

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«УЧЕБНЫЙ ЦЕНТР СКБ КОНТУР»

Утверждаю
Директор АНО ДПО
«Учебный центр СКБ Контур»

 Т.В. Рубан
1 сентября 2023 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

дополнительная общеразвивающая программа

ТЕСТИРОВАНИЕ ПО. БАЗОВЫЙ УРОВЕНЬ

Документ: Свидетельство

Форма обучения: заочная с использованием дистанционных образовательных технологий, электронного обучения

Срок обучения: 15 недель

Объем программы: 152 ак. ч.

Москва, 2023 г.

Оглавление

1. ОБЩИЕ ПОЛОЖЕНИЯ	3
1.1. Назначение программы.....	3
1.2. Нормативные документы, регламентирующие разработку дополнительной общеобразовательной программы - дополнительной общеразвивающей программы	3
2. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	3
2.1. Цели дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.2. Требования к слушателю дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.3. Срок освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	4
2.5. Планируемые результаты обучения по итогам освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.	4
3. СОДЕРЖАНИЕ И ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА.....	5
3.1. Учебный план	5
3.2. Содержание.....	8
3.3 Календарный учебный график.....	20
4.1. Требования к квалификации педагогических кадров, представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса	21
4.2. Требования к материально-техническим условиям.....	21
4.3. Информационные и учебно-методические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы	23
4.4. Общие требования к организации образовательного процесса.....	24
5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	25
5.1. Формы контроля знаний и требования к его проведению	25
5.2. Критерии оценки знаний слушателей	25
5.3. Фонд оценочных средств.....	29

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Назначение программы

Дополнительная общеобразовательная программа дополнительная общеразвивающая программа «Тестирование ПО. Базовый уровень» направлена на получение практических навыков и теоретических знаний, необходимых для тех, кто хочет познакомиться с актуальными инструментами тестирования ПО и освоить навыки тестирования на базовом уровне.

Программа регламентирует цели, планируемые результаты, содержание, условия и технологии реализации образовательного процесса, оценку качества подготовки слушателей и включает в себя: учебный план, фонды оценочных средств, программу итоговой аттестации, календарный учебный график и другие материалы, обеспечивающие качество подготовки слушателей.

1.2. Нормативные документы, регламентирующие разработку дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

1. Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ.
2. Федеральный закон Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
3. Приказ Минпросвещения РФ от 27.07.2022 №629 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам".

2. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

2.1. Цели дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Основная цель программы – дать слушателям базовые знания о тестировании WEB, API, мобильных приложений на iOS и Android.

Задачи дисциплины

В процессе изучения курса «Тестирование ПО. Базовый уровень» необходимо решить следующие задачи и рассказать слушателям:

- о работе с HTML, CSS и SQL на базовом уровне;
- о тестировании API, web- и мобильных приложений;
- о работе в команде удаленно;
- о тестовых стендах;
- о проверке сайтов на уязвимость;
- о том, как работать с Charles Proxy, Android Studio, XCode, Postman, в баг-трекерах и TMS.

2.2. Требования к слушателю дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

К освоению дополнительной общеобразовательной программы – дополнительной общеразвивающей программы допускаются любые лица без предъявления требований к уровню образования.

Возраст слушателей: 18 лет и старше.

2.3. Срок освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Нормативный срок освоения программы – 15 недель.

2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Объем образовательной программы составляет 152 академических часа. Из них 97 – самостоятельная работа, 55 академических часов – работа на образовательной онлайн-платформе.

2.5. Планируемые результаты обучения по итогам освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.

В результате обучения слушатели должны будут овладеть следующими знаниями, умениями и навыками:

Знать:

- основы тестирования ПО;
- техники тест-дизайна;
- основы работы с HTML, CSS и SQL;
- основы тестирования веб-приложений и мобильных приложений
- назначение и использование Git;
- особенности использования тестирования API: назначение, процесс, применимость.

Уметь:

- выбирать и применять различные техники тестирования;
- создавать HTML-документы;
- применять CSS для стилизации веб-страниц;
- тестировать web-приложения на desktop и на смартфонах;
- использовать инструменты для анализа трафика Charles, Fiddler;
- писать тест-кейсы и чек листы для тестирования API;
- тестировать мобильные приложения на Android и iOS;
- составлять связи между таблицами базы данных;
- осуществлять создание, просмотр и редактирование директорий;
- использовать SSH-ключи и работать по SSH-протоколу.
- работать с файлами, локальными и удалёнными репозиториями, тегами.

Владеть:

- навыками тестирования безопасности ПО на базовом уровне;
- навыками работы с различными эмуляторами и симуляторами, а также с фермами;
- навыками проверки сайта на наличие SQL и XSS-уязвимостей;
- навыками работы в команде, в том числе удаленно.

3. СОДЕРЖАНИЕ И ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

3.1. Учебный план для заочной формы обучения с ДОТ и ЭО

№ п/п	Наименование разделов, дисциплин	Всего часов	В том числе		Форма контроля
			Самостоятельная работа	Работа на образовательной онлайн-платформе	
1	Основы тестирования	21	14	7	Зачет
1.1	Введение в тестирование ПО	3	2	1	Тестирование
1.2	Тест-анализ и тест-дизайн: знакомство	3	2	1	Тестирование
1.3	Составление тестовой документации: чек-листы и Mind Мар	3	2	1	Тестирование
1.4	Составление тестовой документации: тест-кейсы	3	2	1	Тестирование
1.5	Составление баг-репортов	3	2	1	Тестирование
1.6	Эвристики и мнемоники в тестировании	3	2	1	Тестирование
1.7	Метрики и отчеты о тестировании	3	2	1	Тестирование
2	Тест-анализ и техники тест-дизайна	21	14	7	Зачет
2.1	Требования к продукту. Анализ, визуализация и декомпозиция	3	2	1	Тестирование
2.2	Классы эквивалентности	3	2	1	Тестирование
2.3	Попарное тестирование	3	2	1	Тестирование
2.4	Граничные значения и доменный анализ	3	2	1	Тестирование
2.5	Таблицы принятия решений	3	2	1	Тестирование
2.6	Переходы состояний	3	2	1	Тестирование
2.7	Другие техники тест-дизайна (причина-следствие, предугадывание ошибок, исследовательское тестирование)	3	2	1	Тестирование
3	Работа с HTML и CSS	12	8	4	Зачет
3.1	Введение в веб-разработку	3	2	1	Тестирование
3.2	Основы HTML	3	2	1	Тестирование
3.3	Основы CSS	3	2	1	Тестирование

№ п/п	Наименование разделов, дисциплин	Всего часов	В том числе		Форма контроля
			Самостоятельная работа	Работа на образовательной онлайн-платформе	
3.4	Разработка веб-приложения	3	2	1	Тестирование
4	Тестирование Веб	9	6	3	Зачет
4.1	Основы тестирования веб-приложений	3	2	1	Тестирование
4.2	DevTools	3	2	1	Тестирование
4.3	Сбор и анализ трафика	3	2	1	Тестирование
5	Тестирование API	12	8	4	Зачет
5.1	Клиент-серверная архитектура	3	2	1	Тестирование
5.2	REST API	3	2	1	Тестирование
5.3	SOAP API	3	2	1	Тестирование
5.4	Техники тест-дизайна при тестировании API и составление тестовой документации	3	2	1	Тестирование
6	Тестирование мобильных приложений	18	12	6	Зачет
6.1	Как устроены мобильные приложения	3	2	1	Тестирование
6.2	Особенности тестирования мобильных приложений	3	2	1	Тестирование
6.3	Тестирование мобильных приложений Android	3	2	1	Тестирование
6.4	Среда разработки Android Studio и Android Debug Bridge (ADB)	3	2	1	Тестирование
6.5	Тестирование мобильных приложений iOS	3	2	1	Тестирование
6.6	Сбор и анализ трафика	3	2	1	Тестирование
7	Базовый SQL	15	10	5	Зачет
7.1	Введение в SQL	3	2	1	Тестирование
7.2	Главный SQL запрос: оператор SELECT	3	2	1	Тестирование
7.3	Объединение данных: оператор JOIN	3	2	1	Тестирование
7.4	Группировка данных	3	2	1	Тестирование
7.5	Модификация данных: INSERT, UPDATE, DELETE	3	2	1	Тестирование
8	Командная строка	21	14	7	Зачет
8.1	Базовые команды в bash: навигация	3	2	1	Тестирование

№ п/п	Наименование разделов, дисциплин	Всего часов	В том числе		Форма контроля
			Самостоятельная работа	Работа на образовательной онлайн-платформе	
8.2	Директории: создание, просмотр и редактирование директорий	3	2	1	Тестирование
8.3	Файлы: создание, просмотр, редактирование и копирование	3	2	1	Тестирование
8.4	Поиск по файлам и директориям	3	2	1	Тестирование
8.5	Ввод и вывод	3	2	1	Тестирование
8.6	Работа с процессами	3	2	1	Тестирование
8.7	Удаленные ресурсы	3	2	1	Тестирование
9	Git	21	11	10	Зачет
9.1	Что такое Git: установка и настройка	4	2	2	Тестирование
9.2	Основные команды Git	4	2	2	Тестирование
9.3	Ветки	4	2	2	Тестирование
9.4	UI Tools в Git	4	2	2	Тестирование
9.5	Командная работа в Git	3	2	1	Тестирование
9.6	Использование Git для решения рабочих задач тестировщика	2	1	1	Тестирование
ИТОГОВАЯ АТТЕСТАЦИЯ		2	–	2	Зачет
Всего:		152	97	55	–

3.2. Содержание

Тема 1. «Основы тестирования»

1.1. «Введение в тестирование ПО (базовые понятия и виды тестирования)»

- История развития подходов к тестированию программного обеспечения.
- Понятия «тестировщик», «тестирование», «quality control (контроль качества продукта)», «quality assurance (обеспечение качества)».
- Цели тестирования ПО.
- Этапы жизненного цикла тестирования
- Уровни тестирования по значимости функциональности, по степени детализации.
- Методы функционального и нефункционального тестирования.
- Виды и назначение тестовой документации.
- Практическое задание: продумать несколько основных проверок любого мобильного приложения

1.2. «Тест-анализ и тест-дизайн: знакомство»

- Тест-анализ, его назначение, функциональные и нефункциональные требования.
- Основные подходы (техники) тест-анализа.
- Понятие и назначение тест-дизайна.
- Популярные техники тест-дизайна.
- Практическое задание: составить тест-кейсы для полной проверки формы авторизации.

1.3. «Составление тестовой документации: чек-листы и Mind Map»

- Чек-лист, его назначение при тестировании ПО.
 - Два типа чек-листа для тестирования ПО.
 - Формат и структура чек-листа.
 - Применение интеллект-карт Mind Map для тестирования комплексных приложений.
 - Особенности составления интеллект-карты Mind Map.
 - Практическое задание: создать чек-лист по тестированию главной страницы сайта.
- Создать Mind Map на свободную тему.

1.4. «Составление тестовой документации: тест-кейсы»

- Понятие «тест-кейс». Назначение тест-кейсов при организации процесса тестирования.
- Низкоуровневые тест-кейсы.
- Формат тест-кейса.
- Структура тест-кейса.
- Статусы тест-кейса.
- Особенности применения тест-кейсов.

1.5. «Составление баг-репортов»

- Влияние и стоимость исправления дефектов в ПО.
- Причины возникновения дефектов в коде. Локализация дефектов.
- Общие правила оформления баг-репортов.
- Инструменты для оформления дефекта.
- Атрибуты описания дефекта: приоритет и серьезность дефекта.
- Пример заведения дефекта в инструменте Atlassian Jira.

- Жизненный цикл дефекта (Defect Lifecycle).
- Подходы к работе с дефектами в команде.
- Практическое задание: выполнить тестирование формы регистрации пользователя на соответствие техническому заданию. Найти и оформить дефекты согласно правилам оформления дефектов.

1.6. «Эвристики и мнемоники в тестировании»

- Эвристики и мнемоники: в чем разница?
- Тестирование при помощи эвристик.
- Пять используемых при тестировании ПО мнемоник: SFDPOT, RCRCRC, FEW, HISSUPPS, SPIES, PROOF.
- Особенности использования мнемоники и эвристики в тестировании.
- Практическое задание: продумать одну собственную эвристику и одну мнемонику для удобства работы тестировщика.

1.7. «Метрики и отчеты тестирования»

- Цели и назначение метрик. Метрики в тестировании.
- Метрики процесса тестирования.
- Метрики работы с дефектами.
- Метрики тестового покрытия.
- Метрики планирования.
- Конструирование метрик.
- Отчеты в тестировании ПО. Типы и содержание отчетов.
- Формат и структура отчета о тестировании.
- Формирование отчетов.
- Практическое задание: составить отчет о выполненном тестировании согласно изученным правилам оформления отчета.

Тема 2. «Тест-анализ и техники тест-дизайна»

2.1. «Требования к продукту. Анализ, визуализация и декомпозиция»

- Понятие «техническое задание». Основные элементы техзадания.
- Макеты, их содержание.
- Виды требований. Методы анализа требований.
- Критерии качества документации.
- Метод декомпозиции, его назначение и преимущества.
- Техники декомпозиции: Vague Terms, Conjunctions, Workflow Steps.
- Практическое задание: Составить техническое задание, добавить в техзадание один пункт, который еще не реализован на сайте.

2.2. «Классы эквивалентности»

- Понятия «тест-дизайн», «техники тест-дизайна».
- Пять основных техник тест-дизайна.
- Понятие «классы эквивалентности». Выделение классов эквивалентности.
- Виды классов эквивалентности.
- Особенности применения классов эквивалентности.

- Практическое задание: определить классы эквивалентности, выбрать представителя из каждого класса и описать тесты, которые возможно провести для их проверки.

2.3. «Попарное тестирование (pairwise)»

- Техника тест-дизайна «попарное тестирование», ее назначение.
- Особенности применения попарного тестирования.
- Таблица «pairwise». Способы составления таблицы «pairwise».
- Алгоритм составления таблицы «pairwise» вручную.
- Составление таблицы «pairwise» с помощью генератора PICT.
- Практическое задание: Составить таблицу pairwise вручную для проверки функциональности.

2.4. «Граничные значения и доменный анализ»

- Назначение техники граничных значений в тестировании.
- Применение техники граничных значений.
- Примеры подбора граничных значений для разных задач.
- Построение таблицы доменного анализа.
- Практическое задание: определить сколько всевозможных проверок можно сделать для представленного поля по требованиям.

2.5. «Таблицы принятия решений»

- Применение техники для написания тест-кейсов «таблица принятия решений».
- Алгоритм создания таблицы принятия решений.
- Примеры применения техники «таблица принятия решений» для написания тест-кейсов.
- Практическое задание: составить таблицу принятия решения и тест-кейсы для представленного задания.

2.6. «Переходы состояний»

- Назначение техники переходов состояний.
- Особенности использования техники переходов и состояний при тестировании приложений.
- Тестирование переходов и состояний.
- Назначение диаграммы переходов и состояний. Создание тест-кейсов на основании диаграмм.
- Использование таблицы переходов и состояний. Составление и заполнение таблицы переходов и состояний.
- Практическое задание: составить диаграмму переходов и состояний для процесса оформления заявки на расход в бухгалтерии фирмы на основании предложенного сценария.

2.7. «Другие техники тест-дизайна (причина-следствие, предугадывание ошибок, исследовательское тестирование)»

- Техника тест-дизайна «Причина — следствие». Порядок создания тест-кейсов при помощи этой техники.

- Техника тест-дизайна «Предугадывание ошибок». Ее назначение. Базовый алгоритм использования техники.

- Использование исследовательского тестирования. Решения, позволяющие сделать исследовательское тестирование более структурированным.

- Практическое задание: используя технику «Причина-следствие», продумать причины, которые могут привести к ошибкам. Используя технику «Предугадывание ошибок», предположить, какие ошибки могут возникнуть в процессе, как они могут повлиять на пользователей. Используя технику «Исследовательское тестирование», написать чек-лист проверок для указанных требований.

Тема 3. «Работа с HTML и CSS»

3.1. «Введение в веб-разработку»

- Виды веб-приложений.
- Клиентская и серверная часть веб-приложения.
- Особенности разработки веб-приложений.
- Подходы к построению веб-сайтов: SPA, MPA и PWA.
- Практическое задание: привести по одному примеру приложений для каждого из изученных видов веб-приложений.

3.2. «Основы HTML»

- Базовая структура и основные элементы HTML-документа
- Элементы оформления текста, таблиц, рисунков
- Элементы навигации и оформления ссылок
- Элементы взаимодействия с пользователем.
- Практическое задание: создать простой HTML-документ с предложенными элементами.

3.3. «Основы CSS»

- Язык таблицы стилей CSS (Cascading Style Sheets).
- Основные категории и свойства CSS
- Способы применения стилей к html-элементам.
- Практическое задание: создать файл с расширением css со стилями к HTML-документу и привязать его любым из двух предложенных способов.

3.4. «Разработка веб-приложения»

- Особенности применения HTML и CSS при разработке веб-приложения.
- Примеры применения CSS при разработке веб-приложения
- Практическое задание: создать веб-страницу для публикации статьи по заданным параметрам.

