорд Майерс, Том Баджетт, Кори Сандлер, Вильямс, 2012, 272 стр. 272, ISBN 978-5-8459-1796-6, 978-1-118-03196-4
3. **Дастин, Э.** Автоматизированное тестирование программного обеспечения / Э. Дастин, Д. Рэшка, Д. Пол. М., 2003.
4. **Иванова, Г.С.** Технология программирования / Г.С. Иванова. М., 2002.
5. **Искусство тестирования программ**, Гленфорд Майерс, Москва: Диалектика, 2019
6. **Калбертсон, Р.** Быстрое тестирование / Р. Калбертсон, К. Браун, Г. Кобб. М., 2006.
7. **Канер С.**, Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес - приложений, Канер С., Москва: Диа Софт, 2018
8. **Канер, С.** Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / С. Канер, Д. Фолк, К.Н. Енг. ДиаСофт, 2018.
9. **Котляров, В.П.** Основы тестирования программного обеспечения / В.П. Котляров, Т.В. Коликова. М., 2006.
10. **Криспин Л.**, Гибкое тестирование. Практическое руководство для тестировщиков ПО и гибких команд / Лайза Криспин, Джанет Грегори; Вильямс, 2016, стр. 464, ISBN 978-5-8459-1625-9, 978-0-321-53446-0
11. **Кулаков К.**, Основы тестирования программного обеспечения: учебное электронное пособие / К. А. Кулаков, В. М. Димитров; Петрозаводск : Издательство ПетрГУ, 2018. ISBN 978-5-8021-3222-7
12. **Макгрегор, Д.** Тестирование объектно-ориентированного программного обеспечения : практ. пособие / Д. Макгрегор, Д. Сайкс. Киев, 2002.
13. **Степанченко, И.В.** Методы тестирования программного обеспечения: учеб. пособие / И.В. Степанченко. Волгоград, 2006.
14. **Тамре, Л.** Введение в тестирование программного обеспечения / Л. Тамре. М., 2003.