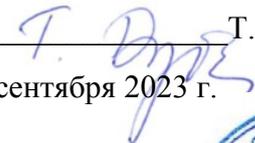


АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«УЧЕБНЫЙ ЦЕНТР СКБ КОНТУР»

Утверждаю
Директор АНО ДПО
«Учебный центр СКБ Контур»

 Т.В. Рубан
1 сентября 2023 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

дополнительная общеразвивающая программа

ТЕСТИРОВАНИЕ ПО. ПРОДВИНУТЫЙ УРОВЕНЬ

Документ: Свидетельство

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

Срок обучения: 17 недель

Объем программы: 174 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.....	24
4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ - ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ.....	25
4.1. Требования к квалификации педагогических кадров, представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса.....	25
4.2. Требования к материально-техническим условиям.....	25
4.3. Информационные и учебно-методические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	27
4.4. Общие требования к организации образовательного процесса.....	28
5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	29
5.1. Формы контроля знаний и требования к его проведению.....	29
5.2. Критерии оценки знаний слушателей.....	30
5.3. Фонд оценочных средств.....	32

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Назначение программы

Дополнительная общеобразовательная программа дополнительная общеразвивающая программа «Тестирование ПО. Продвинутый уровень» направлена на получение практических навыков и теоретических знаний, необходимых для тех, кто хочет освоить навыки тестирования на продвинутом 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 знания и навыки тестировании ПО на продвинутом уровне.

Задачи дисциплины

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

- о современных технологиях и методах тестирования в работе;
- об особенностях разработки автотестов для тестирования API, mobile, WEB;
- об эффективном управлении командой;
- о том, как работать с данными с помощью языка программирования Python на продвинутом уровне.

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 – 17 недель.

2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Объем образовательной программы составляет 174 академических часа. Из них 86 - самостоятельная работа, 88 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:

- особенности тестирования;
- основы контроля качества;
- SQL на углубленном уровне и Python на продвинутом уровне;
- Docker, Selenium, PyTest, Agile-подход.

Уметь:

- проводить тестирование ПО;
- обеспечивать доступность (распространение) информации и опыта среди других участников команды;
- проводить анализ рисков перед тестированием для обеспечения качества;
- оптимизировать тестовые сценарии;
- принимать определенную ответственность за процесс или функциональность системы;
- эффективно планировать задачи и быстро давать оценки по тестированию.

Владеть:

- навыками ручного и автоматизированного тестирования API, мобильных и web-приложений;
- навыками работы с SQL и Python на продвинутом уровне;
- навыками эффективного управления командой.

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	Из них:		Форма контроля
			Самостоятельная работа	Работа на образовательной онлайн-платформе	
1.	Docker для тестировщика	16	8	8	Зачет
1.1	Введение в Docker	4	2	2	Тестирование
1.2	Основные команды Docker	4	2	2	Тестирование
1.3	Запуск сервисов с помощью Docker Compose	4	2	2	Тестирование
1.4	Разворачивание проекта Docker	4	2	2	Тестирование
2.	Углубленный SQL	44	22	22	Зачет
2.1	Введение в базы данных	4	2	2	Тестирование
2.2	Выборка и агрегация данных	4	2	2	Тестирование
2.3	Сортировка и группировка	4	2	2	Тестирование
2.4	Фильтрация данных	4	2	2	Тестирование
2.5	Фильтрация данных по тексту	4	2	2	Тестирование
2.6	Операторы вычисления	4	2	2	Тестирование
2.7	Объединение таблиц	4	2	2	Тестирование
2.8	Создание таблиц	4	2	2	Тестирование
2.9	Работа с данными	4	2	2	Тестирование
2.10	Использование подзапросов для фильтрации	4	2	2	Тестирование
2.11	Работа с комбинированными запросами	4	2	2	Тестирование
3.	Продвинутый Python	44	22	22	Зачет
3.1	Типы и структуры данных	4	2	2	Тестирование

					ие
3.2	Основные конструкции языка	4	2	2	Тестирован ие
3.3	Работа со строками	4	2	2	Тестирован ие
3.4	Работа со списками	4	2	2	Тестирован ие
3.5	Работа со словарями и коллекциями	4	2	2	Тестирован ие
3.6	Функции	4	2	2	Тестирован ие
3.7	Классы и объекты	4	2	2	Тестирован ие
3.8	Объектно-ориентированное программирование (ООП)	4	2	2	Тестирован ие
3.9	Модификаторы доступа, свойства и дескрипторы	4	2	2	Тестирован ие
3.10	Использование библиотек	4	2	2	Тестирован ие
3.11	Обработка исключений	4	2	2	Тестирован ие
4.	Углубленное изучение: автоматизация тестирования	32	16	16	Зачет
4.1	Как тестировать веб- приложения и API. Тестовые фреймворки	4	2	2	Тестирован ие
4.2	Тестирование с помощью Selenium	4	2	2	Тестирован ие
4.3	Тестирование с помощью PyTest	4	2	2	Тестирован ие
4.4	Использование фикстур в PyTest	4	2	2	Тестирован ие
4.5	PyTest	4	2	2	Тестирован ие
4.6	Автотесты на мобильные приложения на Python	4	2	2	Тестирован ие
4.7	Автоматизация тестирования API	4	2	2	Тестирован ие
4.8	Паттерн Page Object Model	4	2	2	Тестирован ие
5.	Управление командой	20	10	10	Зачет
5.1	Кто такой лид команды и как им стать	4	2	2	Тестирован ие
5.2	Приоритизация задач и умение	4	2	2	Тестирован

	сказать - Нет				ие
5.3	Постановка целей команде и контроль за исполнением	4	2	2	Тестирование
5.4	Развитие и мотивация команды. Обратная связь	4	2	2	Тестирование
5.5	Решение конфликтов	4	2	2	Тестирование
6.	Agile в командах	16	8	8	Зачет
6.1	Что такое Agile - подход и зачем он нужен бизнесу	4	2	2	Тестирование
6.2	Роль Scrum - мастера в команде	4	2	2	Тестирование
6.3	Насколько Agile влияет на удовольствие от работы	4	2	2	Тестирование
6.4	Анти-Паттерны Agile - команд	4	2	2	Тестирование
ИТОГОВАЯ АТТЕСТАЦИЯ		2	–	2	Зачет
Всего:		174	86	88	–

* Дополнительная общеразвивающая программа «Тестирование ПО. Продвинутый уровень» находится в разработке, в связи с этим возможны изменения в учебном плане, в том числе в наименовании разделов и тем уроков.

3.2. Содержание*

Тема 1. Docker для тестировщика

1.1. Введение в Docker

- Что такое Docker
- Что такое Docker image («образ»)
- Как работает Docker

- Чем может быть полезен Docker
- Назначение Docker
- Практическое задание:

Нужно зарегистрироваться на docker hub: <https://hub.docker.com/>

Затем установить себе Docker Desktop: <https://www.docker.com/>

Прочитать документацию: <https://docs.docker.com/reference/>

Гайд для установки Docker Desktop

1.2. Основные команды Docker

- Основные команды Docker
- Реестры и репозитории Docker
- Работа с контейнерами: первые действия с контейнерами; запуск и остановка

контейнеров; получение информации о контейнерах

- Управление образами: Docker-образ (image). Что входит в управление образами.