Тема 4. «Тестирование Веб-приложений»

4.1. «Основы тестирования веб-приложений»

- Что такое браузер. Как происходит взаимодействие пользователя с браузером.
- Протокол HTTP, его структура. Коды статуса.
- Виды веб-приложений. Отличия сайта от приложений, PWA-приложений от гибридных и нативных.
- Особенности тестирования web-приложения на desktop.
- Особенности тестирования web-приложения на mobile.
- Практическое задание: протестировать web-приложение свободным поиском (добавление, редактирование и удаление комментариев) и завести найденные баги.

4.2. «DevTools»

- Основные функции Devtools.
- Chrome Devtools.
- Панель разработчика в разных браузерах.
- Практическое задание: на предложенном сайте изменить размер кнопки «Отправить», содержание и цвет кнопки, проследить, не отображаются ли ошибки в консоли.

4.3. «Сбор и анализ трафика»

- Использование инструмента для анализа трафика Charles.
- Инструмент для анализа трафика Fiddler. Конструирование запросов в Fiddler.
- Практическое задание: проснифайте тестовый контур, заменить код ответа, проверить корректно ли обрабатывает фронтенд. Настроить Rewrite для URL.

Тема 5. «Тестирование API»

5.1. «Клиент-серверная архитектура»

- Архитектура современных информационных систем.
- Монолитная или локальная архитектура. Файл-серверная архитектура.
- Клиент-серверная архитектура.
- Способы взаимодействия между системами: через базу данных, через файлы, через api, асинхронный обмен сообщениями.
- Практическое задание: нарисовать пример трехзвенной клиент-серверной архитектуры.

5.2. «REST API»

- Структура http-запроса и http-ответа. Взаимодействие клиента и сервера.
- Метод хранения и передачи структурированной информации JSON.
- REST-архитектура. Методы в архитектуре.
- Способы аутентификации в HTTP-запросах.
- Инструмент для тестирования и конструирования запросов API – POSTMAN.
- Описание переменных, написание автотестов для запросов API, создание мокового сервера для подмены данных в POSTMAN.
- Практическое задание: создать коллекцию в POSTMAN для запросов сайта. Запустить на ней автоматическое прохождение тестов для ожидаемого ответа 500.

5.3. «SOAP API»

- XML, назначение, синтаксис.
- SOAP Протокол в разработке ИС. Структура SOAP-сообщений
- Postman UI. Функции POSTMAN в тестировании UI.
- Практическое задание: отформатировать предложенный json в xml.

5.4. «Техники тест-дизайна при тестировании API и составление тестовой документации»

- Виды документации к API.
- Техники тест-дизайна для тестирования API.
- Составление тест-кейсов и чек листов для тестирования API.
- Конвертирование тест-кейсов в чек-листы.
- Практическое задание: составить 4–6 тест-кейсов для запроса редактирования комментария.

Тема 6. «Тестирование мобильных приложений»

6.1. «Как устроены мобильные приложения»

- Основные особенности мобильного тестирования. Типы мобильных приложений. Архитектура клиент-серверного мобильного приложения.
- Различия между Android и iOS.
- Особенности установки приложений на разных платформах.
- Практическое задание: подобрать список из 5–15 мобильных устройств, на которых рекомендовал бы команде QA протестировать приложение. Обосновать выбор.

6.2. «Особенности тестирования мобильных приложений»

- Специфика тестирования мобильных приложений: прерывания, пуши, работа с сетью, GPS, ориентация
- Отличие тестирования мобильных приложений от веб.
- Практическое задание: провести тестирование приложения исходя из полученных знаний.

6.3. «Тестирование мобильных приложений Android»

- Жизненный цикл Activity Android-приложения.
- Оболочки вендора (vendor skins) в Android.
- Версионирование операционной системы Android. Тестовые девайсы, покрывающие основные потребности и физические характеристики устройств.
- Режим разработчика в ОС Android, его возможности для тестирования и разработки приложений.
- Практическое задание: тестирование Android-приложения.

6.4. «Среда разработки Android Studio и Android Debug Bridge (ADB)»

- Системные логи для Android

- Android Studio для тестирования мобильных приложений.
- Android Debug Bridge в разработке Android-приложений, основные команды приложения.
- Практическое задание: проанализировать системные логи на устройстве Android, проанализировать логи с помощью Android Studio. Подключить устройства к ADB.

6.5. «Тестирование мобильных приложений iOS»

- Жизненный цикл ViewController iOS-приложения.
- Особенности тестирования мобильных приложений на iOS.
- Работа с Xcode и сбор логов с IOS-приложения.
- Практическое задание: протестировать приложение на iOS.

6.6. «Сбор и анализ трафика»

- Использование Charles Proxy анализа трафика мобильных приложений.
- Практическое задание: тестирование тестового приложения с помощью Charles Proxy.

Тема 7. «Базовый SQL»

7.1. Введение в SQL

- Введение в SQL, его роль в управлении базами данных.
- Запросы и основные операторы SQL.
- Виды баз данных. Реляционные базы данных.
- Таблицы, типы данных и связи в реляционных базах данных.
- Практическое задание: создайте с помощью SQL-запроса три таблицы:

1. **Таблица «Клиенты»:**
Поля: ID, Имя, Год рождения, Город
2. **Таблица «Услуги»:**
Поля: ID, Название, Стоимость
3. **Таблица «Заказы»:**
Поля: ID, Дата, Клиент, Услуга

Определите тип данных для каждого поля и настройте при необходимости ограничения по внешнему ключу.

7.2. Главный SQL запрос: оператор SELECT

- Конструкция SELECT-запросов.
 - Операторы выборки и агрегации данных.
 - Операторы SELECT FROM.
 - Условия фильтрации в SELECT-запросе.
 - Сортировка данных ORDER BY.
 - Практическое задание: пройдите это задание в тренажере.
1. Допустим у нас есть таблица товаров с колонками «products_id», «product_name» и «product_price». Создайте запрос на выборку всех товаров, дороже 10000.

2. Допустим у нас есть таблица товаров с колонками «products_id», «product_name» и «product_price». Создайте запрос на выборку всех продуктов с ценой больше 1000 и отсортируйте результат по возрастанию: от самого дешевого до самого дорогого.

7.3. Объединение данных: оператор JOIN

- Оператор присоединения таблиц JOIN.
- Режим OUTER JOIN.
- Режим LEFT OUTER JOIN.
- Режим RIGHT OUTER JOIN.
- Режим INNER JOIN.
- Использование JOIN в SELECT-запросах.
- Объединение таблицы с самой собой. Комбинаторное объединение.
- Практическое задание: Пройдите задание в тренажере. Придумайте два своих примера с использованием различных операторов JOIN и опишите эти запросы.

7.4. Группировка данных

- Функции агрегации для обработки данных.
- Группировка данных оператором GROUP BY.
- Фильтрация групп оператором HAVING.
- Практическое задание: три задания на тренажере.

7.5. Модификация данных: INSERT, UPDATE, DELETE

- Добавление данных в таблицу оператором INSERT.
- Изменение данных таблицы оператором UPDATE.
- Удаление данных из таблицы оператором DELETE. Меры предосторожности при удалении данных.
- Практическое задание: три задания на тренажере.

Тема 8. «Командная строка»

8.1. «Базовые команды в bash: навигация»

- Перемещение по директориям. Команды Bash для перемещения по директориям.
- Просмотр содержимого директории.
- Просмотр текущей активной директории.
- Практическое задание: в инструменте для работы с Bash повторить команды из параграфа «Bash — просмотр содержимого директории».

8.2. «Директории: создание, просмотр и редактирование директорий»

- Просмотр директории.
- Редактирование директории.
- Копирование директории.
- Практическое задание: создать несколько директорий, скопировать, удалить при помощи Bash. Тоже самое сделать при помощи графического интерфейса операционной системы.

8.3. «Файлы: создание, просмотр, редактирование и копирование»

- Просмотр файлов.
- Редактирование файлов.
- Копирование файлов.
- Практическое задание: создать несколько директорий, скопировать, удалить при помощи Bash. То же самое сделать при помощи графического интерфейса операционной системы.

8.4. «Поиск по файлам и директориям»

- Команда find для поиска файлов и директорий.
- Примеры использования команды find с разными параметрами.
- Практическое задание: в инструменте для работы с Bash повторить все команды из параграфа «Команда find».

8.5. «Ввод и вывод»

- Bash – добавление данных в файл, команды для добавления данных.
- Вывод данных из файла с помощью команд Bash.
- Практическое задание: в инструменте для работы с Bash повторить все команды из параграфа «Добавление данных в файл через команду».

8.6. «Работа с процессами»

- Что такое процесс. Два основных вида процессов: системные и пользовательские.
- Как происходит управление процессами.
- Основные команды для управления процессами.
- Практическое задание: в инструменте для работы с Bash повторить все команды из параграфа «Основные команды для управления процессами».

8.7. «Удаленные ресурсы»

- Что такое SSH. Подключение по SSH.
- Putty - программа для подключения по SSH к удаленному компьютеру.
- Работа на удаленном ресурсе.
- Практическое задание: найти информацию об аналогах Putty. Провести сравнение SSH-клиентов и составить сравнительную таблицу.

Тема 9. «Git»

9.1. «Что такое Git: установка и настройка»

- Система управления версиями Git, выполняющая распределенную модель работы с репозиторием. Установка Git
- Системы управления репозиториями кода: GitHub, GitLab, GitFlic.
- Создание и клонирование репозитория.
- Использование SSH-ключей.
- Практическое задание: создать профиль в одной из трех систем: GitHub, GitFlic, GitLab. Сгенерировать ssh-ключи, добавить открытый ключ в две системы. В двух системах создать один удаленный репозиторий, клонировать его в локальное рабочее

окружение. В локальном рабочем окружении создать один локальный репозиторий, клонировать его в две удаленные системы.

9.2. «Основные команды Git»

- Работа с файлами
- Просмотр состояния репозитория
- Работа с локальными репозиториями
- Работа с удалёнными репозиториями
- Работа с тегами
- Практическое задание: выполнить операции с файлами в локальном репозитории.

9.3. «Ветки»

- Ветки
- Слияние веток
- Перебазирование ветки.
- Практическое задание: в существующем репозитории создать три ветки, выполнить операции с файлами веток.

9.4. «UI Tools в Git»

- Приложение Git GUI: 1. Создание репозитория. 2. Клонирование репозитория из удаленной платформы. 3. Создание ветки. 4. Фиксация изменений. 5. Отображение состояния репозитория. 6. Отмена изменений. 7. Слияние ветки. 8. Разбор конфликтов слияния. 9. Удаление ветки. 10. Получение изменений из удалённого репозитория. 11. Отправка изменений на удалённый репозиторий. 12. Создание тегов. 13. История фиксаций. 14. SSH ключи.
- Git Extensions: 1. Создание репозитория. 2. Клонирование репозитория из удаленной платформы. 3. Создание ветки. 4. Фиксация изменений. 5. Отображение состояния репозитория. 6. Отмена изменений. 7. Слияние ветки. 8. Разбор конфликтов слияния. 9. Удаление ветки. 10. Получение изменений из удалённого репозитория. 11. Отправка изменений на удалённый репозиторий. 12. Создание тегов. 13. История фиксаций. 14. SSH ключи.
- TortoiseGit: 1. Создание репозитория. 2. Клонирование репозитория из удаленной платформы. 3. Создание ветки. 4. Фиксация изменений. 5. Отображение состояния репозитория. 6. Отмена изменений. 7. Слияние ветки. 8. Разбор конфликтов слияния. 9. Удаление ветки. 10. Получение изменений из удалённого репозитория. 11. Отправка изменений на удалённый репозиторий. 12. Создание тегов. 13. История фиксаций. 14. SSH ключи.
- GitKraken: 1. Создание репозитория. 2. Клонирование репозитория из удаленной платформы. 3. Создание ветки. 4. Фиксация изменений. 5. Отображение состояния репозитория. 6. Отмена изменений. 7. Слияние ветки. 8. Разбор конфликтов слияния. 9. Удаление ветки. 10. Получение изменений из удалённого репозитория. 11. Отправка изменений на удалённый репозиторий. 12. Создание тегов. 13. История фиксаций. 14. SSH ключи.
- SourceTree: 1. Создание репозитория. 2. Клонирование репозитория из удаленной платформы. 3. Создание ветки. 4. Фиксация изменений. 5. Отображение состояния

репозитория. 6. Отмена изменений. 7. Слияние ветки. 8. Разбор конфликтов слияния. 9. Удаление ветки. 10. Получение изменений из удалённого репозитория. 11. Отправка изменений на удалённый репозиторий. 12. Создание тегов. 13. История фиксаций. 14. SSH ключи.

- **InteliJi IDEA, PyCharm:** 1. Создание репозитория. 2. Клонирование репозитория из удаленной платформы. 3. Создание ветки. 4. Фиксация изменений. 5. Отображение состояния репозитория. 6. Отмена изменений. 7. Слияние ветки. 8. Разбор конфликтов слияния. 9. Удаление ветки. 10. Получение изменений из удалённого репозитория. 11. Отправка изменений на удалённый репозиторий. 12. Создание тегов. 13. История фиксаций
- **Visual Studio Code:** 1. Создание репозитория. 2. Клонирование репозитория из удаленной платформы. 3. Создание ветки. 4. Фиксация изменений. 5. Отображение состояния репозитория. 6. Отмена изменений. 7. Слияние ветки. 8. Разбор конфликтов слияния. 9. Удаление ветки. 10. Получение изменений из удалённого репозитория. 11. Отправка изменений на удалённый репозиторий. 12. Создание тегов. 13. История фиксаций
- **Практическое задание:**
Выберите любой из разобранных клиентов для работы с репозиториум и выполните с его помощью следующие операции:
Клонируйте репозиторий из удаленной платформы
Создайте ветку
Внесите изменения в произвольный файл
Зафиксируйте изменения
Влейте ветку в основную ветку
Пометьте фиксацию слияния новым тегом
Получите изменения из удалённого репозитория
Отправьте зафиксированные изменения на удалённый репозиторий

9.5. «Командная работа в Git»

- Создание Pull Requesta.
- Проведение Code Review.
- Модель ветвления при работе с репозиториум GitFlow.
- Альтернативная модель командной работы с репозиториум Trunk Based Development.
- Подход GitHub Flow к организации командной работы с репозиториум.
- Практическое задание: Выполнить работу согласно алгоритму работы по GitFlow. В ходе работы ветки не удалять.

9.6. «Использование Git для решения рабочих задач тестировщика»

- Git для функционального тестировщика
- Git для автоматизатора и нагрузочника
- Git для тимлида
- Практическое задание. Представьте, что вы тимлид команды, которая работает с репозиториум по Trunk Based Development. Одна из ваших задач — это контроль соблюдения командой подхода работы с репозиториум по Trunk Based Development. Составьте чек-лист необходимых проверок для выполнения контроля корректности

работы команды репозиториум по подходу Trunk Based Development. Аргументируйте необходимость применения каждой проверки: для чего она нужна и что будет, если ее не сделать.

3.3 Календарный учебный график

Календарный график обучения является примерным, составляется и утверждается для каждой группы.

Срок освоения программы — 15 недель. Начало обучения — по мере набора группы. Примерный режим занятий: не более 8 академических часов в день, до 40 часов в неделю. Промежуточная и итоговые аттестации проводятся согласно графику.

Примерный график освоения программы:

№	Темы / недели	В Р	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.	Основы тестирования	РП	7														
		СР	3	11													
2.	Тест-анализ и техники тест-дизайна	РП			7												
		СР			3	11											
3.	Работа с HTML и CSS	РП					4										
		СР					8										
4.	Тестирование Веб	РП						3									
		СР						6									
5.	Тестирование API	РП							4								
		СР							8								
6.	Тестирование мобильных приложений	РП								6							
		СР								4	8						
7.	Базовый SQL	РП										5					
		СР										10					
8.	Командная строка	РП											7				
		СР											3	11			
9.	Git	РП													10		
		СР															11
10.	Итоговая аттестация	РП															2

4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ - ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

4.1. Требования к квалификации педагогических кадров, представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса

Требования к образованию и обучению лица, занимающего должность преподавателя: высшее образование — специалитет или магистратура, направленность (профиль) которого, как правило, соответствует преподаваемому учебному курсу, дисциплине (модулю).

Дополнительное профессиональное образование — профессиональная переподготовка, направленность (профиль) которой соответствует преподаваемому учебному курсу, дисциплине (модулю).

Педагогические работники обязаны проходить в установленном законодательством Российской Федерации порядке обучение и проверку знаний и навыков в области охраны труда.

Рекомендуется обучение по дополнительным профессиональным программам по профилю педагогической деятельности не реже чем один раз в три года.

Требования к опыту практической работы: при несоответствии направленности (профиля) образования преподаваемому учебному курсу, дисциплине (модулю) — опыт работы в области профессиональной деятельности, осваиваемой слушателями или соответствующей преподаваемому учебному курсу, дисциплине (модулю).

Преподаватель: стаж работы в образовательной организации не менее одного года; при наличии ученой степени (звания) — без предъявления требований к стажу работы.

Особые условия допуска к работе: отсутствие ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации

Прохождение обязательных предварительных (при поступлении на работу) и периодических медицинских осмотров (обследований), а также внеочередных медицинских осмотров (обследований) в порядке, установленном законодательством Российской Федерации

Прохождение в установленном законодательством Российской Федерации порядке аттестации на соответствие занимаемой должности.

4.2. Требования к материально-техническим условиям

Организация проводит занятия по адресу: г. Москва, ул. Суцеский Вал, д. 18. Аудитории для занятий расположены на 11-м этаже здания.

Все занимаемые помещения соответствуют обязательным нормам пожарной безопасности и требованиям санитарно-эпидемиологических служб. Помещения имеют централизованные системы водоснабжения, отопления и канализации. Воздухообмен помещений обеспечивается современными системами кондиционирования, за счет приточно-вытяжной вентиляционной системы.

Учебным центром СКБ Контур заключен договор с организацией общественного питания о возможности обеспечения слушателей питанием.

В учебной аудитории проводятся лекции и практические занятия. Аудитория оснащена столами и стульями, в составе учебного оснащения маркерная доска и флипчарт, в случае необходимости подключается мультимедийный проектор, слушателям предоставляются компьютеры.

Компьютерная сеть учебного центра оснащена необходимым оборудованием для доступа в интернет по выделенному каналу. На каждом компьютере обеспечен постоянный доступ к компьютерной программе «Контур.Школа».

Для проведения вебинаров и онлайн-трансляций используется оснащенная современным оборудованием видеостудия:

- помещение оборудовано посадочными местами для спикера(ов);
- спикеру предоставляется персональный компьютер с соответствующими мультимедийными характеристиками (Intel Core i3 либо идентичные по характеристикам, оперативная память: от 4 Гб и выше для всех ОС), со стабильным соединением с сетью Интернет на скорости не менее 1 Мбит/с;
- видеочасть (максимальное разрешение видео — не менее 3840 x 2160).

Размещение материалов вебинаров и доступ к ним участников обеспечивает техническая платформа (сайт, система управления сайтом, другие технические средства):

1. Трансляция вебинара в режиме реального времени.
2. Хранение, систематизация записей вебинаров, с предоставлением участникам возможности просмотра записи онлайн.
3. Хранение, систематизация и доступ к скачиванию материалов учебных программ.
4. Напоминание участникам о предстоящем вебинаре за 1 час до начала мероприятия.
5. Использование защищенных соединений, передача и прием видео и звука по протоколам RTMP(S) или аналогичным.
6. Управление качеством и разрешением передаваемого/принимаемого видео вплоть до разрешения HD 720p на каждого участника мероприятия (адаптивный стриминг).
7. Обмен короткими текстовыми сообщениями (чат).
8. Осуществление записи мероприятий в формате, не требующем конвертации для проигрывания (mp4, AVI, WMA и т.д.).
9. Система регистрации на вебинар.
10. Техническое сопровождение проведения вебинара.
11. Отображение числа участников.
12. Техническая доступность услуги не менее 99,8% времени.
13. Устойчивость при проведении вебинара при одновременном подключении до 3000 участников.
14. Возможность участия пользователей на вебинарах в браузерах Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari с установленным плагином Adobe Flash Player.
15. Передача аудио- и видеосообщения на персональные компьютеры участников реализована при скорости интернет-соединения не менее 134 кбит/с.

Основные функции программы Контур.Школа:

1. Размещение расписания и описания учебных программ и условий обучения.
2. Онлайн-трансляция учебных занятий с возможностью обратной связи.
3. Размещение тестов и проведение онлайн-тестирования.
4. Размещение и выбор образовательного контента и заданий для слушателей.
5. Хранение учебно-методических материалов.
6. Обратная связь слушателей к организаторам и преподавателям.
7. Автоматическая фиксация хода учебного процесса, промежуточных и итоговых результатов слушателей.
8. Хранение информации о ходе учебного процесса и результатов обучения в течение периода обучения.
9. Сбор и хранение заявок на обучение и сведений о слушателях.
10. Создание и актуализация контента и учебно-методических материалов.
11. Информационно-консультационное обслуживание слушателей.

4.3. Информационные и учебно-методические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Дополнительная общеобразовательная программа – дополнительная общеразвивающая программа обеспечивается учебно-методическими материалами по всем модулям образовательной программы.

Фонд учебно-научной библиотеки содержит основную и дополнительную учебную, учебно-методическую, научную литературу, справочно-библиографические и периодические издания (в том числе и на электронных носителях) по всем темам и дисциплинам реализуемой программы.

Нормативно-правовая база

1. Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 №149-ФЗ
2. ГОСТ Р 56922-2016/ISO/IEC/IEEE 29119-3:2013 «Системная и программная инженерия. Тестирование программного обеспечения.», часть 3 «Документация тестирования»

Список литературы

1. HTML и CSS. Разработка и дизайн веб-сайтов [Дакетт Джон]; Эксмо; 2022 г. – 480с.
2. Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков Учебное пособие / [Кириченко А.В.]; Наука и техника; 2021 г. – 432 с.
3. Web-программирование на JavaScript. Учебное пособие для СПО/ [Диков А.В.]; Лань; 2021 г. – 165 с.
4. Искусство тестирования программ / [Сандлер, Майерс, Баджетт]; Диалектика; 2020 г. - 272 с.
5. Основы тестирования программного обеспечения. Учебное пособие / [Старолетов Сергей Михайлович]; Лань; 2022 г. - 382 с.
6. Основы тестирования и верификации программного обеспечения. Учебное пособие / [Старолетов Сергей Михайлович]; Лань; 2023 г. - 344 с.
7. Основы технологий баз данных. Учебное пособие / [Новиков Б. А., Горшкова Е. А.]; ДМК Пресс; 2020 г. – 582 с.
8. Проектирование и разработка web-приложений. Введение в frontend и backend разработку на JavaScript и node.js: учебное пособие для СПО [Зяц А., Васильев Н.]; Лань; 2023 г. – 120 с.

9. Разработка требований к программному обеспечению. 3-е изд., дополненное / [Вигерс К., Битти Дж.]; БХВ; 2023 г. - 736 с.
10. Тестирование программного обеспечения. Базовый курс/[Святослав Куликов]; ЕРАМ Systems, 2022 г.
11. Тест-дизайн. Практическое руководство для начинающих/ [Назина Ольга]; БХВ; 2023 г. – 512 с.
12. Что такое тестирование. Курс молодого бойца / [Назина Ольга]; БХВ; 2022г. – 592 с.
13. Эффективное тестирование программного обеспечения / [Маурисио Аниче]; ДМК-Пресс; 2023 г. - 370 с.

Периодические издания

1. Журнал «Вестник компьютерных и информационных технологий», №2, 2023г.
<http://www.vkit.ru/index.php/archive-rus/1228-02-february>
2. Научно-практический журнал «Программные продукты и системы» №1, 2023г.
<http://www.swsys.ru/index.php>

Интернет-ресурсы

1. Техническая документация по SQLServer от Microsoft <https://docs.microsoft.com/ru-ru/sql/sql-server/?view=sql-server-ver15>
2. Упражнение по отработке навыков пользования SQL <https://www.sql-ex.ru/>

4.4. Общие требования к организации образовательного процесса

При реализации дополнительной общеобразовательной программы – дополнительной общеразвивающей программы используется лекционно-семинарская система, практические занятия, индивидуальные и групповые консультации, мастер-классы и другие интерактивные формы обучения. Интегративно-дифференцированная организация занятий в процессе подготовки слушателей позволяет не только адаптировать образовательный процесс к индивидуальным особенностям и интересам слушателей, но и обеспечить свободу выбора и вариативность образования, сформировать у слушателей стремление к самообразованию, способствовать решению индивидуальных задач развития личности.

Особую значимость на уровне технологии обучения имеет организационно-педагогическое условие, предполагающее организацию самостоятельной работы слушателей как средство формирования профессиональных компетенций. Самостоятельная работа слушателей является обязательной составляющей образовательного процесса подготовки слушателей дополнительной общеобразовательной программы – дополнительной общеразвивающей программы.

Применение дистанционных образовательных технологий и электронного обучения

Дистанционные образовательные технологии применяются частично.

В учебном процессе могут использоваться следующие организационные формы учебной деятельности:

- обзорные (установочные) лекции с использованием дистанционных образовательных технологий;
- самостоятельная работа с СДО: работа с электронным учебником, просмотр видеолекций и др.;
- самостоятельная работа с программами контроля знаний (тестами);
- консультации (индивидуальные с применением электронных средств, групповые и предэкзаменационные);
- семинары с использованием дистанционных образовательных технологий;
- коллоквиумы;
- итоговые аттестационные работы.

5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

5.1. Формы контроля знаний и требования к его проведению

Для проведения промежуточной и итоговой аттестации программы разработан фонд оценочных средств по программе, являющийся неотъемлемой частью учебно-методического комплекса.

Объектами оценивания выступают:

- степень освоения теоретических знаний;
- уровень овладения практическими умениями и навыками по всем видам учебной работы, активность на занятиях.

Текущий контроль знаний слушателей проводится преподавателем, ведущим занятия в учебной группе, на протяжении всего обучения по программе.

Текущий контроль знаний включает в себя наблюдение преподавателя за учебной работой слушателей и проверку качества знаний, умений и навыков, которыми они овладели на определенном этапе обучения посредством выполнения упражнений на практических занятиях и в иных формах, установленных преподавателем.

Промежуточная аттестация — оценка качества усвоения слушателями содержания учебных блоков непосредственно по завершении их освоения, проводимая в форме зачета посредством тестирования или в иных формах, в соответствии с учебным планом и учебно-тематическим планом.

Итоговая аттестация — процедура, проводимая с целью установления уровня знаний, слушателей с учетом прогнозируемых результатов обучения и требований к результатам освоения образовательной программы. Итоговая аттестация слушателей осуществляется в форме зачета посредством тестирования.

Слушатель допускается к итоговой аттестации после изучения тем образовательной программы в объеме, предусмотренном для лекционных и практических занятий.

Слушателям, освоившим дополнительную общеобразовательную программу - дополнительную общеразвивающую программу по теме «Тестирование ПО. Базовый уровень», и успешно прошедшим итоговую аттестацию, выдается свидетельство установленного образца с указанием названия программы, календарного периода обучения, длительности обучения в академических часах.

Для аттестации слушателей на соответствие их персональных достижений требованиям соответствующей ОП созданы фонды оценочных средств, включающие типовые задания, тесты и методы контроля, позволяющие оценить знания, умения и уровень приобретенных компетенций.

Фонды оценочных средств соответствуют целям и задачам программы подготовки специалиста, учебному плану и обеспечивают оценку качества общепрофессиональных и профессиональных компетенций, приобретаемых слушателями.

5.2. Критерии оценки знаний слушателей

Предмет оценивания	Объект оценивания (навыки)	Показатель оценки (знания, умения)
--------------------	----------------------------	------------------------------------

(компетенции)		
<p>Основы тестирования</p>	<p>— Понимание основ тестирования ПО — Проведение тестирования</p>	<p>Знания:</p> <ul style="list-style-type: none"> • Понятия и цели тестирования программного обеспечения. • Основные подходы (техники) проведения тест-анализа. • Основы применения методов тестирования «тест-анализ», «тест-дизайн». • Особенности составления тестовой документации: чек-листов, интеллект карт, тест-кейсов, баг-репортов. • Подходы к работе с дефектами в ПО, инструменты для оформления дефектов. • Цели и назначение метрик в тестировании. • Типы и содержание отчетов о тестировании. • Особенности формирования отчетов о тестировании. <p>Умения:</p> <ul style="list-style-type: none"> • Разрабатывать и использовать чек-листы, интеллект карты Mind Map, тест-кейсы для тестирования ПО. • Описывать дефекты, оформлять баг-репорты. • Подбирать метрики и создавать отчеты о тестировании.
<p>Применение при тестировании ПО различных техник тест-анализа и тест-дизайна</p>	<p>Понимание технологии составления технического задания Применение различных техник</p>	<p>Знания:</p> <ul style="list-style-type: none"> • Основные элементы техзадания. • Критерии качества составления документации. • Способы составления таблицы «pairwise» для проведения попарного тестирования ПО. • Назначение и применение техники граничных значений в тестировании. • Особенности техники «таблица принятия решений» для написания тест-кейсов. • Назначение и особенности применения техники переходов состояний. <p>Умения:</p> <ul style="list-style-type: none"> • Применять техники декомпозиции: Vague Terms, Conjunctions, Workflow Steps. • Создавать таблицы «pairwise» вручную и с помощью генератора. • Строить и применять таблицы доменного анализа. • Применять технику «таблица принятия решений» для написания тест-кейсов. • Создавать тест-кейсы на основании техники переходов и состояний.

Работа на языках программирования HTML и CSS	- Применение знаний по основам языков программирования для выполнения разных задач при тестировании программного обеспечения	<p>Знания:</p> <ul style="list-style-type: none"> • Основы языков программирования HTML и CSS. • Базовые концепции программирования, их применение при тестировании ПО. <p>Умения:</p> <ul style="list-style-type: none"> • Читать код, созданный на языке программирования HTML и CSS. • Соблюдать минимальные стандарты оформления кода.
Проведение тестирования веб-приложений	- Понимание основ тестирования веб-приложений - Проведение проверки веб-приложений на наличие уязвимости	<p>Знания:</p> <ul style="list-style-type: none"> • Особенности тестирования web-приложения. • Основные функции Devtools. • Особенности тестирования web-приложения на mobile. • Функционал и возможности инструментов для анализа трафика Charles, Fiddler. <p>Умение:</p> <ul style="list-style-type: none"> • Тестировать web-приложения на desktop. • Тестировать веб-приложения на смартфонах. • Открывать и использовать панель разработчика Devtools в разных браузерах. • Использовать инструменты для анализа трафика Charles, Fiddler.
Тестирование API	Проведение тестирования API.	<p>Знания:</p> <ul style="list-style-type: none"> • Особенности архитектуры современных информационных систем. • Особенности взаимодействия клиента и сервера. • Структура http-запроса и http-ответа. • XML, назначение, синтаксис. • SOAP Протокол в разработке ИС. <p>Умения:</p> <ul style="list-style-type: none"> • Тестировать web-приложения на desktop • Описывать переменные, писать автотесты для запросов API, создавать моковый сервер для подмены данных в POSTMAN. • Использовать POSTMAN в тестировании UI. • Писать тест-кейсы и чек листы для тестирования API. • Конвертировать тест-кейсы в чек-листы.
Тестирование мобильных приложений	- Выбор и применение инструментов тестирования мобильных приложений.	<p>Знания:</p> <ul style="list-style-type: none"> • Особенности и инструменты мобильного тестирования. • Архитектура клиент-серверного мобильного приложения.

		<ul style="list-style-type: none"> • Особенности тестирования мобильных приложений на Android и iOS. <p>Умения:</p> <ul style="list-style-type: none"> • Тестировать мобильные приложения на Android и iOS. • Использовать Android Studio, Android Debug Bridge для тестирования мобильных приложений. • Работать в Charles Proxu для анализа трафика мобильных приложений.
Работа с базовым SQL	Применение знаний по основам работы с SQL.	<p>Знания:</p> <ul style="list-style-type: none"> • Основные особенности и свойства базы данных. • Структуру базы данных, типы данных в SQL. • Операторы выборки и агрегации данных. • Операторы для работы с данными. • Базовые операторы для присоединения таблиц JOIN в SQL. <p>Умения:</p> <ul style="list-style-type: none"> • Составлять связи между таблицами базы данных. • Составлять запросы с операторами выборки и агрегации. • Объединять таблицы с помощью операторов для присоединения таблиц JOIN.
Работа с командной строкой	Создание, просмотр и редактирование директорий и файлов. Работа с процессами.	<p>Знания:</p> <ul style="list-style-type: none"> • Основные особенности перемещения по директориям. • Команды Bash для перемещения по директориям. • Команды Bash для добавления и вывода данных из файла. • Основы работы на удаленном ресурсе через подключения по SSH. <p>Умения:</p> <ul style="list-style-type: none"> • Осуществлять создание, просмотр, редактирование и копирование файлов. • Проводить поиск по файлам и директориям используя команды find. • Добавлять и выводить данные из файла используя команды Bash. • Выполнять подключения по SSH к удаленному компьютеру.
Навыки работы с Git	Применение знаний по работе с Git.	<p>Знания:</p> <ul style="list-style-type: none"> • Основы применения SSH-ключей и работы по SSH-протоколу. • Особенности работы в системе управления версиями Git. • Основные команды Git.

		<ul style="list-style-type: none"> • Особенности командной работы в Git. <p>Умения:</p> <ul style="list-style-type: none"> • Использовать SSH-ключи и работать по SSH-протоколу. • Создавать пары ключей: закрытый и открытый. • Добавлять в удаленную систему открытый ключ для доступа к системе без пароля. • Клонировать удаленный репозиторий в локальное рабочее окружение по протоколу SSH. • Работать с файлами, локальными и удалёнными репозиториями, тегами. • Создавать ветки, передавать локальную ветку на удалённый сервер, выполнять слияние веток. • Выполнять командную работу в Git. • Использовать Git для решения рабочих задач тестировщика.
--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Критерии оценки аттестации – зачета:

1. Оценка «**Зачтено**» выставляется слушателю, продемонстрировавшему твердое и всесторонние знания материалы, умение применять полученные в рамках занятий практические навыки и умения. Достижения за период обучения и результаты текущей аттестации демонстрировали отличный уровень знаний и умений слушателей. Не менее 80% правильных ответов при решении тестов.