Рекомендации по управлению образами. Инструменты управления

- Работа с сетью: Мостовые сети. Хост-сети. Оверлейные сети. Сети Macvlan
- Практическое задание: Запустите 2 контейнера из образов MySQL и PHPmyadmin.

1.3. Запуск сервисов с помощью Docker Compose

- Определение Docker Compose
- Для чего нужен Docker Compose
- Как работает Docker Compose
- Service в Docker Compose
- Networks
- Volumes (тома)
- Console
- Ключевые моменты
- Практическое задание: создайте и запустите файл docker compose.

1.4. Разворачивание проекта Docker

- Почему стоит разворачивать проект именно в Docker
- План разворачивания проекта
- Настройка среды Docker
- Создание Dockerfile
- Отправка dockerfile в репозиторий
- Развертывание контейнера Docker в рабочей среде
- Ключевые моменты
- Практическое задание: Разверните собственный проект. Это может быть любая

БД, может быть сайт-визитка. Возьмите за основу лекции и гайды и используйте все шаги (для полного погружения можно переустановить Docker), чтобы развернуть свой проект в Docker.

Тема 2. Углубленный SQL

2.1. Введение в базы данных

- Что такое базы данных
- Свойства базы данных

- Виды баз данных и их характеристики: иерархические, сетевые, объектно-ориентированные, реляционные, нереляционные, документоориентированные, графовые, столбчатые базы данных.

- Что такое SQL

- Основные элементы базы данных

- Основные операторы построения самих запросов

- Практическое задание: Представьте, что вам нужно составить базу данных ваших одноклассников. Напишите, какой вид базы данных вы для этого выберете. Аргументируйте свой ответ 1–2 предложениями. Напишите, какие атрибуты вы бы добавили в свою БД. Количество атрибутов — не более 10.

2.2. Выборка и агрегация данных

- Выборка и агрегация данных: SELECT FROM

- Основные определения и понятия

- Операторы выборки и агрегации данных

- Операторы SELECT FROM

- Выборка и агрегация данных: WHERE

- Основные определения и понятия

- Операторы выборки и агрегации данных

- Оператор WHERE

- Выборка и агрегация данных: GROUP BY

- Основные определения и понятия

- Операторы выборки и агрегации данных

- Оператор GROUP BY

- Выборка и агрегация данных: ORDER BY

- Основные определения и понятия

- Операторы выборки и агрегации данных

- Оператор ORDER BY

- Выборка и агрегация данных: HAVING

- Основные определения и понятия

- Операторы выборки и агрегации данных

- Оператор HAVING

- Практическое задание: Перейдите по ссылке и выполните предлагаемое задание.

2.3. Сортировка и группировка

- Основные агрегатные функции:

- COUNT — определение количество строк.

- AVG — вычисление среднего значения.

- SUM — определение суммы значений.

- MIN — определение минимального значения.

- MAX — определение максимального значения.

- Как происходит группировка?

- Практическое задание: автопроверка кода на нашей платформе Ulearn.

Перейдите по этой или этой ссылке и выполните одно из предлагаемых заданий.

2.4. Фильтрация данных

- Что такое WHERE?

- Вспомогательный функционал WHERE

- Правила использования логических операторов
- Практическое задание: TOP_movies. Для данной таблицы создайте запрос на отображение всех строк, фильмы в которых были выпущены после 1995 года.

empld	name	genre	year	duration
1	Зеленая миля	драма	1999	189
2	Список Шиндлера	драма	1993	195
3	Побег из Шоушенка	драма	1994	142
4	Форрест Гамп	драма	1994	142
5	Тайна Коко	мультфильм	2017	105
6	Властелин колец: Возвращение короля	фэнтези	2003	201
7	Интерстеллар	фантастика	2014	169
8	Криминальное чтиво	криминал	1994	154
9	Бойцовский клуб	триллер	1999	139
10	Назад в будущее	фантастика	1985	116

2.5. Фильтрация данных по тексту

- Способы фильтрации текста в SQL-запросах
- Ключевые слова, используемые для проверки значений на соответствие шаблону
- Использование ключевых слов и шаблонов
- Практическое задание: TOP_MOVIES. Для данной таблицы создайте запрос на отображение всех строк, фильмы в которых были выпущены в 1990-х годах. В своем запросе используйте ключевое слово LIKE.

empld	name	genre	year	duration
1	Зеленая миля	драма	1999	189
2	Список Шиндлера	драма	1993	195
3	Побег из Шоушенка	драма	1994	142
4	Форрест Гамп	драма	1994	142
5	Тайна Коко	мультфильм	2017	105
6	Властелин колец: Возвращение короля	фэнтези	2003	201
7	Интерстеллар	фантастика	2014	169
8	Криминальное чтиво	криминал	1994	154
9	Бойцовский клуб	триллер	1999	139
10	Назад в будущее	фантастика	1985	116

2.6. Операторы вычисления

- Что такое «операторы вычисления»?
- Основные операторы
- Примеры с операторами вычисления

Практическое задание: Furniture. Для данной таблицы создайте запрос на отображение всех строк, в которых общая стоимость предмета мебели и его доставки больше 20 000.

Id	name	price	delivery_price
1	Стул	2000	100
2	Стол	12000	500
3	Диван	19000	2000
4	Кровать	21000	2000
5	Шкаф	20000	1000

2.7. Объединение таблиц

- Оператор JOIN
- Особенности использования JOIN

- Примеры использования JOIN
- Правила для объединения таблиц: INNER JOIN, LEFT JOIN, RIGHT JOIN, OUTER JOIN.

- Присоединение таблиц: INNER JOIN

- Основные определения и понятия
- Оператор присоединения таблиц
- Режим INNER JOIN

- Присоединение таблиц: OUTER JOIN

- Основные определения и понятия
- Оператор присоединения таблиц
- Режим OUTER JOIN

- Присоединение таблиц: LEFT и RIGHT OUTER JOIN

- Основные определения и понятия
- Оператор присоединения таблиц
- Режим LEFT OUTER JOIN
- Режим RIGHT OUTER JOIN

- Присоединение таблиц: CROSS JOIN

- Основные определения и понятия
- Оператор присоединения таблиц
- Режим CROSS JOIN

- Практическое задание: Для отработки составления запросов на объединение таблиц мы подготовили задания с автопроверкой кода на нашей платформе Ulearn. Перейдите по ссылке и выполните предлагаемое задание.

2.8. Создание таблиц

- CREATE TABLE— команда для создания таблиц.
- Основные типы данных, используемых в базах.
- Пример создания таблицы
- Редактирование таблиц
- Практическое задание: Для отработки составления запросов на создание

таблиц мы подготовили задания с автопроверкой кода на нашей платформе Ulearn. Перейдите по ссылке и выполните предлагаемое задание.