2. Оценка «**Не зачтено**» выставляется слушателю, который в недостаточной мере овладел теоретическим материалом по дисциплине, допустил ряд грубых ошибок при выполнении практических заданий, а также не выполнил требований, предъявляемых к промежуточной аттестации. Достижения за период обучения и результаты текущей аттестации демонстрировали неудовлетворительный уровень знаний и умений слушателя. Менее 80% правильных ответов при решении тестов.

5.3. Фонд оценочных средств

Тест к уроку 1.1. «Введение в тестирование ПО»

1. QA – это
 1. Обеспечение качества (в него входит тестирование как процесс и контроль качества продукта)
 2. Контроль качества
 3. Оценка требований
2. Что из этого НЕ ЯВЛЯЕТСЯ целью тестирования
 1. Проверить, все ли указанные требования выполнены
 2. Предотвращение дефектов
 3. Обеспечение качества ПО для конечных пользователей
 4. Сокращение сроков проекта
3. Что из этого НЕ ОТНОСИТСЯ к артефактам тестировщика (документ, который создает непосредственно тестировщик)
 1. Тест-кейс

2. Чек-лист
3. Требования к продукту
4. Отчет по тестированию
4. Проверка графического дизайна главной страницы сайта макету – это
 1. Функциональное тестирование
 2. Тестирование интерфейса
 3. Тестирование удобства
 4. Тестирование масштабируемости
5. Если тестировщик при работе использует инструменты перехвата трафика, Devtools, просматривает запросы в БД, логи, то он проводит тестирование методом
 1. Белого ящика
 2. Серого ящика
 3. Черного ящика

Тест к уроку 1.2. «Тест-анализ и тест-дизайн: знакомство»

1. Что из этого пример функциональных требований
 1. Отчет №38 должен формироваться за 5 минут
 2. На форме заявки на кредит у каждого поля ввода должна быть всплывающая подсказка
 3. Интерфейс страницы Личного кабинета соответствует макету №4
2. Что из этого НЕ ЯВЛЯЕТСЯ атрибутом требований
 1. непротиворечивость
 2. однозначность
 3. полнота
 4. постоянство
3. Когда для тестировщика эффективна оценка рисков
 1. На этапе тест-анализа
 2. На этапе тестирования
 3. В конце проекта
4. Что такое ментальная карта
 1. График реализации функций продукта
 2. Список найденных дефектов
 3. Представление объекта в виде диаграммы (декомпозиция)
 4. Карта психических отклонений человека
5. Проектирование и создание тест-кейсов – это этап
 1. Тест-аналитики
 2. Тест-дизайна
 3. Планирования проекта

Тест к уроку 1.3. «Составление тестовой документации: чек-листы и Mind Map»

1. Что такое чек-лист?
 1. Алгоритм действий, которые требуется совершить для проверки работы программы
 2. Графическое представление идей и концепций
 3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 4. Отчет об ошибке
2. Что из перечисленного является назначением чек-листа?
 1. Стандартизация
 2. Уменьшение общего числа ошибок
 3. Возможность делегирования
 4. Все вышеперечисленное

3. Какие статусы используются в отметке о выполнении тестов чек-листа?
 1. пройден, провален, пропущен, разблокирован
 2. пройден, провален, пропущен, заблокирован
 3. пройден, провален
 4. запланирован, в работе, выполнен
4. Что такое Mind Map?
 1. Алгоритм действий, которые требуется совершить для проверки работы программы
 2. Графическое представление идей и концепций
 3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 4. Отчет об ошибке

Тест к уроку 1.4. «Составление тестовой документации: тест-кейсы»

1. Что такое тест-кейс?
 1. Алгоритм действий, которые требуется совершить для проверки работы программы
 2. Графическое представление идей и концепций
 3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 4. Артефакт, который описывает совокупность конкретных шагов, условий и параметров, необходимый для проверки реализации тестируемой функциональности или ее части.
2. Для чего не используются тест-кейсы?
 1. Структуризация и систематизация подхода к тестированию
 2. Вычисление метрик тестового покрытия
 3. Отчитаться об ошибке, чтоб разработчики начали ее исправлять
 4. Прозрачность общения между стейкхолдерами
3. Какие атрибуты являются обязательными для тест-кейса?
 1. Предусловия, постусловия, приоритет
 2. Идентификатор, название, шаги и ожидаемый результат
 3. Идентификатор, название, шаги и фактический результат
 4. Идентификатор, предусловия, название, шаги, тестовые данные, комментарий и ожидаемый результат
4. Что такое фактический результат?
 1. Описание того, что должно произойти после каждого шага
 2. Данные, необходимые для выполнения теста
 3. Что произошло на самом деле при выполнении каждого шага кейса
 4. Отчет об ошибке
5. Какие статусы могут быть у тест-кейса?
 1. Пройден, провален, известная ошибка, сломан, заблокирован, пропущен
 2. Пройден, провален, пропущен, заблокирован
 3. Статусом тест-кейса является фактический результат
 4. Запланирован, в работе, выполнен

Тест к уроку 1.5. «Составление баг-репортов»

1. Выберите наиболее корректный заголовок
 1. В корзине возникает ошибка NullPointerException при попытке оформления заказа
 2. Невозможно оформить заказ пользователя
 3. Ошибка NullPointerException при попытке оформить заказ

4. В корзине возникает сообщение NullPointerException при нажатии на кнопку «Оформить заказ»
 2. Выберите невероятную причину дефекта?
 1. Неполные требования к программному продукту
 2. Изменения во взаимосвязанных библиотеках или сервисах
 3. Изменение ожиданий пользователя от программного продукта
 4. Недостаточная компетентность разработчика
 3. Как называется процесс сбора и уточнения сведений о возникновении дефекта?
 1. Подтверждение
 2. Поиск
 3. Локализация
 4. Оформление
 4. Кто обычно определяет приоритет исправления дефекта?
 1. Разработчик
 2. Руководитель проекта \ Владелец продукта
 3. Тестирующий
 4. Пользователь
 5. Без какого атрибута нельзя оформить дефект?
 1. Скриншот и стектрейс
 2. Заголовок
 3. Шаги воспроизведения
 4. Версия программного продукта
- Тест к уроку 1.6. «Эвристики и мнемоники в тестировании»
1. Что такое эвристика?
 1. Графическое представление идей и концепций
 2. Использование предыдущего опыта и информации о вероятности наступления различных событий
 3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 4. Совокупность правил и приемов, которые помогают запоминать нужные сведения и данные
 2. Что такое мнемоника?
 1. Использование предыдущего опыта и информации о вероятности наступления различных событий
 2. Графическое представление идей и концепций
 3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 4. Совокупность правил и приемов, которые помогают запоминать нужные сведения и данные
 3. Для чего используются мнемоники в тестировании?
 1. чтобы провести доскональное и полное тестирование всей функциональности
 2. чтобы быстро предположить, как именно будет работать та или иная функциональность в рамках конкретной задачи
 3. чтобы сгенерировать идеи для тестирования, когда свои уже закончились
 4. чтобы посмеяться в обеденный перерыв
 4. Какая из мнемоник используется для тестирования локализации?
 1. SFDPOT
 2. RCRCRC
 3. FEW HICcupPS
 4. SPIES

Тест к уроку 1.7. «Метрики и отчеты тестирования»

1. Выберите цели формирования отчетов:
 1. Зафиксировать результат выполнения задачи
 2. Определить виновного в возникших проблемах
 3. Соотнести текущий прогресс с плановым
 4. Сформировать у руководства позитивный взгляд на выполнение задач
2. Выберите корректные цели сбора метрик:
 1. Анализ текущей ситуации для поиска решения проблем
 2. Найти проблемы и наказать виновных
 3. Мониторинг прогресса достижения цели
 4. Порадовать руководство цифрами
3. Какого параметра метрик не существует:
 1. Потребитель метрики
 2. Генератор метрики
 3. Способ сбора
 4. Регулярность сбора и оценки
 5. Интерпретатор метрики

Тест к теме 1. «Основы тестирования»

1. Что из этого НЕ ЯВЛЯЕТСЯ целью тестирования?
 1. Проверить, все ли указанные требования выполнены
 2. Предотвращение дефектов
 3. Обеспечение качества ПО для конечных пользователей
 4. Сокращение сроков проекта
2. Что из этого НЕ ОТНОСИТСЯ к артефактам тестировщика (документ, который создает непосредственно тестировщик)?
 1. Тест-кейс
 2. Чек-лист
 3. Требования к продукту
 4. Отчет по тестированию
3. Когда для тестировщика эффективна оценка рисков?
 1. На этапе тест-анализа
 2. На этапе тестирования
 3. В конце проекта
4. Проектирование и создание тест-кейсов – это этап
 1. Тест-аналитики
 2. Тест-дизайна
 3. Планирования проекта
5. Что из перечисленного является назначением чек-листа?
 1. Стандартизация
 2. Уменьшение общего числа ошибок
 3. Возможность делегирования
 4. Все вышеперечисленное
6. Что такое Mind Map?
 1. алгоритм действий, которые требуется совершить для проверки работы программы
 2. графическое представление идей и концепций
 3. базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 4. отчет об ошибке
7. Какие атрибуты являются обязательными для тест-кейса?

1. Предусловия, постусловия, приоритет
 2. Идентификатор, название, шаги и ожидаемый результат
 3. Идентификатор, название, шаги и фактический результат
 4. Идентификатор, условия, название, шаги, тестовые данные, комментарий и ожидаемый результат
8. Какие статусы могут быть у тест-кейса?
1. пройден, провален, известная ошибка, сломан, заблокирован, пропущен
 2. пройден, провален, пропущен, заблокирован
 3. статусом тест-кейса является фактический результат
 4. запланирован, в работе, выполнен
9. Как называется процесс сбора и уточнения сведений о возникновении дефекта?
1. Подтверждение
 2. Поиск
 3. Локализация
 4. Оформление
10. Кто обычно определяет приоритет исправления дефекта?
1. Разработчик
 2. Руководитель проекта \ Владелец продукта
 3. Тестирующий
 4. Пользователь
11. Что такое эвристика?
1. графическое представление идей и концепций
 2. использование предыдущего опыта и информации о вероятности наступления различных событий
 3. базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 4. совокупность правил и приемов, которые помогают запоминать нужные сведения и данные
12. Для чего используются мнемоники в тестировании?
1. чтобы провести доскональное и полное тестирование всей функциональности
 2. чтобы быстро предположить, как именно будет работать та или иная функциональность в рамках конкретной задачи
 3. чтобы сгенерировать идеи для тестирования, когда свои уже закончились
 4. чтобы посмеяться в обеденный перерыв
13. Выберите цели формирования отчетов:
1. Зафиксировать результат выполнения задачи
 2. Определить виновного в возникших проблемах
 3. Соотнести текущий прогресс с плановым
 4. Сформировать у руководства позитивный взгляд на выполнение задач
14. Какого параметра метрик не существует?
1. Потребитель метрики
 2. Генератор метрики
 3. Способ сбора
 4. Регулярность сбора и оценки
 5. Интерпретатор метрики

Тест к уроку 2.1. «Требования к продукту. Анализ, визуализация и декомпозиция»

1. Что такое техзадание?
 1. Алгоритм действий, которые требуется совершить для проверки работы программы

2. Документ, включающий в себя список задач, обязанностей и требований, которые заказчик предъявляет к разрабатываемому продукту
 3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 - 4.Arteфакт, который описывает совокупность конкретных шагов, условий и параметров, необходимый для проверки реализации тестируемой функциональности или ее части.
2. Что из перечисленного обязательно должно быть в ТЗ?
 1. Основные положения, цель и назначение ПО, стоимость разработки
 2. Основные положения, цель и назначение ПО, требования, личные контакты всех, кто составлял ТЗ
 3. Основные положения, требования к ПО, состав и содержание работ, порядок контроля и приемки
 4. Подробное описание каждой функциональности с используемыми технологиями
 3. Что такое функциональные требования?
 1. Описание того, что должно произойти после каждого действия в системе
 2. Описание действий системы с точки зрения бизнес-логики
 3. Любые требования к системе
 4. Описание того, как именно должны осуществляться те или иные действия системы
 4. Что является критерием качества документации?
 1. Корректность и полнота
 2. Проверяемость и прослеживаемость
 3. Недвусмысленность и гибкость
 4. Все вышеперечисленное
 5. Как проводится горизонтальная декомпозиция требований?
 1. В соответствии с отдельной функциональностью.
 2. Исходя из каждого отдельного пожелания заказчика
 3. Исходя из типа проводимых работ, из задействованных в работе системы компонентов или из области работ конкретных специалистов
 4. Исходя из пожеланий директора

Тест к уроку 2.2. «Классы эквивалентности»

1. Что такое техники тест-дизайна?
 1. Алгоритм действий, которые требуется совершить для проверки работы программы
 2. Этап процесса тестирования, где создаются тест-кейсы, в соответствии с требованиями и целями тестирования
 3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 4. Правила и подходы, которые помогают создавать минимальное количество тест-кейсов для максимального покрытия
2. Что из перечисленного является техникой тест-дизайна?
 1. Анализ граничных значений
 2. Парное тестирование
 3. Диаграмма перехода состояний
 4. Все вышеперечисленное
3. Что такое классы эквивалентности?
 1. Группы значений, образующиеся при разбиении тестовых данных по определенному общему признаку

2. Группы положительных и отрицательных чисел
3. Группы числовых и буквенных тестовых данных
4. Граничные значения для проверки каждого поля
4. Как применить технику эквивалентного разделения?
 1. Проанализировать требования, взять одно валидное значение, провести с ним тестирование, убедиться, что ошибка не появляется
 2. Проанализировать требования, выделить классы эквивалентности, выбрать один класс эквивалентности, провести тестирование с каждым значением в этом классе
 3. Проанализировать требования, выделить классы эквивалентности, выбрать по одному значению из каждого класса эквивалентности, провести тестирование с этими значениями
 4. Выбрать несколько случайных значений, провести проверки с ними
5. Что входит в граничные условия при тестировании проверки на совершеннолетие (при регистрации нового пользователя в российской системе)?
 1. 17, 95
 2. 5, 18
 3. 14, 21
 4. 17, 18

Тест к уроку 2.3. «Попарное тестирование (pairwise)»

1. Что такое попарное тестирование?
 1. Техника тест-дизайна, в которой разбиение тестовых данных происходит таким образом, чтобы каждое тестируемое значение одних проверяемых параметров хотя бы раз сочеталась с каждым из тестируемых значений других проверяемых параметров
 2. Этап процесса тестирования, где создаются тест-кейсы, в соответствии с требованиями и целями тестирования
 3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 4. Правила и подходы, которые помогают создавать минимальное количество тест-кейсов для максимального покрытия
2. Почему техника попарного тестирования считается достаточной?
 1. Потому что нам сложно сделать таблицу для трех и более параметров, учитывая все комбинации
 2. Потому что более опытные тестировщики так считают
 3. Потому что именно ошибки, которые возникают при взаимодействии двух параметров являются наиболее критичными
3. Для чего нужно попарное тестирование?
 1. Чтобы разбить тестовые данные на группы по определенному общему признаку
 2. Чтобы выделить граничные значения для всех классов эквивалентности
 3. Для составления тест-кейсов и тест сценариев для проверки программного обеспечения с большим количеством параметров
 4. Для составления тест-кейсов для проверки веб и мобильных приложений
4. Как составить таблицу для попарного тестирования вручную?
 1. Проанализировать требования, взять одно валидное значение, провести с ним тестирование, убедиться, что ошибка не появляется
 2. Проанализировать требования, выделить классы эквивалентности, выбрать один класс эквивалентности, провести тестирование с каждым значением в этом классе

3. Проанализировать требования, выделить классы эквивалентности, выбрать по одному значению из каждого класса эквивалентности, провести тестирование с этими значениями
4. Проанализировать требования, перечислить тестируемые переменные, выделить КЭ, отсортировать параметры и заполнить полностью таблицу

Тест к уроку 2.4. «Граничные значения и доменный анализ»

1. Назовите граничные значения множества [5,5; 6,1]
 1. 5,4; 6,2
 2. 5,4; 5,5; 6,1; 6,2
 3. 5,4; 5,5; 5,6; 6; 6,1; 6,2
 4. 5,5; 5,6; 5,7; 6,1; 6,2; 6,3
2. Есть спидометр, который отслеживает скорость автомобиля и выводит сообщение на приборную панель:
 - От 0 до 10 - медленно
 - От 11 до 30 - отличная скорость
 - от 31 до 60 - кажется вы превысили
3. Составьте числовой ряд, для проверки граничных значений.
 1. -1,0,1,9,10,11,29,30,31,32,59,60,62
 2. 0,1,9,10,11,29,30,31,32,59,60,62
 3. 0,1,9,10,11,29,30,31,32,59,60
 4. -1,0,1,9,10,11,29,30,31,32,59,60
4. Укажите граничные значения для тестирования отображения знаков зодиака для Льва форма из слайда 14 (даты 23.07-23.08)
 1. 22/07; 23/07; 23/07; 22/08; 22/08; 23/08
 2. 23:59 22/07; 00:00 23/07; 00:01 23/07; 23:58 22/08; 23:59 22/08; 00:00 23/08
 3. 23:59 22/07; 00:00 23/07; 00:01 23/07; 23:58 23/08; 23:59 23/08; 00:00 24/08
 4. 23:59; 00:00; 00:01; 23:58; 23:59; 00:00