2.9. Работа с данными

- Работа с данными: INSERT
 - Основные определения и понятия
 - Оператор для работы с данными
 - Оператор INSERT
- Работа с данными: UPDATE
 - Основные определения и понятия
 - Оператор для работы с данными
 - Оператор UPDATE
- Работа с данными: DELETE
 - Основные определения и понятия
 - Оператор для работы с данными
 - Оператор DELETE

- Практическое задание. Для отработки составления запросов на работу с данными мы подготовили задания с автопроверкой кода на нашей платформе Ulearn. Перейдите по ссылке и выполните предлагаемое задание.

2.10. Использование подзапросов для фильтрации

- Что такое подзапрос
- Способы ввода подзапросов в выражении SELECT
- Использование ключевых слов
- Практическое задание. По данным таблицам создайте запрос для полного объединения обеих таблиц по идентификационному номеру жанра. В запросе используйте ключевое слово UNION.

1. Top_movies

id	name	genre_id	year	duration
1	Зеленая миля	1	1999	189
2	Список Шиндлера	1	1993	195
3	Побег из Шоушенка		1994	142
4	Форрест Гамп	1	1994	142
5	Тайна Коко	2	2017	105
6	Властелин колец: В	3	2003	201
7	Интерстеллар	4	2014	169
8	Криминальное чтиво	5	1994	154
9	Бойцовский клуб		1999	139
10	Назад в будущее	4	1985	116

2. Genres

id	name
1	драма
2	мультфильм
3	фэнтези
4	фантастика
5	криминал
6	триллер

2.11. Работа с комбинированными запросами

- UNION — оператор для комбинирования запросов в языке SQL
- Правила написания комбинированных запросов
- Практическое задание. Для отработки составления запросов с использованием подзапросов мы подготовили задания с автопроверкой кода на нашей платформе Ulearn. Перейдите по ссылке и выполните предлагаемое задание.

Тема 3. Продвинутый Python

3.1. Типы и структуры данных

- Типы данных в языке Python
 - Вступление
 - Числовые типы данных
 - Целые
 - Числа с плавающей точкой
 - Комплексные числа
 - Строки
 - Типы данных логических значений
 - Шпаргалка по типам
- Структуры данных в языке Python
 - Вступление

- Списки
- Кортежи
- Словари
- Множества
- Шпаргалка по структурам данных
- Практическое задание.

Задание 1. Напишите программу, которая запрашивает у пользователя его имя, возраст, рост и вес, а затем выводит на экран информацию о пользователе в следующем формате:

```
"Привет, [имя]! Твой возраст [возраст] лет, твой рост [рост] см, твой вес [вес] кг."
```

При написании программы используйте различные типы данных, такие как строки, целые и дробные числа. Не забудьте о преобразовании типов данных, если это необходимо.

Задание 2. Напишите программу, которая принимает на вход список из 10 случайных целых чисел в диапазоне от 1 до 100 и использует алгоритм сортировки (например, сортировку пузырьком), чтобы отсортировать этот список в порядке возрастания.

Затем программа должна вывести отсортированный список на экран.

Пример входных данных:

```
[56, 22, 87, 19, 4, 75, 36, 91, 38, 50]
```

Пример выходных данных:

```
[4, 19, 22, 36, 38, 50, 56, 75, 87, 91]
```

3.2. Основные конструкции языка

- Переменные, константы и оператор присваивания в Python
 - Введение
 - Переменные
 - Константы
 - Оператор присваивания
- Условный оператор в языке Python
 - Условный оператор if
 - Условный оператор if-else
 - Условный оператор if-elif-else
 - Оператор условного выражения (тернарный оператор)
- Циклы в языке Python
 - Цикл for
 - Цикл while
- Практическое задание.

Задание 1.

1. Создайте программу на Python, которая будет вычислять площадь круга.
 2. Задайте значение радиуса круга в переменную и используйте оператор присваивания для присвоения значения.
 3. Используйте константу для хранения значения числа π .
 4. Используйте формулу $S = \pi r^2$ для вычисления площади круга, где S — площадь, r — радиус.
 5. Выведите результат на экран с помощью функции `print()`.
- Задание 2. Напишите функцию `sale_pirojki`, функция принимает 1 параметр: n .

n — количество пирожков, которые покупатель купит. Разные числа имеют разные цены (см. следующую таблицу). Возвращает, сколько денег покупатель потратит, чтобы купить это количество пирожков.

Количество пирожков	Цена за штуку
$n < 5$	100
$n \geq 5$ and $n < 10$	95
$n \geq 10$	90

Задание 3. Напишите функцию, которая принимает число n и в ответ отдает сумму чисел от 1 до n включительно. Например, $n = 5$, следовательно, функция должна вернуть $1 + 2 + 3 + 4 + 5 = 15$.

3.3. Работа со строками

- Операции со строками
 - Что такое строки и зачем они нужны
 - Литералы строк
 - Индексация и доступ к символам строки
 - Операции с использованием строк
 - Функции и методы строк
- Практическое задание.

Напишите программу, которая считает количество слов в тексте, введенном пользователем, и выведет на экран их количество.

Пример работы программы:

```
Введите текст: Написать программу, которая считает количество слов в тексте
Количество слов в тексте: 8
```

3.4. Работа со списками

- Операции со списками
 - Списки в Python
 - Списки упорядочены
 - Списки могут содержать произвольные объекты
 - Индексация списков
 - Вложенность списков
 - Методы списков
- Практическое задание

Напишите программу, которая принимает на вход список чисел и выводит на экран два списка: список четных чисел и список нечетных чисел из входного списка. Пример использования:

```
Введите числа через пробел: 1 2 3 4 5 6
Список четных чисел: [2, 4, 6]
Список нечетных чисел: [1, 3, 5]
```

3.5. Работа со словарями и коллекциями

- Операции со словарем

- Определение словаря
- Доступ к значениям словаря
- Индексация в словарях
- Ограничения на ключи
- Методы словарей
- Операции с коллекцией
- Определение коллекции
- set
- dict
- list
- tuple
- Практическое задание.

Задание 1. Напишите программу, которая будет работать с информацией о книгах в библиотеке.

1. Создайте пустой словарь library.
2. Заполните словарь книгами, используя функцию add_book(title, author, year, genre), которая принимает параметры: название книги (title), автор (author), год издания (year) и жанр (genre). Функция должна добавлять книгу в словарь library в виде пары «ключ — значение», где ключом будет название книги, а значением будет словарь с информацией о книге (автор, год издания и жанр).
3. Напишите функцию print_books(), которая будет выводить на экран все книги, которые есть в словаре library в формате «Название книги — Автор (Год издания, жанр)».

Результат выполнения программы:

```

Список книг в библиотеке:
Война и мир - Л. Толстой ( 1869 , Роман )
Преступление и наказание - Ф. Достоевский ( 1866 , Роман )
Мастер и Маргарита - М. Булгаков ( 1967 , Роман )
1984 - Дж. Оруэлл ( 1949 , Антиутопия )

```

Задание 2. Необходимо написать функцию, на вход которой поступает список целых чисел, и которая должна вернуть новый список, содержащий только четные целые числа, отсортированные в порядке возрастания.

Для примера список может быть таким:

```
lst = [3, 7, 12, 5, 14, 8, 9, 2]
```

На выходе функции список должен стать таким:

```
[2, 8, 12, 14]
```

Пример использования:

```

lst = [3, 7, 12, 5, 14, 8, 9, 2]
sort_evens(lst)
print(lst)
[2, 8, 12, 14]

```

3.6. Функции

- Использование функций
- Определение функции
- Важность функций
- Вызовы функций и их определение
- Передача аргументов в функцию
- Возвращаемое значение

- Практическое задание.

Напишите программу-калькулятор, которая будет выполнять математические операции (+, -, *, /) в зависимости от выбора пользователя и выводить результат. Для реализации программы используйте функции. Пример работы программы:

```

Выберите операцию, которую хотите выполнить:
+ - сложение двух чисел
- - вычитание двух чисел
* - умножение двух чисел
/ - деление двух чисел
Введите знак операции: +
Введите первое число: 5
Введите второе число: 3.6
5.0 + 3.6 = 8.6
Выберите операцию, которую хотите выполнить:
+ - сложение двух чисел
- - вычитание двух чисел
* - умножение двух чисел
/ - деление двух чисел
Введите знак операции: /
Введите первое число: 6
Введите второе число: 0
На ноль делить нельзя!

```

3.7. Классы и объекты

- Классы и объекты как понятия
 - Все есть объект
 - Классы в Python
 - Объекты в Python
- Практическое задание
 - Измените стандартный объем стакана кофе на 250 мл. Добавьте возможность создать стакан сразу наполненным некоторым количеством кофе. Добавьте метод `drink`, позволяющий выпить некоторое количество кофе из стакана. Не забудьте, что количество кофе в стакане не бесконечно. Пусть статус возвращает информацию в литрах.

3.8. Объектно-ориентированное программирование (ООП)

- Что такое объектно-ориентированное программирование (ООП)?
- Основополагающие принципы ООП:
 - абстракция
 - инкапсуляция
 - наследование
 - полиморфизм
- Практическое задание.
 - Опишите абстракцию «пользователь»: выберите несколько полей и создайте класс для получившейся модели.

– Добавьте поле «Пароль» и методы для работы с ним: установить пароль, проверить пароль. Методы шифрования реализовывать не нужно, но помните про инкапсуляцию. Поле с паролем не должно быть публичным.

– Добавьте поле «Баланс». Используя наследование, создайте классы «Покупатель» и «Продавец».

– Создайте внешнюю функцию «Сделка» с параметром «Пользователь». При передаче в функцию баланс покупателя должен уменьшаться, а продавца увеличиваться. Попробуйте использовать полиморфизм для решения этой задачи.

3.9. Модификаторы доступа, свойства и дескрипторы

- Модификаторы доступа, свойства и дескрипторы
 - Модификаторы доступа
 - Дескрипторы
 - Свойства
 - Ключевые моменты
- Практическое задание
 - Создайте класс, моделирующий стакан кофе. Предусмотрите возможность нанести на стакан произвольную надпись. Длина текста не должна превышать 42 символов. Для валидации используйте дескриптор.
 - Создайте свойство для атрибута `_filled_ml`, определяющее наполненность стакана кофе. Не забудьте про валидацию.

3.10. Использование библиотек

- Модули и пакеты
- Виртуальное окружение
- Библиотеки
- Практическое задание:
 1. Создайте пакет Python, добавьте в него два вложенных пакета с модулями внутри. Создайте в модуле одного из пакетов класс. Попробуйте его импортировать в модуле второго пакета с помощью абсолютного и относительного импорта:

```
top_level_package
package_one
module - здесь создаём класс
package_two
module - здесь импортируем
```
 2. Создайте два виртуальных окружения и установите в них `pydantic 1.9` и `1.10`.
 3. Создайте классы `Moscow`, `Ekaterinburg`, `Vladivostok`. Свойство `local_time` должно возвращать текущее время в соответствующем городе. Воспользуйтесь библиотекой: <https://pypi.org/project/pytz/>

3.11. Обработка исключений

- Об исключениях
- Обработка исключений
- Дополнительные блоки
- Прокидывание исключений
- Практическое задание:
 1. Создайте свой класс исключения `ValidationError`, наследуемый от `ValueError`. Создайте его наследников: `TooSmallValue` и `TooBigValue`.
 2. Создайте класс `CupOfCoffee`, атрибут `size_ml` и свойство `filled_ml`. Добавьте валидацию значения свойства `filled_ml`: оно не должно превышать значение атрибута `size_ml` и быть меньше 0. Используйте при валидации созданные ранее классы исключений.

3. Создайте внешнюю функцию, которая будет заполнять значение свойства `filled_ml` и обрабатывать исключения.

Тема 4. Углубленное изучение: автоматизация тестирования

4.1. Как тестировать веб- приложения и API. Тестовые фреймворки

- Принципы автоматизации тестирования
 - Автоматизация тестирования
 - Этапы построения процесса автоматизации
 - Основные принципы автоматизации тестирования
- Тестовый фреймворк – Pytest
 - Что такое тестовый фреймворк
 - Что такое Pytest
 - Основы Pytest
- Практическое задание. Найдите 10 инструментов автоматизации тестирования и сделайте мини-обзор по каждому инструменту: какие плюсы, минусы, для чего используется.

4.2. Тестирование с помощью Selenium

- Автоматизация тестирования при помощи Selenium
- Установка и настройка Python и Selenium
- Написание теста
- Практическое задание. Напишите такой же selenium-тест на python, какой был разобран на уроке, только не для Google, а для Yandex.

4.3. Тестирование с помощью PyTest

- Подготовка окружения
- Тесты и проверки Pytest
- Практическое задание:
 - Напишите `assert` (покройте тестами) на следующую реализацию:

```
def sum(a, b):  
    return a - b
```

Проверки:

1. Вычитание двух чисел.
2. Вычитание отрицательных чисел.
3. Вычитание нечисла.

```
def sum(a, b):  
    return a - b
```

4.4. Использование фикстур в PyTest

- Фикстуры
 - Что такое фикстура?
 - Примеры использования фикстуры
 - Особенности использования фикстуры
 - Практическое задание:
 1. Создайте файл `_test.py` и напишите тест, который переходит на страницу <https://comments-school-1.testkontur.ru/comments/preview/1/default/test> и проверяет, что заголовок страницы «Превью комментариев».
 2. Напишите тест, который проверяет имя автора первого комментария и текст его комментария.
 3. Создайте файл `confest.py` и опишите фикстуру так, чтобы до выполнения теста выполнялось открытие браузера и переход на страницу, а после выполнения теста — закрытие браузера.
 4. Скопируйте код из файлов `_test.py` и `confest.py` и отправьте работу на проверку.