Тест к уроку 2.5. «Таблицы принятия решений»

1. Сколько колонок нужно для проверки 5 условий
 1. 5
 2. 32
 3. 16
 4. 2
2. Заполните промежуток TTFFTTxxxxFTTFF
 1. FFTT
 2. FTTF
 3. FFTTF
 4. FTTF
3. Заполните промежуток TTTTxxxxTTTTFFFF
 1. FFTT
 2. TTTT
 3. FFFF
 4. FFFT
4. Сколько колонок нужно для проверки 4 условий
 1. 4
 2. 8
 3. 16
 4. 2

Тест к уроку 2.6. «Переходы состояний»

1. Что из перечисленного является преимуществом использования диаграмм состояний и переходов?
 1. Облегчение создания продукта
 2. Визуализация состояний продукта
 3. Повышение скорости выполнения кода
 4. Увеличение количества ошибок
2. Какой из нижеперечисленных уровней тестового покрытия является низким?
 1. Набор тестов, в котором проверяются все состояния системы
 2. Набор тестов, в котором проверяются все переходы системы
 3. Набор тестов, в котором проверяются все действия пользователя
 4. Набор тестов, в котором проверяется только основной сценарий
3. Реакция приложения на действия пользователя и самой системы:
 1. Состояние
 2. Переход
 3. Операция
 4. Действие
4. Как следует размещать главную последовательность состояний на диаграмме состояний и переходов?
 1. В виде ответвлений
 2. На одной горизонтальной линии
 3. В центре диаграммы
 4. В виде вертикальных линий
5. Использование таблиц переходов и состояний является наиболее оправданным при тестировании:
 1. Приложений со сложной структурой и логикой
 2. Веб-приложений
 3. Приложений без согласованных требований
 4. Приложений на начальном этапе разработке

Тест к уроку 2.7. «Другие техники тест-дизайна (причина-следствие, предугадывание ошибок, исследовательское тестирование)»

1. Какую технику тестирования можно использовать для написания тест-кейсов на основе возможных ошибок?
 1. Причина-следствие
 2. Исследовательское тестирование
 3. Предугадывание ошибок
 4. Метод граничных значений
2. Исследовательское тестирование — это техника тестирования, которая:
 1. Включает в себя только автоматические тесты, не требующие участия тестировщиков
 2. Используется только для тестирования мобильных приложений.
 3. Позволяет выявить все возможные дефекты приложения с помощью тест-кейсов.
 4. Включает в себя исследование приложения с целью выявления потенциальных проблем и дефектов без заранее определенных тест-кейсов.
3. Техника тест-дизайна "Причина-следствие":
 1. Используется для создания комбинаций условий и получения ответа от системы
 2. Используется для создания случайных тестовых сценариев
 3. Используется для нахождения ошибок в коде
 4. Используется для создания документации
4. Использование исследовательского тестирования наиболее эффективно: (21809)

1. когда мало времени на тестирование
 2. на проектах со сложной бизнес-логикой
 3. когда требования уже написаны, но разработка приложения еще не началась
 4. когда на проекте много начинающих тестировщиков.
5. Техника тест-дизайна “Предугадывание ошибок” предполагает: (21810)
1. создание формальной документации, в которой содержится информация о возможных рисках и дефектах
 2. наличие у тестировщика опыта тестирования разнообразных проектов
 3. наличие большого количества времени на тестирование
 4. составление отчета по результатам ее применения.

Тест к теме 2. «Тест-анализ и техники тест-дизайна»

1. Что такое техзадание?
 1. Алгоритм действий, которые требуется совершить для проверки работы программы
 2. Документ, включающий в себя список задач, обязанностей и требований, которые заказчик предъявляет к разрабатываемому продукту
 3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
 - 4.Arteфакт, который описывает совокупность конкретных шагов, условий и параметров, необходимый для проверки реализации тестируемой функциональности или ее части.
2. Что такое функциональные требования?
 1. Описание того, что должно произойти после каждого действия в системе
 2. Описание действий системы с точки зрения бизнес-логики
 3. Любые требования к системе
 4. Описание того, как именно должны осуществляться те или иные действия системы
3. Что из перечисленного является техникой тест-дизайна?
 1. Анализ граничных значений
 2. Парное тестирование
 3. Диаграмма перехода состояний
 4. Все вышеперечисленное
4. Как применить технику эквивалентного разделения?
 1. Проанализировать требования, взять одно валидное значение, провести с ним тестирование, убедиться, что ошибка не появляется
 2. Проанализировать требования, выделить классы эквивалентности, выбрать один класс эквивалентности, провести тестирование с каждым значением в этом классе
 3. Проанализировать требования, выделить классы эквивалентности, выбрать по одному значению из каждого класса эквивалентности, провести тестирование с этими значениями
 4. Выбрать несколько случайных значений, провести проверки с ними
5. Почему техника парного тестирования считается достаточной?
 1. Потому что нам сложно сделать таблицу для трех и более параметров, учитывая все комбинации
 2. Потому что более опытные тестировщики так считают
 3. Потому что именно ошибки, которые возникают при взаимодействии двух параметров являются наиболее критичными
6. Как составить таблицу для парного тестирования вручную?

1. Проанализировать требования, взять одно валидное значение, провести с ним тестирование, убедиться, что ошибка не появляется
 2. Проанализировать требования, выделить классы эквивалентности, выбрать один класс эквивалентности, провести тестирование с каждым значением в этом классе
 3. Проанализировать требования, выделить классы эквивалентности, выбрать по одному значению из каждого класса эквивалентности, провести тестирование с этими значениями
 4. Проанализировать требования, перечислить тестируемые переменные, выделить КЭ, отсортировать параметры и заполнить полностью таблицу
7. Назовите граничные значения множества [5,5; 6,1]
 1. 5,4; 6,2
 2. 5,4; 5,5; 6,1; 6,2
 3. 5,4; 5,5; 5,6; 6; 6,1; 6,2
 4. 5,5; 5,6; 5,7; 6,1; 6,2; 6,3
 8. Есть спидометр, который отслеживает скорость автомобиля и выводит сообщение на приборную панель:
 - От 0 до 10 - медленно
 - От 11 до 30 - отличная скорость
 - от 31 до 60 - кажется вы превысили
 9. Составьте числовой ряд, для проверки граничных значений.
 1. -1,0,1,9,10,11,29,30,31,32,59,60,62
 2. 0,1,9,10,11,29,30,31,32,59,60,62
 3. 0,1,9,10,11,29,30,31,32,59,60
 4. -1,0,1,9,10,11,29,30,31,32,59,60
 10. Заполните промежуток TTFFTTxxxxxFTTFF
 1. FFTT
 2. FTTFF
 3. FFTTF
 4. FTTFF
 11. Заполните промежуток TTTTxxxxTTTTFFFF
 1. FFTT
 2. TTTT
 3. FFFF
 4. FFFT
 12. Что из перечисленного является преимуществом использования диаграмм состояний и переходов?
 1. Облегчение создания продукта
 2. Визуализация состояний продукта
 3. Повышение скорости выполнения кода
 4. Увеличение количества ошибок
 13. Как следует размещать главную последовательность состояний на диаграмме состояний и переходов?
 1. В виде ответвлений
 2. На одной горизонтальной линии
 3. В центре диаграммы
 4. В виде вертикальных линий
 14. Какую технику тестирования можно использовать для написания тест-кейсов на основе возможных ошибок?
 1. Причина-следствие

2. Исследовательское тестирование
 3. Предугадывание ошибок
 4. Метод граничных значений
15. Техника тест-дизайна "Причина-следствие":
1. Используется для создания комбинаций условий и получения ответа от системы
 2. Используется для создания случайных тестовых сценариев
 3. Используется для нахождения ошибок в коде
 4. Используется для создания документации

Тест к уроку 3.1. «Введение в веб-разработку»

1. Что такое веб-приложение?
 1. Приложение, которое устанавливаются на компьютер и может быть использовано без подключения к интернету.
 2. Приложение, которое устанавливается на девайсы, например, телефон или планшет, и может использоваться без подключения к интернету.
 3. Приложение, которые работает с использованием интернета.
 4. Приложение, у которых интерфейсом служит командная строка.
2. Из чего состоит веб-приложение?
 1. Ядро, пользователь интерфейс, компоненты повторного использования, файлы окружения
 2. Сервер, клиент
 3. База данных, браузер Анализ граничных значений
 4. Сервер, клиент, база данных, браузер, интернет
3. Что из перечисленного обычно отличает веб-приложения от веб-сайта?
 1. Аутентификация и использование интернета
 2. Отображение статических страниц, а не генерация их «на лету»
 3. Возможность манипулировать данными, аутентификация
 4. Все вышеперечисленное
4. Какие особенности учитывают при разработке веб-приложения?
 1. Скорость
 2. Гибкость
 3. Ранний фидбек
 4. Все вышеперечисленное
5. Что не следует реализовывать в интерфейсе веб-приложения?
 1. Адаптивный дизайн, чтобы интерфейс оставался удобным и читабельным на каждом устройстве
 2. Как можно больше различных шрифтов, чтобы выделить каждую часть приложения
 3. Визуальную иерархию, чтобы интуитивно было понятно, где искать функциональность
 4. Лучшие практики и устоявшиеся решения

Тест к уроку 3.2. «Основы HTML»

1. Что обязательно должно быть в любом HTML-документе?
 1. `<html> </html>` и `<input> </input>`
 2. `<!DOCTYPE>` и `<html> </html>`
 3. `<!DOCTYPE>` и `<header> </header>`
 4. `<!DOCTYPE>`, `<html> </html>`, `<header> </header>`, `<body> </body>`, `<input> </input>`
2. Что можно отнести к элементам метаинформации?
 1. тег `<input>`
 2. тег `<header>`

3. `<!DOCTYPE>`
4. тег `<meta>`
3. Какой тег нужно использовать, если вы хотите выделить текст жирным и обозначить его важность логически?
 1. ` `
 2. ` `
 3. ` `
 4. `<i> </i>`
4. Какой тег используется для создания маркированного списка?
 1. ` `
 2. ` `
 3. `<nav> </nav>`
 4. ` `
5. Какой скрипт в результате выдаст кнопку правильного размера с изображением и текстом «Нажми»?
 1. `<button type="submit" src="https://cdn-icons-png.flaticon.com/512/439/439842.png" alt="" /> Нажми </button>`
 2. `<button type="submit"> Привет </button>`
 3. `<button type="submit"> `
 4. `<button type="submit"> Нажми </button>`

Тест к уроку 3.3. «Основы CSS»

1. Что такое CSS?
 1. Это прогрессивное веб-приложение
 2. Это язык программирования
 3. Это язык гипертекстовой разметки
 4. Это язык таблицы стилей
2. Из чего состоят правила CSS?
 1. селектора и атрибута
 2. селектора, свойства и команды
 3. свойства и команды
 4. селектора, свойства и значения
3. Какой селектор приведен в примере?
`.article-title { margin-bottom: 5px; }`
 1. Селектор идентификатора
 2. Селектор элемента
 3. Селектор класса
 4. Селектор идентификатора
4. Какие CSS свойства входят в категорию Текст?
 1. list-style-image, list-style-position, list-style-type
 2. text-align, text-decoration-color, text-indent
 3. border-left-color, border-left-style, border-left-width
 4. background-blend-mode, background-clip, background-color
5. При каком способе применения стилей разработчик создает отдельный CSS-документ и импортирует его в HTML-документ с помощью команды `@import`?
 1. Встроенные стили

2. Внутренние стили
3. Импортированные стили
4. Связанные стили

Тест к уроку 3.4. «Разработка веб-приложения»

1. Для чего используется CSS?
 1. Для разметки веб-страницы
 2. Для определения стилей веб-страницы
 3. Для создания прогрессивного веб-приложения
 4. Для тестирования веб-приложений
2. При каком способе применения стилей разработчик создает отдельный CSS-документ и импортирует его в HTML-документ с помощью тега <link>?
 1. Встроенные стили
 2. Внутренние стили
 3. Импортированные стили
 4. Связанные стили
3. Какой тег отвечает за перенос строки в HTML?
 1. <p>
 2. <meta>
 3. <section>
 4.

4. Какое свойство используется для закругления углов изображения в CSS?
 1. width
 2. padding
 3. border-collapse
 4. border-radius
5. Какого цвета будет «Заголовок» в h2 на веб-странице?

```
<!DOCTYPE html>
<html>
  <head>
    <title>Имя вкладки</title>
    <style>
      h2 {
        font-size: 120%;
        font-family: Verdana, Arial, Helvetica, sans-serif;
        color: green;
      }
    </style>
  </head>
  <body>
    <header>
      <h2 style="font-size: 120%; font-family: monospace; color: blue "> Заголовок </h2>
    </header>
    <main>
      <h2 style="font-size: 120%; font-family: monospace; color: red" >Техники тест дизайна</h2>
    </main>
  </body>
</html>
```

1. черный
2. зеленый
3. синий
4. красный

Тест к теме 3. «Работа с HTML и CSS»

1. Что такое веб-приложение?
 1. Приложение, которое устанавливается на компьютер и может быть использовано без подключения к интернету.
 2. Приложение, которое устанавливается на девайсы, например, телефон или планшет, и может использоваться без подключения к интернету.
 3. Приложение, которое работает с использованием интернета.
 4. Приложение, у которого интерфейсом служит командная строка.
2. Какие особенности учитывают при разработке веб-приложения?
 1. Скорость

2. Гибкость
 3. Ранний фидбек
 4. Все вышеперечисленное
3. Что можно отнести к элементам метаинформации?
1. тег `<input>`
 2. тег `<header>`
 3. `<!DOCTYPE>`
 4. тег `<meta>`
4. Какой тег используется для создания маркированного списка?
1. ` `
 2. ` `
 3. `<nav> </nav>`
 4. ` `
5. Что такое CSS?
1. Это прогрессивное веб-приложение
 2. Это язык программирования
 3. Это язык гипертекстовой разметки
 4. Это язык таблицы стилей
6. Какой селектор приведен в примере?
- ```
.article-title { margin-bottom: 5px; }
```
1. Селектор идентификатора
  2. Селектор элемента
  3. Селектор класса
  4. Селектор идентификатора
7. При каком способе применения стилей разработчик создает отдельный CSS-документ и импортирует его в HTML-документ с помощью команды `@import`?
1. Встроенные стили
  2. Внутренние стили
  3. Импортированные стили
  4. Связанные стили
6. Для чего используется CSS?
1. Для разметки веб-страницы
  2. Для определения стилей веб-страницы
  3. Для создания прогрессивного веб-приложения
  4. Для тестирования веб-приложений
7. Какой тег отвечает за перенос строки в HTML?
1. `<p>`
  2. `<meta>`
  3. `<section>`
  4. `<br>`
8. Какого цвета будет «Заголовок» в h2 на веб-странице?

```

<!DOCTYPE html>
<html>
 <head>
 <title>Имя вкладки</title>
 <style>
 h2 {
 font-size: 120%;
 font-family: Verdana, Arial, Helvetica, sans-serif;
 color: green;
 }
 </style>
 </head>
 <body>
 <header>
 <h2 style="font-size: 120%; font-family: monospace; color: blue "> Заголовок </h2>
 </header>
 <main>
 <h2 style="font-size: 120%; font-family: monospace; color: red" >Техники тест дизайна</h2>
 </main>
 </body>
</html>

```

1. черный
2. зеленый
3. синий
4. красный

#### Тест к уроку 4.1. «Основы тестирования веб-приложений»

1. Что такое ip компьютера?
  1. Инвентарный номер
  2. Адрес компьютера в магазине
  3. Адрес компьютера в сети
  4. Номер, по которому можно позвонить на компьютер
2. Одностраничное приложение это?
  1. PWA
  2. MPA
  3. SPA
  4. SWA
3. Многостраничное приложение это?
  1. PWA
  2. MPA
  3. SPA
  4. SWA
4. Какой код ошибки придет, если ошибка на стороне сервера?
  1. 600
  2. 500
  3. 400
  4. 300

#### Тест к уроку 4.2. «DevTools»

1. Как называется вкладка, где можно изменить элементы?
  1. Элементы
  2. Консоль
  3. Сеть
  4. Приложение
2. Как называется вкладка, где можно проверить скорость запросов?
  1. Элементы
  2. Консоль
  3. Сеть
  4. Приложение
3. Как называется вкладка, где можно увидеть контент запросов и ответов?
  1. Элементы