4.5. PyTest

- Параметризация тестов
 - Термины и определения
 - Создание функции
 - Создание параметризованного теста
- Маркировка тестов
 - Термины
 - Виды маркеров
 - Маркировка тестов
- Практическое задание:
 1. В параграфе «Параметризация тестов» были рассмотрены параметры типов данных: `boolean`, `numbers`, `strings`, `dictionaries`. Параметризируйте значения типов данных: списки (`lists`) и кортежи (`tuples`). Результат должен быть в виде кода на языке Python.
 2. Тестировщик-автоматизатор написал три теста: один — для разработчиков, два — для тестировщиков. Разделите тесты при помощи маркеров (`dev` и `test`), чтобы можно было запускать из командной строки тесты для тестировщиков командой `pytest devtest_test.py -k "dev"` и тест для разработчиков командой `pytest devtest_test.py -k "test"`. Результат должен быть в виде кода на языке Python.

4.6. Автотесты на мобильные приложения на Python

- Установка и настройка программ
- Написание и запуск теста под Android
- Практическое задание. Помимо теста на сложение чисел в калькуляторе, разобранного на уроке, реализуйте тесты на вычитание чисел, деление и умножение. Результат пришлите в виде кода, скопированного из среды разработки.

4.7. Автоматизация тестирования API

- Общие принципы тестирования API
 - Что такое API
 - Особенности взаимодействия с API
 - Процесс тестирования API
- Автоматизация тестирования API в Postman
 - Подготовка окружения
 - Тестирование API в Postman
- Автоматизация тестирования API на Python
 - Написание первого API-теста
 - Написание новых тестов
- Практическое задание:
 1. Создайте PUT-запрос в Postman на обновление любого фильма. Используйте документацию API и знания, полученные в уроке.
 2. Аналогично создайте PATCH-запрос в Postman на обновление любого фильма.
 3. Реализуйте тест на Python с PUT-запросом на обновление фильма.
 4. Реализуйте тест на Python с PATCH-запросом на обновление фильма.
 5. Результат пришлите в виде кода обоих тестов, скопированных из среды разработки.

4.8. Паттерн Page Object Model

- Виды паттернов проектирования в автоматизации тестирования
 - Что такое паттерн автоматизации тестирования
 - Паттерн проектирования в автоматизации тестирования
 - Паттерн проектирования Page Object
 - Паттерн проектирования Arrange Act Assert
 - Наименование теста. Параметризация тестов
 - Сборка и разборка окружения

- Паттерн проектирования ObjectMother
- Маркировка тестов
- Логирование. Отладка
- Реализация паттерна Page Object на Python
- Описание сценариев
- Написание тестов с использованием обычного подхода
- Написание тестов с применением паттерна Page Object
- Практическое задание:
 1. Повторите действия в уроке.
 2. Реализуйте более умное ожидание в тесте вместо `wait.sleep` (чтобы тест дождался появления элементов, которые могут не успевать загрузиться).

Тема 5. Управление командой

5.1. Кто такой лид команды и как им стать

- Команда и роли в проекте
- Обязанности тимлида
- Профессиональный профиль тимлида
- Личностные характеристики тимлида
- Как стать тимлидом
- Ключевые моменты
- Практическое задание. Напишите три качества или навыка, которые, на ваш взгляд, вам следует развить для подготовки к роли тимлида. Добавьте комментарии, почему вы выбрали именно эти качества/навыки и как планируете использовать их в работе. Объем комментария к каждому качеству/навыку — два предложения.

5.2. Приоритизация задач и умение сказать – Нет

- Приоритизация в проекте
 - Приоритизация задач
 - Практические способы приоритизации задач:
 1. Распределение функциональности по категориям
 2. Оценка задач по шкале «Усилия/влияние»
 3. Использование углубленного метода RICE
- Практическое задание:
 1. Представьте, что мы создаем приложение для обмена сообщениями, мессенджер. У вас есть список функциональностей, которые вы хотели бы реализовать. Рассчитайте приоритет каждой из них по методу RICE.
 2. Список функциональностей:
 3. Возможность устанавливать статус в своем профиле.
 4. Звуковое оповещение при получении нового сообщения.
 5. Графическая отметка о получении вашего сообщения оппонентом.

5.3. Постановка целей команде и контроль за исполнением

- Постановка целей команде и контроль за исполнением
- Постановка целей команде
- Стратегия для постановки цели — SMART
- Контроль за исполнением целей
- Практическое задание. Преобразуйте предложенные потребности в цели, соответствующие критериям SMART. Хочу больше продаж. Хочу новых пользователей. Хочу читать чаще и больше.

5.4. Развитие и мотивация команды. Обратная связь

- Мотивация и развитие команды
- Что такое развитие и мотивация
- Вопросы, которые можно задать сотрудникам
- Подходы, способствующие развитию и мотивации команды

- Практическое задание. Вы заметили, что ваш сотрудник работает гораздо хуже, чем раньше. Вы знаете, что задачи, которые у него в работе, не сложные для него. Вы наблюдали за ним и задавали вопросы его коллегам, но не смогли понять причину. Напишите, как вы поступите в этой ситуации. Ваша цель — добиться того, чтобы сотрудник вернулся к прошлой производительности или даже улучшил ее.

Максимальный объем работы — 1 500 символов.

5.5. Решение конфликтов

- Решение конфликтов
- Причины возникновения конфликтов
- Предотвращение конфликтов
- Решение конфликтов
- Ключевые моменты
- Практическое задание. Вы — тимлид проекта. Ваш более опытный сотрудник указывает менее опытному на его постоянно повторяющиеся ошибки. Менее опытный сотрудник воспринимает это на свой счет и обижается на коллегу. Опишите, что бы вы сделали в этой ситуации для остановки развития конфликта и скорейшего его разрешения. Максимальный объем — 1 000 символов.

Тема 6. Agile в командах

6.1. Что такое Agile - подход и зачем он нужен бизнесу

- Что такое Agile-подход и зачем он нужен бизнесу
 - Обзор моделей разработки программного обеспечения
 - Водопадная модель
 - V-образная модель
 - Инкрементно-итеративная модель
 - Agile-подход
 - Основополагающие принципы Agile-манифеста
 - Преимущества и недостатки Agile-подхода
- Практическое задание:
 1. Изучите основные функциональные элементы любого онлайн-магазина, такие как каталог товаров, корзина, оформление заказа, оплата и доставка товара.
 2. Разделите список требований на группы, относящиеся к каждому из функциональных элементов. Например, все требования, связанные с каталогом товаров, должны быть в одной группе.
 3. Для каждого элемента составьте список требований, используя шаблон User Story. Например: «Я, как покупатель, хочу иметь возможность фильтровать товары по цене, чтобы быстро найти подходящие мне товары».
 4. Для каждого требования установите приоритет. Это поможет определить, какие функциональные элементы должны быть выполнены в первую очередь.
 5. Оцените время и ресурсы, необходимые для реализации каждого требования. Это поможет вам понять, насколько реалистичны ваши требования и сколько времени и ресурсов понадобится на их выполнение.
 6. Определите, какая модель разработки ПО лучше всего подойдет для разработки проекта.

6.2. Роль Scrum - мастера в команде

- Роль Scrum-мастера в команде
 - Обзор гибких методологий разработки
 - Scrum
 - Kanban
 - Lean
 - Extreme Programming (XP)
 - Crystal
 - Что такое Scrum

- Роли в Scrum. Задачи Scrum-мастера
- Ритуалы в Scrum
- Практическое задание:
 1. Используйте требования, которые написаны для практического задания к уроку 16.
 2. Зарегистрируйтесь в приложении на ваш выбор Trello или Jira (выбирайте бесплатную версию).
 3. Добавьте список требований, составленный в виде пользовательских историй, в выбранное приложение для управления проектами (Trello или Jira).
 4. Разделите список на карточки или задачи и определите их статусы, например, «в работе», «готово», «на проверке». Это позволит вам отслеживать прогресс выполнения требований и управлять процессом разработки.
 5. Предоставьте доступ к проекту по ссылке или приложите скриншот проекта для проверки задания.

6.3. Насколько Agile влияет на удовольствие от работы

- Насколько Agile влияет на удовольствие от работы
 - Лучшая коммуникация
 - Увеличение автономии
 - Быстрая обратная связь
 - Более эффективная работа в команде
 - Улучшенный процесс разработки
- Практическое задание:
 1. Изучите основные принципы Agile-подхода и методы, которые используются для управления проектами.
 2. Исследуйте влияние Agile-подхода на удовлетворенность работой сотрудников, используя научные статьи, исследования, интервью с экспертами и т.д.
 3. Напишите эссе на тему «Насколько Agile влияет на удовольствие от работы», включив в него следующие аспекты:
 4. Описание основных принципов Agile-подхода и их отличие от традиционных методов управления проектами.
 5. Обсуждение основных факторов, которые влияют на удовлетворенность работой, включая связь между Agile-подходом и удовлетворенностью работой.
 6. Анализ исследований и практических примеров, которые демонстрируют, как Agile-подход может улучшить удовлетворенность работой сотрудников.
 7. Ваши собственные мысли и рекомендации, которые могут помочь лучше понять, как Agile-подход может влиять на удовлетворенность работой.

6.4. Анти-Паттерны Agile – команд

- Отчет вместо стендапа
- Антипаттерны Agile-команд
- Советы по организации и проведению ежедневных встреч в Scrum
 - Водопадные спринты
 - Водопадная итерация
 - Оценка задач
 - Как правильно читать Burndown
 - Работа с Backlog
 - Правильная оценка задач
- Практическое задание:
 1. Выберите из списка требований, составленных в практическом задании к уроку 16, требование, которое будет считаться базовым для определения Story Point.
 2. Оцените выбранное требование в 1 Story Point.
 3. Оцените остальные требования в Story Point, ориентируясь на базовое требование. При оценке можете использовать вопросы:

4. Насколько оцениваемое требование сложнее в реализации, чем базовое требование?
5. Насколько больше усилий потребуется для реализации оцениваемого требования в сравнении с базовым?

* Дополнительная общеразвивающая программа «Тестирование ПО. Продвинутый уровень» находится в разработке, в связи с этим возможны изменения в наименовании разделов, тем, в том числе в содержании уроков.

3.3 Календарный учебный график

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

Срок освоения программы — 17 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	16	17		
1.	Docker для тестировщика	РП	4	4																	
		СР	4	4																	
2.	Углубленный SQL	РП			6	6	6	4													
		СР			6	6	6	4													
3.	Продвинутый Python	РП							6	6	6	4									
		СР							6	6	6	4									
4.	Углубленное изучение: автоматизация тестирования	РП											8	8							
		СР											8	8							
5.	Управление командой	РП													5	5					
		СР													5	5					
6.	Agile в командах	РП																4	4		
		СР																4	4		
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. Суцевский 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;
- видеокамера (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 видеоинформации 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. Основы тестирования программного обеспечения. Учебное пособие / [Старолетов Сергей Михайлович]; Лань; 2022 г. - 382 с.
5. Основы тестирования и верификации программного обеспечения. Учебное пособие / [Старолетов Сергей Михайлович]; Лань; 2023 г. - 344 с.
6. Основы технологий баз данных. Учебное пособие / [Новиков Б. А., Горшкова Е. А.]; ДМК Пресс; 2020 г. – 582 с.
7. Проектирование и разработка web-приложений. Введение в frontend и backend разработку на JavaScript и node.js: учебное пособие для СПО [Заяц А., Васильев Н.]; Лань; 2023 г. – 120 с.
8. Разработка требований к программному обеспечению. 3-е изд., дополненное / [Вигерс К., Битти Дж.]; БХВ; 2023 г. - 736 с.

9. Тестирование программного обеспечения. Базовый курс/[Святослав Куликов]; ЕРАМ Systems, 2022 г.
10. Тест-дизайн. Практическое руководство для начинающих/ [Назина Ольга]; БХВ; 2023 г. – 512 с.
11. Искусство тестирования программ / [Сандлер, Майерс, Баджетт]; Диалектика; 2020 г. - 272 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

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. Продвинутый 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 (компетенции)	Объект оценивания (навыки)	Показатель оценки (знания, умения)
Работа с	Применение	Знания:

Docker	знаний по работе с Docker	<ul style="list-style-type: none"> • Основные команды Docker • Особенности разворачивания проекта Docker Умения: <ul style="list-style-type: none"> • Осуществить запуск сервисов с помощью Docker Compose • Развернуть свой проект в Docker
Работа с SQL (продвинутый уровень)	Применение знаний по основам работы с углублённым SQL	Знания: <ul style="list-style-type: none"> • Основы работы с базами данных • Выборка и агрегация данных • Сортировка и группировка данных • Фильтрация данных Docker и по тексту • Операторы вычисления • Объединение и создание таблиц • Использование подзапросов для фильтрации • Работа с комбинированными запросами Умения: <ul style="list-style-type: none"> • Составить базы данных • Создать запросы на отображение всех строк • Составить запросы на объединение таблиц • Составление запросов на работу с данными • Составить запросы с использованием подзапросов
Работа с Python (продвинутый уровень)		Знания: <ul style="list-style-type: none"> • Типы и структуры данных • Основные конструкции языка • Особенности работы со строками, списками, словарями и • коллекциями • Функции, классы и объекты • Объектно-ориентированное программирование (ООП) • Модификаторы доступа, • свойства и дескрипторы • Использование библиотек • Обработка исключений Умения: <ul style="list-style-type: none"> • Написать программы для запроса необходимой информации, вычислений и др. • Написать программы для работы с информацией о книгах в библиотеке • Создать классы, функции, виртуальные окружения • Написать selenium-тесты
Автоматизация тестирования на углубленном уровне	Применение знания и навыков по автоматизации тестирования	Знания: <ul style="list-style-type: none"> • Принципы автоматизации тестирования • Правила тестирования веб-приложений и API. • Тестовые фреймворки • Тестирование с помощью Selenium, PyTest • Автотесты на мобильные приложения на Python • Автоматизация тестирования API