2. Консоль
  3. Сеть
  4. Приложение
4. Как называется вкладка, где можно удалить куки?
    1. Элементы
    2. Консоль
    3. Сеть
    4. Приложение

Тест к уроку 4.3. «Сбор и анализ трафика»

1. На каком этапе Чарльз встраивается в клиент-серверную архитектуру?
  1. На первом месте
  2. Посередине
  3. Только после сервера
  4. Только после клиента
2. Как называется функция замедления интернета?
  1. Proxy
  2. Throttling
  3. Rewrite
  4. BreakPoint
3. Как называется функция принудительного разрыва соединения?
  1. Proxy
  2. Throttling
  3. Rewrite
  4. BreakPoint
4. Как называется функция перезаписи запросов или ответов к серверу?
  1. Proxy
  2. Throttling
  3. Rewrite
  4. BreakPoint

Тест к теме 4. «Тестирование Веб-приложений»

1. Что такое ip компьютера?
  1. Инвентарный номер
  2. Адрес компьютера в магазине
  3. Адрес компьютера в сети
  4. Номер, по которому можно позвонить на компьютер
2. Одностраничное приложение это?
  1. PWA
  2. MPA
  3. SPA
  4. SWA
3. Какой код ошибки придет, если ошибка на стороне сервера?
  1. 600
  2. 500
  3. 400
  4. 300
4. Как называется вкладка, где можно проверить скорость запросов?
  1. Элементы
  2. Консоль
  3. Сеть
  4. Приложение

5. Как называется вкладка, где можно увидеть контент запросов и ответов?
  1. Элементы
  2. Консоль
  3. Сеть
  4. Приложение
6. Как называется вкладка, где можно удалить куки?
  1. Элементы
  2. Консоль
  3. Сеть
  4. Приложение
5. На каком этапе Чарльз встраивается в клиент-серверную архитектуру?
  1. На первом месте
  2. Посередине
  3. Только после сервера
  4. Только после клиента
6. Как называется функция замедления интернета?
  1. Proxy
  2. Throttling
  3. Rewrite
  4. BreakPoint
7. Как называется функция принудительного разрыва соединения?
  1. Proxy
  2. Throttling
  3. Rewrite
  4. BreakPoint
8. Как называется функция перезаписи запросов или ответов к серверу?
  1. Proxy
  2. Throttling
  3. Rewrite
  4. BreakPoint

#### Тест к уроку 5.1. «Клиент-серверная архитектура»

1. Какой тип архитектуры описан? Несколько пользователей могут работать с одними данными. Выделяется один сервер. Но сервер самостоятельно не обрабатывает никакие данные, все изменения передаются по сети, поэтому иногда это приводит к высоким нагрузкам и замедлению производительности.
  1. Клиент-серверная
  2. Файл-серверная
  3. Монолитная
  4. Многозвенная
2. Какой тип архитектуры описан? Сервер вынесен отдельно от клиентов. На нем хранятся и сразу же обрабатываются данные. И есть клиенты, на стороне которых представляются и обрабатываются данные.
  1. Клиент-серверная
  2. Файл-серверная
  3. Монолитная
  4. Многозвенная
3. Как называется способ взаимодействия ИС с помощью удаленного вызова?
  1. Через файлы

2. Через БД
  3. Через API
  4. Асинхронный обмен сообщениями
4. Как называется промежуточный компонент взаимодействия?
    1. Шина
    2. Колесо
    3. Обруч
    4. Кольцо

Тест к уроку 5.2. «REST API»

1. С каким форматом данных работает REST?
  1. JSON
  2. XML
  3. HTTP
  4. HTTPS
2. Как называется способ аутентификации через соц сеть?
  1. SSO
  2. JWT
  3. OAuth
  4. Беспарольная аутентификация
3. Что нам говорит код ответа, который начинается на 5..?
  1. Ошибка клиента
  2. Ошибка сервера
  3. Редирект
  4. Успешный ответ
4. Что ставится на 2 место в URL?
  1. Протокол
  2. Адрес сервера
  3. Метод
  4. id элемента

Тест к уроку 5.3. «SOAP API»

1. С каким форматом данных работает протокол SOAP?
  1. XML
  2. JSON
  3. HTTP
  4. HTTPS
2. Как называется главный элемент в синтаксисе XML?
  1. Главный
  2. Основной
  3. Корневой
  4. Фундаментальный
3. Выберите синтаксис комментария:
  1. <!-- Это комментарий -->
  2. <-- Это комментарий !-->
  3. <!-- Это комментарий !-->
  4. <-- Это комментарий -->
4. Во что заключаются значения атрибутов в xml?
  1. Кавычки
  2. Круглые скобки
  3. Фигурные скобки
  4. Восклицательный знак



#### Тест к уроку 5.4. «Техники тест-дизайна при тестировании API и составление тестовой документации»

1. Дайте понятие определению - алгоритм действий, которые требуется совершить для проверки работы программы (кнопок, полей ввода и т.д.). В него входят шаги, которые предпринимаются перед проверкой (предусловия), являются проверкой, а также ожидаемый результат — то, что получим после выполненных действий.
  1. Шаг
  2. Ожидаемый результат
  3. Тест-кейс
  4. Чек-лист
2. Дайте понятие определению - перечень, состоящий из пунктов, которые нужно выполнить либо проверить.
  1. Шаг
  2. Ожидаемый результат
  3. Тест-кейс
  4. Чек-лист
3. Как в Swagger помечается обязательное поле?
  1. Тире
  2. Скобки круглые
  3. Звездочка
  4. Скобки фигурные
4. Какая техника тест-дизайна используется при тестировании различных кодов ответа?
  1. Исследовательское тестирование
  2. Матрица решений
  3. Граничные значение
  4. Эквивалентное разбиение

#### Тест к теме 5. «Тестирование API»

1. Какой тип архитектуры описан? Несколько пользователей могут работать с одними данными. Выделяется один сервер. Но сервер самостоятельно не обрабатывает никакие данные, все изменения передаются по сети, поэтому иногда это приводит к высоким нагрузкам и замедлению производительности.
  1. Клиент-серверная
  2. Файл-серверная
  3. Монолитная
  4. Многозвенная
2. Как называется способ взаимодействия ИС с помощью удаленного вызова?
  1. Через файлы
  2. Через БД
  3. Через API
  4. Асинхронный обмен сообщениями
3. Как называется способ аутентификации через соц сеть?
  1. SSO
  2. JWT
  3. Oauth
  4. Беспарольная аутентификация
4. Что ставится на 2 место в URL?
  1. Протокол
  2. Адрес сервера
  3. Метод
  4. id элемента

5. Как называется главный элемент в синтаксисе XML?
  1. Главный
  2. Основной
  3. Корневой
  4. Фундаментальный
6. Во что заключаются значения атрибутов в xml?
  1. Кавычки
  2. Круглые скобки
  3. Фигурные скобки
  4. Восклицательный знак
7. Дайте понятие определению - перечень, состоящий из пунктов, которые нужно выполнить либо проверить.
  1. Шаг
  2. Ожидаемый результат
  3. Тест-кейс
  4. Чек-лист
5. Какая техника тест-дизайна используется при тестировании различных кодов ответа?
  1. Исследовательское тестирование
  2. Матрица решений
  3. Граничные значение
  4. Эквивалентное разбиение
6. Выберите синтаксис комментария:
  1. <!-- Это комментарий -->
  2. <-- Это комментарий !-->
  3. <!-- Это комментарий !-->
  4. <-- Это комментарий -->
7. Что нам говорит код ответа, который начинается на 5..?
  1. Ошибка клиента
  2. Ошибка сервера
  3. Редирект
  4. Успешный ответ

Тест к уроку 6.1. «Как устроены мобильные приложения»

1. Какое утверждение верно при тестировании приложений на ОС Андроид / iOS?
  1. не обязательно тестировать гибридные мобильные приложения на обеих платформах, даже если предполагается, что приложение должно работать и на Android и на iOS
  2. при тестировании МП нужно проверить его работоспособность на всех выпущенных версиях ОС
  3. анализ целевой аудитории приложения, если приложение уже доступно пользователям, может помочь определиться с выбором версий ОС для дальнейшего тестирования приложения
2. Какой из перечисленных пунктов НЕ является типом мобильного приложения?
  1. гибридное МП
  2. Web View
  3. mobile Web
3. Что из нижеперечисленного НЕ является форматом бандла для iOS или Android приложений?
  1. abb
  2. ipa
  3. adb

4. apk
4. Как называется утилита для озвучки контента в приложениях на iOS?
  1. Talk Back
  2. Voice Over
5. Как называется UI-guideline для приложений на Андроид?
  1. HIG
  2. Material Design
6. Какой из нижеперечисленных типов мобильных приложений может поддерживать работу полностью без интернета?
  1. нативные МП
  2. гибридные МП
  3. mobile Web

Тест к уроку 6.2. «Особенности тестирования мобильных приложений»

1. Что из нижеперечисленного является системным прерыванием работы приложения?
  1. потеря подключения к WiFi
  2. сворачивание приложения пользователем
  3. аварийное завершение работы мобильного приложения из-за утечек памяти
  4. потеря сигнала GPS
2. Как называется технология, которая позволяет определить геолокацию устройства при использовании данных и от WiFi и от спутников?
  1. A-GPS
  2. GPS
3. Что из перечисленных функциональностей не требует запроса дополнительного permission?
  1. камера
  2. открытие клавиатуры
  3. звонки
  4. галерея и доступ к фото
4. Что из перечисленного не является прерыванием работы приложения?
  1. потеря сетевого соединения во время работы приложения
  2. клик пользователя в приложении, вызывающий переход на другой экран
  3. сворачивание приложения пользователем
5. Работа приложения в фоновом режиме называется:
  1. работой в foreground
  2. работой в background

Тест к уроку 6.3. «Тестирование мобильных приложений Android»

1. Что такое жизненный цикл активити в Android-приложениях?
  1. Набор методов, которые определяют, как Android управляет активити во время ее жизненного цикла.
  2. Серия шагов, необходимых для создания и запуска активности в приложении Android.
  3. Список действий, которые должны быть выполнены для просмотра и изменения информации об активности.
2. Какой метод вызывается перед уничтожением активити?
  1. onPause()
  2. onDestroy()
  3. onStop()
3. Что такое чистый Android и как он связан с тестированием мобильных приложений на Android?

1. Это версия операционной системы Android без каких-либо модификаций или дополнений со стороны производителей устройств. Чистый Android использует только стандартные элементы и функции, которые предоставляются в операционной системе.
2. Это версия операционной системы Android с множеством модификаций и дополнений от производителей устройств. Чистый Android не может использоваться для тестирования мобильных приложений.
3. Чистый Android - это специальная версия операционной системы Android, которая используется только для тестирования мобильных приложений и не предназначена для ежедневного использования.
4. Какой фактор может влиять на энергопотребление Android-устройства?
  1. Загруженность процессора
  2. Уровень яркости экрана
  3. GPS
  4. Все перечисленное выше
5. Зачем нужно ограничение процессора в режиме разработчика?
  1. Для тестирования приложений на мощных устройствах
  2. Для улучшения производительности устройства
  3. Для тестирования приложений на слабых устройствах

#### Тест к уроку 6.4. «Среда разработки Android Studio и Android Debug Bridge (ADB)»

1. Что такое системные логи для Android?
  1. Журналы, в которых фиксируются различные события, происходящие в операционной системе и приложениях.
  2. Файлы, содержащие только записи о сбоях и аварийном завершении приложения.
  3. Текстовые файлы, где фиксируются все действия пользователя.
2. Какой уровень системных логов для Android требует немедленного вмешательства разработчика?
  1. ERROR.
  2. WARN.
  3. VERBOSE.
3. Какова роль эмуляторов в тестировании мобильных приложений?
  1. Позволяют тестировать приложения на разных устройствах без наличия реальных устройств
  2. Заменяют реальные устройства для тестирования
  3. Позволяют ускорить тестирование приложений
4. Какой инструмент является стандартным для тестирования в Android Studio?
  1. JUnit
  2. Espresso
  3. Appium
5. Как запустить ADB на компьютере?
  1. Установить Android Studio.
  2. Установить пакет Android SDK Platform Tools.
  3. Необходимо иметь root-доступ к компьютеру.

#### Тест к уроку 6.5. «Тестирование мобильных приложений iOS»

1. Какие методы вызываются при создании ViewController?
  1. viewDidLoad()
  2. viewWillAppear()
  3. viewDidAppear()
  4. все вышеперечисленные

2. Какие методы вызываются при изменении ориентации экрана?
  1. viewWillAppear()
  2. viewWillTransition()
  3. viewWillLayoutSubviews()
  4. viewDidLayoutSubviews()
3. Что такое HIG?
  1. Интерфейс голосового ввода, разработанный компанией Apple.
  2. Руководство по графическому дизайну для приложений на iOS, разработанное компанией Apple.
  3. Система оптимизации производительности устройств iOS, разработанная компанией Apple.
4. Как начать сбор логов в Xcode?
  1. Подключить устройство iOS к компьютеру, выбрать связанное с устройством приложение и нажать на кнопку "Open Console";
  2. Выбрать устройство iOS в качестве целевого устройства для отладки, запустить приложение и автоматически начнется сбор логов;
  3. Создать новый проект в Xcode и задать настройки сбора логов.
5. Зачем необходимо собирать логи после выпуска приложения?
  1. Для анализа поведения приложения и выявления возможных проблем у пользователей
  2. Для увеличения количества загрузок приложения в магазинах приложений
  3. Для сбора статистических данных о пользователях

#### Тест к уроку 6.6. «Сбор и анализ трафика»

1. Что такое Charles Proxy?
  1. Веб-файервол
  2. Инструмент для тестирования производительности
  3. Прокси-сервер для отладки веб-приложений
2. Какой тип трафика можно отслеживать с помощью Charles Proxy?
  1. Только HTTP
  2. Только HTTPS
  3. И HTTP, и HTTPS
3. Как настроить мобильное устройство для использования Charles Proxy?
  1. Установив профиль VPN
  2. Установив корневой сертификат
  3. Подключив устройство к компьютеру, на котором работает Charles Proxy
4. Какова цель ограничения скорости (throttling) в Charles Proxy?
  1. Замедлить трафик для имитации реальных сетевых условий
  2. Ускорить трафик для тестирования производительности
  3. Блокировать трафик
5. Как можно настроить Charles Proxy для перехвата трафика мобильного приложения на удаленном устройстве?
  1. Установив Charles Proxy на удаленное устройство
  2. Используя VPN-соединение и настраивая Charles Proxy для перехвата VPN-трафика
  3. Используя прокси-сервер на удаленном устройстве и настраивая Charles Proxy для перехвата трафика через прокси-сервер

#### Тест к теме 6. «Тестирование мобильных приложений»

1. Какое утверждение верно при тестировании приложений на ОС Android / iOS?

1. не обязательно тестировать гибридные мобильные приложения на обеих платформах, даже если предполагается, что приложение должно работать и на Android и на iOS
2. при тестировании МП нужно проверить его работоспособность на всех выпущенных версиях ОС
3. анализ целевой аудитории приложения, если приложение уже доступно пользователям, может помочь определиться с выбором версий ОС для дальнейшего тестирования приложения
2. Как называется утилита для озвучки контента в приложениях на iOS?
  1. Talk Back
  2. Voice Over
3. Что из нижеперечисленного является системным прерыванием работы приложения?
  1. потеря подключения к WiFi
  2. сворачивание приложения пользователем
  3. аварийное завершение работы мобильного приложения из-за утечек памяти
  4. потеря сигнала GPS
4. Что из перечисленных функциональностей не требует запроса дополнительного permission?
  1. камера
  2. открытие клавиатуры
  3. звонки
  4. галерея и доступ к фото
5. Какой метод вызывается перед уничтожением активити?
  1. onPause()
  2. onDestroy()
  3. onStop()
6. Какой фактор может влиять на энергопотребление Android-устройства?
  1. Загруженность процессора
  2. Уровень яркости экрана
  3. GPS
  4. Все перечисленное выше
7. Какой уровень системных логов для Android требует немедленного вмешательства разработчика?
  1. ERROR.
  2. WARN.
  3. VERBOSE.
8. Как запустить ADB на компьютере?
  1. Установить Android Studio.
  2. Установить пакет Android SDK Platform Tools.
  3. Необходимо иметь root-доступ к компьютеру.
9. Какие методы вызываются при изменении ориентации экрана?
  1. viewWillAppear()
  2. viewWillTransition()
  3. viewWillLayoutSubviews()
  4. viewDidLayoutSubviews()
10. Как начать сбор логов в Xcode?
  1. Подключить устройство iOS к компьютеру, выбрать связанное с устройством приложение и нажать на кнопку "Open Console";
  2. Выбрать устройство iOS в качестве целевого устройства для отладки, запустить приложение и автоматически начнется сбор логов;

3. Создать новый проект в Xcode и задать настройки сбора логов.
11. Какой тип трафика можно отслеживать с помощью Charles Proxy? (22620)
  1. Только HTTP
  2. Только HTTPS
  3. И HTTP, и HTTPS
12. Какова цель ограничения скорости (throttling) в Charles Proxy? (22622)
  1. Замедлить трафик для имитации реальных сетевых условий
  2. Ускорить трафик для тестирования производительности
  3. Блокировать трафик

Тест к уроку 7.1. «Введение в SQL»

1. Что из перечисленного входит в основные элементы базы данных?
  1. Таблица
  2. Поле
  3. Строка
  4. Все вышеперечисленное
2. Что такое таблица?
  1. вертикальный компонент, определяющий, какие атрибуты для нового компонента данных нам требуются.
  2. объект базы данных, который содержит всю информацию по конкретной сущности. Она должна иметь уникальное имя. Таблицы состоят из строк и столбцов.
  3. набор логически упорядоченных значений ключей, то есть набор уникальных идентификаторов строк
  4. обозначение допустимых значений для конкретного столбца.
3. Для чего используется тип данных INT?
  1. Для интеграционного ключа
  2. Для целых чисел
  3. Для текста
  4. Для внутренней отметки о текущем времени
4. Представим, что у нас есть две таблицы в базе данных «Заказчики» и «Заказы». Какую связь следует использовать между этими таблицами?
  1. Один-к-одному
  2. Один-ко-многим
  3. Многие-ко-многим
  4. Следует объединить в одну таблицу
5. Если между таблицами «Заказчики» и «Заказы» нужна связь, то в которой из них должен содержаться первичный ключ, а в которой – внешний?
  1. Первичный ключ в таблице с заказами, а в таблице с заказчиками внешний ключ на виды услуг
  2. Первичный ключ в таблице с заказчиками, а в таблице с заказами внешний ключ на заказчиков
  3. Первичный ключ в таблице с заказами, а в таблице с заказчиками внешний ключ на заказы
  4. Ключи не нужны, так как их следует объединить в одну таблицу

Тест к уроку 7.2. «Главный SQL-запрос: оператор SELECT»

1. Для чего используется оператор SELECT?
  1. для указания из какой таблицы выбираем данные
  2. для указания, какие столбцы выбираем из таблицы
  3. для группировки результатов выборки
  4. для сортировки результатов выборки

2. Какой SQL запрос выведет список книг в жанре фикшн?
  1. `SELECT * FROM BOOKS;`
  2. `SELECT * FROM BOOKS WHERE genre ='фэнтэзи';`
  3. `SELECT 'фикшн' FROM BOOKS WHERE genre ='фикшн';`
  4. `SELECT * FROM BOOKS WHERE genre ='фикшн';`
3. Какой оператор из перечисленных использует ASC и DESC для сортировки по возрастанию или убыванию?
  1. UPDATE
  2. SELECT
  3. ORDER BY
  4. GROUP BY
4. Для чего используется оператор вычисления %?
  1. Для конкатенации строк
  2. Для получения десятичной дроби
  3. Для определения целочисленного остатка от деления
  4. Для указания в шаблоне на любое количество символов
5. Где по умолчанию отображаются значения NULL в результирующей выборке по возрастанию?
  1. В начале
  2. В конце
  3. Зависит от других полей
  4. В рандомном порядке

Тест к уроку 7.3. «Объединение данных: оператор JOIN»

1. Для чего используется оператор JOIN?
  1. Для группировки результатов выборки
  2. Для указания из какой таблицы выбираем данные
  3. Для указания, какие столбцы выбираем из таблицы
  4. Для объединения таблиц
2. Какой оператор используется для выведения всех записей из двух таблиц?
  1. FULL OUTER JOIN
  2. LEFT JOIN
  3. JOIN
  4. RIGHT JOIN
3. Какой оператор используется для отображения полной первой таблицы, независимо от того, есть ли парное значение во второй?
  1. FULL OUTER JOIN
  2. LEFT JOIN
  3. JOIN
  4. RIGHT JOIN
4. Сколько строк будет в таблице, получившейся в результате запроса CROSS JOIN, если в исходных таблицах было 7 и 3 значения соответственно?
  1. 7
  2. 3
  3. 10
  4. 21
5. Какое значение подставляется в пустые ячейки при использовании RIGHT и LEFT JOIN запросов?
  1. Ноль
  2. 0
  3. NULL



4. Остается пустым
6. Что такое комбинаторное объединение?
  1. Объединение, при котором возвращается левая таблица с соответствующими ей значениями в правой таблице
  2. Объединение, при котором возвращается правая таблица с соответствующими ей значениями в левой таблице
  3. Объединение, при котором возвращаются данные из разных таблиц без каких-либо условий сопоставления

Тест к уроку 7.4. «Группировка данных»

1. Что из перечисленного относится к агрегатным функциям, используемым с оператором GROUP BY?
  1. COUNT, FROM, \*
  2. AVG, MIN, MAX
  3. SUM, ASC, DESC
  4. ASC, DESC
2. Для чего используется оператор GROUP BY?
  1. Для обозначения таблицы, из которой выбираем данные
  2. Для сортировки результатов от меньшего к большему
  3. Для выявления среднего арифметического среди значений столбца
  4. Для группировки выборки данных по определенному столбцу
3. Какой оператор используется для сортировки результатов выборки?
  1. GROUP BY
  2. ORDER BY
  3. HAVING
  4. SELECT FROM
4. Что будет результатом данного запроса?

```
SELECT genre, name, COUNT(*) FROM TOP_MOVIES GROUP BY genre HAVING COUNT(*)>1;
```

1. Таблица из трех столбцов: жанр, имя фильма и количество строк с одинаковым жанром
  2. Ошибка
  3. Таблица из двух столбцов: жанр и количество записей
  4. Таблица из двух столбцов: жанр и количество записей, и только те записи, где определенный жанр встречался более одного раза
5. Что будет результатом данного запроса?

```
SELECT duration, COUNT(*) FROM TOP_MOVIES GROUP BY duration HAVING duration BETWEEN 90 and 150;
```

1. Таблица из двух столбцов, содержащая все фильмы и их продолжительность
2. Ошибка
3. Таблица из двух столбцов, содержащая записи с теми фильмами, продолжительность которых от полутора до двух часов
4. Таблица из двух столбцов, содержащая записи с теми фильмами, сгруппированные по продолжительности и только те, в которых продолжительность фильмов от полутора до двух часов

Тест к уроку 7.5. «Модификация данных: INSERT, UPDATE, DELETE»

1. Какой оператор используется для изменения существующих записей в таблице?
  1. GROUP BY
  2. UPDATE
  3. CREATE
  4. SELECT
2. Для чего используется оператор INSERT?

1. Для создания таблицы
  2. Для добавления новой строки
  3. Для редактирования сразу всех строк
  4. Для редактирования одной строки
3. Что будет результатом такого SQL-запроса?

```
DELETE FROM employees;
```

1. Удалятся только четный строки таблицы employees
  2. Удалятся только нечетные строки таблицы employees
  3. Отчет об ошибке
  4. Удалятся все строки таблицы employees
4. Какое ключевое слово используется для управления дубликатами при добавлении данных?
1. INSERT
  2. UPDATE
  3. ON CONFLICT
  4. DEFAULT
5. Для чего используется ключевое слово RETURNING?
1. Чтобы отобразить все значения таблицы
  2. Чтобы отобразить измененные значения
  3. Чтобы отменить сделанные изменения
  4. Чтобы вызвать предыдущий запрос

#### Тест к теме 7. «Базовый SQL»

1. Что такое таблица?
  1. Вертикальный компонент, определяющий, какие атрибуты для нового компонента данных нам требуются.
  2. Объект базы данных, который содержит всю информацию по конкретной сущности. Она должна иметь уникальное имя. Таблицы состоят из строк и столбцов.
  3. Набор логически упорядоченных значений ключей, то есть набор уникальных идентификаторов строк
  4. Обозначение допустимых значений для конкретного столбца.
2. Представим, что у нас есть две таблицы в базе данных «Заказчики» и «Заказы». Какую связь следует использовать между этими таблицами?
  1. Один-к-одному
  2. Один-ко-многим
  3. Многие-ко-многим
  4. Следует объединить в одну таблицу
3. Какой оператор из перечисленных использует ASC и DESC для сортировки по возрастанию или убыванию?
  1. UPDATE
  2. SELECT
  3. ORDER BY
  4. GROUP BY
4. Где по умолчанию отображаются значения NULL в результирующей выборке по возрастанию?
  1. В начале
  2. В конце
  3. Зависит от других полей
  4. В рандомном порядке

5. Какой оператор используется для выведения всех записей из двух таблиц?
  1. FULL OUTER JOIN
  2. LEFT JOIN
  3. JOIN
  4. RIGHT JOIN
6. Какое значение подставляется в пустые ячейки при использовании RIGHT и LEFT JOIN запросов?
  1. Ноль
  2. 0
  3. NULL
  4. Остается пустым
7. Для чего используется оператор GROUP BY?
  1. Для обозначения таблицы, из которой выбираем данные
  2. Для сортировки результатов от меньшего к большему
  3. Для выявления среднего арифметического среди значений столбца
  4. Для группировки выборки данных по определенному столбцу
8. Что будет результатом данного запроса?
 

```
SELECT duration,COUNT(*) FROM TOP_MOVIES GROUP BY duration HAVING duration BETWEEN 90 and 150;
```

  1. Таблица из двух столбцов, содержащая все фильмы и их продолжительность
  2. Ошибка
  3. Таблица из двух столбцов, содержащая записи с теми фильмами, продолжительность которых от полутора до двух часов
  4. Таблица из двух столбцов, содержащая записи с теми фильмами, сгруппированные по продолжительности и только те, в которых продолжительность фильмов от полутора до двух часов
9. Для чего используется оператор SELECT?
  1. Для указания из какой таблицы выбираем данные
  2. Для указания, какие столбцы выбираем из таблицы
  3. Для группировки результатов выборки
  4. Для сортировки результатов выборки
10. Какое ключевое слово используется для управления дубликатами при добавлении данных?
  1. INSERT
  2. UPDATE
  3. ON CONFLICT
  4. DEFAULT

#### Тест к уроку 8.1. «Базовые команды в bash»

1. Что такое Bash?
  1. Усовершенствованная командная оболочка
  2. Командная оболочка
  3. Тоже самое, что SH.
  4. Типизированный язык программирования
2. Команда для просмотра текущей активности директории
  1. Ls
  2. Dir
  3. Echo pwd
  4. Pwd
3. Команда для просмотра содержимого текущей директории?
  1. Pwd
  2. Echo pwd

3. Ls
4. Echo Ls
4. Выберите неверное утверждение
  1. Работать в Bash можно и в Linux, и в Windows
  2. Работать с Bash можно в PowerShell
  3. Работать с Bash можно в Git Bash
  4. Работать в Bash можно только в Linux
5. Команда для перемещения по каталогам?
  1. Dvd
  2. Cd rom
  3. Dvd rom
  4. Cd

Тест к уроку 8.2. «Директории: создание, просмотр и редактирование директорий»

1. Команда для копирования файлов?
  1. Mxcopy
  2. Mkdir
  3. Copy
  4. Cp
2. Команда для создания директории?
  1. Pwd
  2. Echo pwd
  3. mkdir
  4. Ls
3. Выберите альтернативы команде Get-Children -Force
  1. gci -Force
  2. dir Force
  3. ls -Force
  4. Все вышеперечисленное
4. Что сделает команда cp -r dir1/\* dir3?
  1. Скопирует только директории
  2. Ничего, выдаст ошибку
  3. Скопирует все файлы из директории dir1 в директорию dir3
  4. Скопирует все файлы и директории из директории dir1 в dir3

Тест к уроку 8.3. «Файлы: создание, просмотр, редактирование и копирование»

1. Какой команды для создания файла не существует?
  1. touch
  2. cat
  3. echo
  4. createFile
2. Что делает команда cat 4.txt 5.txt > 6.txt?
  1. Возьмет все самое лучшее из первого файла и запишет в третий
  2. Об этом знает только cat
  3. Считает спецсимволы из двух файлов и запишет в третий
  4. Считает текст из первых двух файлов и запишет в новый файл
3. Что делает команда : > myFile.txt?
  1. Создаст непустой файл
  2. Продолжит выполнение предыдущей команды
  3. Ничего
  4. Создаст пустой файл с именем myFile.txt

Тест к уроку 8.4. «Поиск по файлам и директориям»

1. Команда для вывода версии find?
  1. FIND --version
  2. find -version
  3. find --version
  4. Все вышеперечисленные
2. Команда для поиска файлов?
  1. Find file
  2. Search
  3. Find
  4. Findfile
3. Выберите неверное утверждение?
  1. Работать в Bash можно и в Linux, и в Windows
  2. Работать с Bash можно в PowerShell
  3. Работать с Bash можно в Git Bash
  4. Работать в Bash можно только в Linux
4. Команда, которая найдет файл myFile.txt?
  1. find -iname "myfile"
  2. find -iname "myfile.TXT"
  3. find -name "myfile.txt"
  4. find -iname "myfile.txt"

Тест к уроку 8.5. «Ввод и вывод»

1. Какой параметр указывает на то, что можно использовать перенос строки в echo?
  1. N
  2. K
  3. M
  4. E
2. Что сделает данная команда? echo "myWord" >> 1.txt
  1. Запишет два слова my и Word в файл 1.txt
  2. Запишет слово myWord в файл 1.txt с его перезаписью
  3. Запишет слово myWord в файл 1.txt без его перезаписи
  4. Упадет с ошибкой
3. Что сделает данная команда? printf "one\ntwo\n" > 2.txt
  1. Ничего
  2. Запишет в файл 2.txt текст в одну строку
  3. Запишет в файл 2.txt текст с форматированием
  4. Запишет в файл 2.txt текст в две строки
4. Какая команда была рассмотрена для записи текста в несколько файлов?
  1. Cat
  2. Echo
  3. Printf
  4. Tee

Тест к уроку 8.6. «Работа с процессами»

1. Команда для просмотра текущих процессов?
  1. Ls
  2. Dir
  3. Echo process
  4. Ps
2. Команда для просмотра процессов в Windows?
  1. Ls
  2. Echo pwd

3. Tasklist
4. Ps
3. Выберите неверное утверждение?
  1. У процесса может быть несколько потоков
  2. Процесс — это программа, исполняющаяся в операционной системе
  3. Explorer.exe – это системный процесс
  4. Пользовательские процессы могут запускаться только в интерактивном режиме
4. Основные состояния процессов?
  1. Открыт, закрыт, на паузе
  2. Приостановлен, запущен, остановлен
  3. Выполняется, не выполняется
  4. Заблокирован, готов к выполнению, выполняется

Тест к уроку 8.7. «Удаленные ресурсы»

1. Какой SSH-клиент был рассмотрен в уроке?
  1. Kitty
  2. MobaXterm
  3. RDP
  4. Putty
2. Что такое Putty?
  1. Шпатлевка
  2. SSH-ключ
  3. SSH-клиент
  4. SSH-генератор
3. Как переводится SSH?
  1. Безопасная оболочка
  2. Командная оболочка
  3. Secure Shell
  4. Серьезная оболочка
4. В какой программе можно перетаскивать файлы при SSH-подключении?
  1. Putty
  2. Kitty
  3. RDP
  4. WinSCP
5. Какая команда позволяет работать с SSH через командную строку?
  1. WinSCP
  2. Echo
  3. Putty
  4. SSH

Тест к теме 8. «Командная строка»

1. Что такое Bash?
  1. Усовершенствованная командная оболочка
  2. Командная оболочка
  3. Тоже самое, что SH.
  4. Типизированный язык программирования
2. Команда для просмотра содержимого текущей директории?
  1. Pwd
  2. Echo pwd
  3. Ls
  4. Echo Ls

3. Команда для копирования файлов?
  1. Мкcopy
  2. Mkdir
  3. Copy
  4. Cp
4. Выберите альтернативы команде Get-Children -Force
  1. gci -Force
  2. dir Force
  3. ls -Force
  4. Все вышеперечисленное
5. Какой команды для создания файла не существует?
  1. touch
  2. cat
  3. echo
  4. createFile
6. Что сделает команда : > myFile.txt?
  1. Создаст непустой файл
  2. Продолжит выполнение предыдущей команды
  3. Ничего
  4. Создаст пустой файл с именем myFile.txt
7. Команда для поиска файлов?
  1. Find file
  2. Search
  3. Find
  4. Findfile
8. Команда, которая найдет файл myFile.txt?
  1. find -iname "myfile"
  2. find -iname "myfile.TXT"
  3. find -name "myfile.txt"
  4. find -iname "myfile.txt"
9. Что сделает данная команда? echo "myWord" >> 1.txt
  1. Запишет два слова my и Word в файл 1.txt
  2. Запишет слово myWord в файл 1.txt с его перезаписью
  3. Запишет слово myWord в файл 1.txt без его перезаписи
  4. Упадёт с ошибкой
10. Какая команда была рассмотрена для записи текста в несколько файлов?
  5. Cat
  6. Echo
  7. Printf
  8. Tee
11. Команда для просмотра процессов в Windows?
  1. Ls
  2. Echo pwd
  3. Tasklist
  4. Ps
12. Основные состояния процессов?
  1. Открыт, закрыт, на паузе
  2. Приостановлен, запущен, остановлен
  3. Выполняется, не выполняется
  4. Заблокирован, готов к выполнению, выполняется

13. Какой SSH-клиент был рассмотрен в уроке?