		<ul style="list-style-type: none"> • Виды паттернов проектирования • Паттерн Page Object Model Умения: <ul style="list-style-type: none"> • Реализовывать различные тесты • Создать PUT-запрос в Postman и PATCH-запрос в Postman • Реализовать тест на Python • Реализовать паттерн Page Object на Python
Управление командой	<p>- Понимание основ управления командой</p> <p>- Использование полученных знаний в работе с командой</p>	Знания: <ul style="list-style-type: none"> • Основы лидерства • Правила приоритизации задач • Постановка целей команде и контроль за исполнением • Развитие и мотивация команды. • Особенности осуществления обратной связи • Решение конфликтов Умения: <ul style="list-style-type: none"> • Расставлять приоритеты в задачах • Ставить цели команде и контролировать их исполнение • Применять навыки развития и мотивации команды • Обеспечивать эффективную обратную связь • Разрешать конфликты
Работа с Agile	Применение знаний по работе с Agile	Знания: <ul style="list-style-type: none"> • Agile-подход и его назначение • Роль Scrum-мастера в команде • Основные принципы Agile-подхода и методы, которые используются для управления проектами • Анти-Паттерны Agile – команд Умения: <ul style="list-style-type: none"> • Навыки использования Agile-подхода

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 Docker»

1. Что такое Docker?

1. Виртуальная машина

- 2. Контейнерная платформа
- 3. Облачный репозиторий
- 4. Система управления базой данных
- 2. Что такое Docker контейнер?
 - 1. автономный исполняемый пакет, содержащий все необходимое для запуска приложения
 - 2. фоновый процесс, который работает на хост-машине
 - 3. Воспроизводимая среда
 - 4. Документация Docker
- 3. Как работает Docker?
 - 1. Он использует архитектуру клиент-сервер
 - 2. Он работает по локальной сети
 - 3. Он работает непосредственно на операционной системе хоста
 - 4. Он использует облачную модель развертывания
- 4. В чем польза от использования Docker?
 - 1. Он обеспечивает лучшую защиту данных, чем традиционные виртуальные машины
 - 2. Облегчает доступ к аппаратным ресурсам
 - 3. Облегчает развертывание приложений в различных средах
 - 4. Он требует меньше места для хранения, чем традиционные виртуальные машины
- 5. В чем польза использования docker для тестировщиков?
 - 1. Обеспечивает более быстрое и эффективное тестирование
 - 2. Контейнеры Docker позволяют тестировщикам создавать воспроизводимую среду
 - 3. Можно запускать тесты на различных операционных системах
 - 4. Все вышеперечисленное выше

Тест к уроку 1.2. «Основные команды Docker»

- 1. Что делает команда docker rm?
 - 1. Остановка запущенного контейнера
 - 2. Отправка образа в репозиторий
 - 3. Удаление контейнера
 - 4. Удаление образа
- 2. Что необходимо, чтобы переместить образы docker в хранилище?
 - 1. Использовать команду docker run
 - 2. Сжать их в размере меньше чем 2 mb
 - 3. Остановить все запущенные контейнеры
 - 4. Пометить их именем хранилища и версией образа
- 3. Контейнер создается?
 - 1. Из образа docker
 - 2. Из документации docker
 - 3. На основе данных от разработчиков docker
 - 4. Его можно создать в репозитории Docker Hub
- 4. К рекомендациям по управлению образами Docker не относятся:
 - 1. Уменьшение размера образа.
 - 2. Кэширование изображений
 - 3. Использовать Docker Hub
 - 4. Они все относятся к рекомендациям по управлению образами
- 5. Чтобы развернуть приложение требующего прямого доступа к физической сети, мы воспользуемся сетью?
 - 1. Оверлейные сети
 - 2. Сети Macvlan
 - 3. Хост-сети
 - 4. Мостовые сети

Тест к уроку 1.3. «Запуск сервисов с помощью Docker Compose»

1. Что такое Docker Compose?
 1. окружение
 2. локальная сеть
 3. инструмент
 4. контейнер
2. Что не входит в плюсы его использования?
 1. Управление несколькими контейнерами
 2. Упрощение разработки
 3. Масштабируемость
 4. Уменьшение размера контейнеров
3. Что не является сервисом в Docker Compose?
 1. Контейнер
 2. Порт
 3. Том
 4. Среда
4. Какая сеть используется в Docker Compose по умолчанию?
 1. Хост
 2. Оверлей
 3. Макvlan
 4. Мостовая
5. Какое расширение используется для Compose файлов?
 1. .docker
 2. .compose
 3. .yaml
 4. .comps

Тест к уроку 1.4. «Разворачивание проекта Docker»

1. Что не относится к преимуществам развертывания проекта в Docker?
 1. Переносимость
 2. Изоляция
 3. Масштабируемость
 4. Объективность
2. Что не входит в план развертывания проекта в Docker?
 1. Установка Docker
 2. Проверка Docker
 3. Обновление Docker
 4. Настройка Docker
3. Команда “docker push”?
 1. Переместит образ в реестр контейнеров
 2. Скачает образ из реестра контейнеров
 3. Создаст пустой dockerfile
 4. Покажет информацию о запущенных контейнерах
4. После перемещения образа в репозиторий, следует шаг?
 1. Это заключительный шаг
 2. Развертывание контейнера
 3. Настройка Docker
 4. Создание локальной сети
5. Docker Compose присутствует в ... ?
 1. Настройке Docker
 2. Развертывание контейнера
 3. Перемещении образа в репозиторий
 4. Он нигде не присутствует, это отдельный файл со своими настройками

Тест к теме 1. «Docker для тестировщика»

1. Что такое Docker?
 1. Виртуальная машина
 2. Контейнерная платформа
 3. Облачный репозиторий
 4. Система управления базой данных
2. В чем польза использования docker для тестировщиков?
 1. Обеспечивает более быстрое и эффективное тестирование
 2. Контейнеры Docker позволяют тестировщикам создавать воспроизводимую среду
 3. Можно запускать тесты на различных операционных системах
 4. Все вышеперечисленное выше
3. Что необходимо, чтобы переместить образы docker в хранилище?
 1. Использовать команду docker run
 2. Сжать их в размере меньше чем 2 mb
 3. Остановить все запущенные контейнеры
 4. Пометить их именем хранилища и версией образа
4. Что не является сервисом в Docker Compose?
 1. Контейнер
 2. Порт
 3. Том
 4. Среда
5. Какое расширение используется для Compose файлов?
 1. .docker
 2. .compose
 3. .yaml
 4. .cmpr
6. Какой сетью необходимо воспользоваться, чтобы развернуть приложение требующего прямого доступа к физической сети?
 1. Оверлейные сети
 2. Сети Macvlan
 3. Хост-сети
 4. Мостовые сети
7. Что не входит в план развертывания проекта в Docker?
 1. Установка Docker
 2. Проверка Docker
 3. Обновление Docker
 4. Настройка Docker
8. Что делает команда “docker push”?
 1. Переместит образ в реестр контейнеров
 2. Скачает образ из реестра контейнеров
 3. Создаст пустой dockerfile
 4. Покажет информацию о запущенных контейнерах
9. Какое будет следующее действие после перемещения образа в репозиторий?
 1. Заключительное
 2. Развертывание контейнера
 3. Настройка Docker
 4. Создание локальной сети
10. Где присутствует Docker Compose?
 1. В настройках Docker
 2. В развертывании контейнера
 3. В перемещении образа в репозиторий