1. Kitty
2. MobaXterm
3. RDP
4. Putty

14. Как переводится SSH?

1. Безопасная оболочка
2. Командная оболочка
3. Secure Shell
4. Серьезная оболочка

15. Какая команда позволяет работать с SSH через командную строку?

1. WinSCP
2. Echo
3. Putty
4. SSH

Тест к уроку 9.1. «Что такое Git: установка и настройка»

1. Что такое VCS?

1. Клубная Видео система (Video Club System)
2. Система контроля версий (Version Control System)
3. Виртуальная система кластеров (Virtual Clustering System)
4. Истории города Vice City (Vice City Stories)

2. К какому типу VSC относится Git?

1. централизованная система
2. децентрализованная система

3. Какой командой можно создать репозиторий в локальном окружении?

1. pull
2. clone
3. init
4. push

4. Какой командой можно добавить файл в репозиторий?

1. clone
2. add
3. pull
4. init

5. Содержимое какого ключа нужно добавить в удаленную систему для возможности подключения по SSH?

1. закрытого, открытый должен остаться локально у пользователя. При подключении, удалённый сервер шифрует сообщение закрытым ключом и отправляет его пользователю, который своим открытым ключом его расшифровывает.
2. открытого, закрытый ключ должен остаться локально у пользователя. При подключении, удалённый сервер шифрует сообщение открытым ключом и отправляет его пользователю, который своим закрытым ключом его расшифровывает.
3. закрытого и открытого. Пара ключей должна быть на сервере и локально у пользователя. При подключении сервер сравнивает ключи и проверяет доступ.

Тест к уроку 9.2. «Основные команды Git»

1. Какой командой можно посмотреть текущее состояние локального репозитория?

1. git show
2. git log



3. git status
4. git local show
2. Вы внесли изменение в файл. Какой командой можно зафиксировать изменения в репозитории?
  1. git add
  2. git commit -m
  3. git commit -a
  4. git push
3. Является ли файл .gitignore частью репозитория?
  1. да
  2. нет
4. Какой командой можно отметить еще не зафиксированные изменения?
  1. git reset
  2. git rm
  3. git restore
  4. git remote
5. Какой параметр является синонимом параметра --staged?
  1. --saved
  2. --cached
  3. --logged
  4. --fetched

#### Тест к уроку 9.3. «Ветки»

1. Какой командой можно создать ветку?
  1. git branch <имя ветки>
  2. git switch --create <имя ветки>
  3. git checkout -b <имя ветки>
  4. верны все варианты выше
2. Что будет выведено в результате выполнения команды git branch?
  1. список веток локального репозитория без ссылок на ветки на удалённом репозитории
  2. список веток локального репозитория включая ссылки на ветки на удалённом репозитории
  3. информация о состоянии текущей ветки
  4. сообщение о нехватке параметров для выполнения команды
3. Какой командой можно передать локальную ветку на удалённый сервер?
  1. git push
  2. git push <имя репозитория> <имя ветки>
  3. git fetch <имя ветки>
  4. git pull
4. Какой порядок действий при слиянии ветки с основной веткой разработки?
  1. git merge main  
git push
  2. git switch main  
git merge <branch\_name>  
git push
  3. git switch main  
git pull  
git merge <branch\_name>  
git push
  4. git pull

```
git merge main
git push
```

5. Какой порядок действий при переименовании ветки, ранее опубликованной на удалённом репозитории?

1. git branch --move <old\_branch\_name> <new\_branch\_name>  
git push --set-upstream <remote\_server> <new\_branch\_name>
2. git branch --move <old\_branch\_name> <new\_branch\_name>  
git push <remote\_server> --delete <old\_branch\_name>
3. git branch --move <old\_branch\_name> <new\_branch\_name>  
git branch --delete <old\_branch\_name>
4. git branch --move <old\_branch\_name> <new\_branch\_name>  
git push --set-upstream <remote\_server> <new\_branch\_name>  
git push <remote\_server> --delete <old\_branch\_name>

|                                                                    |          |                                                        |          |
|--------------------------------------------------------------------|----------|--------------------------------------------------------|----------|
| git merge main<br>git push                                         | <b>1</b> | git switch main<br>git merge <branch_name><br>git push | <b>2</b> |
| git switch main<br>git pull<br>git merge <branch_name><br>git push | <b>3</b> | git pull<br>git merge main<br>git push                 | <b>4</b> |

|                                                                                                                                                                           |          |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| git branch --move <old_branch_name> <new_branch_name><br>git push --set-upstream <remote_server> <new_branch_name>                                                        | <b>1</b> |
| git branch --move <old_branch_name> <new_branch_name><br>git push <remote_server> --delete <old_branch_name>                                                              | <b>2</b> |
| git branch --move <old_branch_name> <new_branch_name><br>git branch --delete <old_branch_name>                                                                            | <b>3</b> |
| git branch --move <old_branch_name> <new_branch_name><br>git push --set-upstream <remote_server> <new_branch_name><br>git push <remote_server> --delete <old_branch_name> | <b>4</b> |

Тест к уроку 9.4. «UI Tools в Git»

1. Какое приложение является интегрированным приложением операционной системы?

1. Git Kraken
2. SourceTree
3. Git GUI
4. TortoiseGit

2. Какое клиентское приложение устанавливается по-умолчанию с Git?

1. Git Kraken
2. Git GUI
3. Git Extensions
4. SourceTree

3. Требуется ли специальная установка клиентского приложения для работы с git репозиториями при использовании интегрированных сред разработки?

1. да
2. нет

4. Какое приложение обычно используется для подключения по SSH в операционной системе Windows?

1. Ping
2. Putty
3. WinSSH
4. SSHGit

Тест к уроку 9.5. «Командная работа в Git»

1. Какого подхода к командной работе с репозиторием не существует?

1. GitHub Flow
2. Trunk Based Development
3. GitFlow
4. Test Branch Development

2. В каком подходе используется несколько основных веток для ведения разработки?

1. GitHub Flow
2. GitFlow
3. Trunk Based Development

3. Обязана ли команда использовать один из существующих подходов в своей работе?

1. Да
2. Нет

4. От чего зависит создание релизной ветки в Trunk Base Development?

1. От размера команды
2. От скорости разработки команды
3. От требуемой частоты релизов
4. От желания инженеров

5. Из какой ветки создаётся hotfix ветка в GitFlow?

1. develop
2. release
3. master
4. feature

6. В какую ветку вливается релизная ветка в GitFlow?

1. master и develop
2. develop и hotfix
3. master
4. feature и develop

Тест к уроку 9.6. «Использование Git для решения рабочих задач тестировщика»

1. Кому не нужно использовать git репозиторий в своей работе?

1. инженеру по тестированию
2. цветочнику
3. программисту
4. тимлиду

2. Обязательно ли использовать при работе с репозиторием какой-то известный подход?

1. Да, потому что они проверены многими командами
2. Нет, можно придумать свой велосипед

3. Могут ли инженеры по функциональному тестированию участвовать в code review?

1. да, могут, с целью изучения внутренней программной структуры продукта
2. да, могут, с целью анализа качества тестовых данных для модульных тестов и их покрытия
3. да, могут, с целью анализа кода для улучшения и уточнения своих функциональных тест-кейсов
4. верно всё вышеперечисленное

4. Как репозиторий помогает во взаимодействии в команде?
  1. репозиторий является местом, вокруг которого разворачиваются процессы создания продукта и формируются правила взаимодействия команды
  2. репозиторий является местом, в которое все отправляют свои результаты работы над продуктом, в независимости от принятых командных правил
  3. репозиторий является хранилищем кода команды
  4. репозиторий является средством обмена кодом между разработчиками
5. Можно ли использовать репозиторий для личного создания программного продукта?
  1. Нет, это бессмысленно, потому что нет команды
  2. Да, т.к. возможности репозитория, позволяют повысить эффективность создания продукта, даже если ты один в команде

Тест к теме 9. «Git»

1. Что такое VCS?
  1. Клубная Видео система (Video Club System)
  2. Система контроля версий (Version Control System)
  3. Виртуальная система кластеров (Virtual Clustering System)
  4. Истории города Vice City (Vice City Stories)
2. Какой командой можно создать репозиторий в локальном окружении?
  1. pull
  2. clone
  3. init
  4. push
3. Какой командой можно посмотреть текущее состояние локального репозитория?
  1. git show
  2. git log
  3. git status
  4. git local show
4. Является ли файл .gitignore частью репозитория?
  1. Да
  2. Нет
5. Что будет выведено в результате выполнения команды git branch?
  1. список веток локального репозитория без ссылок на ветки на удалённом репозитории
  2. список веток локального репозитория включая ссылки на ветки на удалённом репозитории
  3. информация о состоянии текущей ветки
  4. сообщение о нехватке параметров для выполнения команды
6. Какой порядок действий при переименовании ветки, ранее опубликованной на удалённом репозитории?
  1. `git branch --move <old_branch_name> <new_branch_name>`  
`git push --set-upstream <remote_server> <new_branch_name>`
  2. `git branch --move <old_branch_name> <new_branch_name>`  
`git push <remote_server> --delete <old_branch_name>`
  3. `git branch --move <old_branch_name> <new_branch_name>`  
`git branch --delete <old_branch_name>`
  4. `git branch --move <old_branch_name> <new_branch_name>`  
`git push --set-upstream <remote_server> <new_branch_name>`  
`git push <remote_server> --delete <old_branch_name>`
7. Какое клиентское приложение устанавливается по-умолчанию с Git?
  1. Git Kraken

2. Git GUI
3. Git Extensions
4. SourceTree
8. Какое приложение обычно используется для подключения по SSH в операционной системе Windows?
  1. Ping
  2. Putty
  3. WinSSH
  4. SSHGit
9. В каком подходе используется несколько основных веток для ведения разработки?
  1. GitHub Flow
  2. GitFlow
  3. Trunk Based Development
10. Из какой ветки создаётся hotfix ветка в GitFlow?
  1. develop
  2. release
  3. master
  4. feature
11. Обязательно ли использовать при работе с репозиторием какой-то известный подход?
  1. Да, потому что они проверены многими командами
  2. Нет, можно придумать свой велосипед
12. Как репозиторий помогает во взаимодействии в команде?
  1. Репозиторий является местом, вокруг которого разворачиваются процессы создания продукта и формируются правила взаимодействия команды
  2. Репозиторий является местом, в которое все отправляют свои результаты работы над продуктом, в независимости от принятых командных правил
  3. Репозиторий является хранилищем кода команды
  4. Репозиторий является средством обмена кодом между разработчиками

#### Итоговое тестирование

1. Что из этого является примером функциональных требований?
  1. Отчет №38 должен формироваться за 5 минут
  2. На форме заявки на кредит у каждого поля ввода должна быть всплывающая подсказка
  3. Интерфейс страницы Личного кабинета соответствует макету №4
2. Что такое тест-кейс?
  1. Алгоритм действий, которые требуется совершить для проверки работы программы
  2. Графическое представление идей и концепций
  3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении
  - 4.Arteфакт, который описывает совокупность конкретных шагов, условий и параметров, необходимый для проверки реализации тестируемой функциональности или ее части
3. Что такое эвристика?
  1. Графическое представление идей и концепций
  2. Использование предыдущего опыта и информации о вероятности наступления различных событий
  3. Базовый инструмент тестирования, представляющий список пунктов с возможностью поставить отметку об их выполнении

4. Совокупность правил и приемов, которые помогают запоминать нужные сведения и данные
4. Что из перечисленного обязательно должно быть в ТЗ?
  1. Основные положения, цель и назначение ПО, стоимость разработки
  2. Основные положения, цель и назначение ПО, требования, личные контакты всех, кто составлял ТЗ
  3. Основные положения, требования к ПО, состав и содержание работ, порядок контроля и приемки
  4. Подробное описание каждой функциональности с используемыми технологиями
5. Почему техника попарного тестирования считается достаточной?
  1. Потому что нам сложно сделать таблицу для трех и более параметров, учитывая все комбинации
  2. Потому что более опытные тестировщики так считают
  3. Потому что именно ошибки, которые возникают при взаимодействии двух параметров являются наиболее критичными
6. Исследовательское тестирование — это техника тестирования, которая...
  1. Включает в себя только автоматические тесты, не требующие участия тестировщиков
  2. Используется только для тестирования мобильных приложений.
  3. Позволяет выявить все возможные дефекты приложения с помощью тест-кейсов.
  4. Включает в себя исследование приложения с целью выявления потенциальных проблем и дефектов без заранее определенных тест-кейсов.
7. Что можно отнести к элементам метаинформации?
  1. тег `<input>`
  2. тег `<header>`
  3. `<!DOCTYPE>`
  4. тег `<meta>`
8. Какой селектор приведен в примере?
 

```
.article-title {
margin-bottom: 5px;
}
```

  1. Селектор идентификатора
  2. Селектор элемента
  3. Селектор класса
  4. Селектор идентификатора
9. Какой тег отвечает за перенос строки в HTML?
  1. `<p>`
  2. `<meta>`
  3. `<section>`
  4. `<br>`
10. Что такое ip компьютера?
  1. Инвентарный номер
  2. Адрес компьютера в магазине
  3. Адрес компьютера в сети
  4. Номер, по которому можно позвонить на компьютер
11. Как называется вкладка, где можно проверить скорость запросов?
  1. Элементы
  2. Консоль

3. Сеть
  4. Приложение
12. Как называется функция принудительного разрыва соединения?
1. Proxy
  2. Throttling
  3. Rewrite
  4. BreakPoint
13. Какой тип архитектуры описан?  
Сервер вынесен отдельно от клиентов. На нем хранятся и сразу же обрабатываются данные. И есть клиенты, на стороне которых представляются и обрабатываются данные.
1. Клиент-серверная
  2. Файл-серверная
  3. Монолитная
  4. Многозвенная
14. Как называется способ аутентификации через соц сеть?
1. SSO
  2. JWT
  3. OAuth
  4. Беспарольная аутентификация
15. Как называется главный элемент в синтаксисе XML?
1. Главный
  2. Основной
  3. Корневой
  4. Фундаментальный
16. Как называется утилита для озвучки контента в приложениях на iOS?
1. Talk Back
  2. Voice Over
17. Что такое чистый Android и как он связан с тестированием мобильных приложений на Android?
1. Это версия операционной системы Android без каких-либо модификаций или дополнений со стороны производителей устройств. Чистый Android использует только стандартные элементы и функции, которые предоставляются в операционной системе.
  2. Это версия операционной системы Android с множеством модификаций и дополнений от производителей устройств. Чистый Android не может использоваться для тестирования мобильных приложений.
  3. Чистый Android - это специальная версия операционной системы Android, которая используется только для тестирования мобильных приложений и не предназначена для ежедневного использования.
18. Какова роль эмуляторов в тестировании мобильных приложений?
1. Позволяют тестировать приложения на разных устройствах без наличия реальных устройств
  2. Заменяют реальные устройства для тестирования
  3. Позволяют ускорить тестирование приложений
19. Как начать сбор логов в Xcode?
1. Подключить устройство iOS к компьютеру, выбрать связанное с устройством приложение и нажать на кнопку "Open Console";
  2. Выбрать устройство iOS в качестве целевого устройства для отладки, запустить приложение и автоматически начнется сбор логов;

3. Создать новый проект в Xcode и задать настройки сбора логов.
20. Для чего используется тип данных *INT*?
  1. Для интеграционного ключа
  2. Для целых чисел
  3. Для текста
  4. Для внутренней отметки о текущем времени
21. Какой оператор используется для отображения полной первой таблицы, независимо от того, есть ли парное значение во второй?
  1. FULL OUTER JOIN
  2. LEFT JOIN
  3. JOIN
  4. RIGHT JOIN
22. Что будет результатом такого SQL-запроса? **DELETE FROM employees;**
  1. Удаляются только четные строки таблицы employees
  2. Удаляются только нечетные строки таблицы employees
  3. Отчет об ошибке
  4. Удаляются все строки таблицы employees
23. Команда для просмотра содержимого текущей директории?
  1. Pwd
  2. Echo pwd
  3. Ls
  4. Echo Ls
24. Команда для поиска файлов?
  1. Find file
  2. Search
  3. Find
  4. Findfile
25. Что сделает данная команда?  
`printf "one\ntwo\n" > 2.txt`
  1. Ничего
  2. Запишет в файл 2.txt текст в одну строку
  3. Запишет в файл 2.txt текст с форматированием
  4. Запишет в файл 2.txt текст в две строки
26. Команда для просмотра процессов в Windows?
  1. Ls
  2. Echo pwd
  3. Tasklist
  4. Ps
27. Как переводится SSH?
  1. Безопасная оболочка
  2. Командная оболочка
  3. Secure Shell
  4. Серьезная оболочка
28. К какому типу VSC относится Git?
  1. Централизованная система
  2. Децентрализованная система
29. Какой параметр является синонимом параметра `--staged`?
  1. `--saved`
  2. `--cached`
  3. `--logged`



4. `--fetch`
30. Какой командой можно передать локальную ветку на удалённый сервер?
1. `git push`
  2. `git push <имя репозитория> <имя ветки>`
  3. `git fetch <имя ветки>`
  4. `git pull`