4. Нигде, это отдельный файл со своими настройками

Тест к уроку 2.1. «Введение в базы данных»

1. Что такое базы данных?
 1. это прогрессивное веб-приложение
 2. это таблица со строками и столбцами
 3. это набор структурированной информации, который хранится в электронном виде
 4. это система для управления запросами к данным
2. Что из перечисленного является свойствами баз данных?
 1. стандартизация и нормализация
 2. быстродействие и релевантность
 3. независимость и безопасность данных
 4. все вышеперечисленное
3. В каких базах данные хранятся в форматах XML, JSON, BSON?
 1. Реляционные
 2. Документоориентированные
 3. Иерархические
 4. Сетевые
4. Что такое SQL?
 1. язык структурированных запросов
 2. язык гипертекстовой разметки
 3. это язык таблицы стилей
 4. язык веб-программирования
5. Какой из операторов используется для добавления новых данных в объект?
 1. CREATE
 2. SELECT
 3. INSERT
 4. UPDATE

Тест к уроку 2.2. «Выборка и агрегация данных»

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 оператор используется для сортировки результатов выборки?
 1. GROUP BY
 2. ORDER BY
 3. HAVING
 4. SELECT FROM
3. 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
4. Какой SQL запрос выведет список книг в жанре фикшн?
 1. SELECT * FROM BOOKS;
 2. SELECT * FROM BOOKS WHERE genre ='фэнтези';
 3. SELECT 'фикшн' FROM BOOKS WHERE genre ='фикшн';

4. `SELECT * FROM BOOKS WHERE genre ='фикшн';`
5. Какой оператор из перечисленных использует ASC и DESC для сортировки по возрастанию или убыванию?
1. UPDATE
 2. SELECT
 3. ORDER BY
 4. GROUP BY

Тест к уроку 2.3. «Сортировка и группировка»

1. 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
2. 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 записи, в которых продолжительность фильмов от полутора до двух часов

Тест к уроку 2.4. «Фильтрация данных»

1. Для чего используется оператор WHERE?
1. для группировки результатов выборки
 2. для определения количества строк.
 3. для указания условий выборки
 4. для объединения двух столбцов в один
2. Для чего используется функция BETWEEN?
1. для проверки на совпадение с одним из списка
 2. для указания любого количества символов
 3. для проверки на совпадение с заданным шаблоном
 4. для проверки на попадание в указанный интервал

Тест к уроку 2.5. «Фильтрация данных по тексту»

1. Для чего используется ключевое слово LIKE?
1. для группировки результатов выборки
 2. для проверки значения на соответствие шаблону
 3. для обозначения главной строки выборки
 4. для объединения двух столбцов в один
2. Какая часть шаблона отвечает за совпадение с любым из приведенных значений?
1. %
 2. _
 3. []

4. [^]

Тест к уроку 2.6. «Операторы вычисления»

1. Для чего используются операторы вычисления?
 1. для группировки результатов выборки
 2. для проверки значения на соответствие шаблону
 3. для проведения арифметических действий с числами
 4. для объединения двух столбцов в один
2. 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

Тест к уроку 2.7. «Объединение таблиц»

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 у результирующей таблицы, если в исходных таблицах было 7 и 3 значений соответственно?
 1. 7
 2. 3
 3. 10
 4. 21
5. Какое значение подставляется в пустые ячейки при использовании RIGHT и LEFT OUTER JOIN запросов?
 1. ноль
 2. 0
 3. NULL
 4. остается пустым

Тест к уроку 2.8. «Создание таблиц»

1. Какая команда создает таблицу в SQL-запросе?
 1. DROP TABLE
 2. CREATE TABLE
 3. ALTER TABLE
 4. ADD TABLE
2. Какие данные можно хранить в столбце с типом VARCHAR(256)?
 1. целочисленные значения от -2147483648 до 2147483647.

2. большой объем текста до 65 KB.
3. строка с максимально возможной длиной 256 символов
4. любые данные, весом 256 бит.

Тест к уроку 2.9. «Работа с данными»

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

Тест к уроку 2.10. «Использование подзапросов для фильтрации»

1. Что такое подзапрос в SQL?
 1. любой текст начинающийся с оператора
 2. запрос для создания новой строки
 3. запрос внутри другого запроса SQL
 4. запрос на изменение существующих данных
2. Чему эквивалентно такое условие $x \in \{5, 13\}$?
 1. $x > 13$
 2. $x < 5$
 3. $(x = 5) \text{ AND } (x = 13)$
 4. $x \text{ NOT IN } (5, 13)$

Тест к уроку 2.11. «Работа с комбинированными запросами»

1. Что такое комбинированный SQL-запрос?
 1. любой текст начинающийся с оператора
 2. объединение нескольких запросов в одном, с выводом результата в виде единой таблицы
 3. запрос внутри другого запроса SQL
 4. запрос на изменение сразу нескольких разных строк в нескольких таблицах
2. Какой оператор используется для работы с комбинированными запросами?
 1. UNIFY
 2. MERGE
 3. COMBINE
 4. UNION

Тест к теме 2 «Углубленный SQL»

1. Что из перечисленного является свойствами баз данных?

1. стандартизация и нормализация
2. быстродействие и релевантность
3. независимость и безопасность данных
4. все вышеперечисленное

2. Что такое SQL?

1. язык структурированных запросов
2. язык гипертекстовой разметки
3. это язык таблицы стилей
4. язык веб-программирования

3. Какой оператор используется для сортировки результатов выборки?

1. GROUP BY
2. ORDER BY
3. HAVING
4. SELECT FROM

4. Какой оператор из перечисленных использует ASC и DESC для сортировки по возрастанию или убыванию?

1. UPDATE
2. SELECT
3. ORDER BY
4. GROUP BY

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 записи, в которых продолжительность фильмов от полутора до двух часов

6. Для чего используется функция BETWEEN?

1. для проверки на совпадение с одним из списка
2. для указания любого количества символов
3. для проверки на совпадение с заданным шаблоном
4. для проверки на попадание в указанный интервал

7. Какая часть шаблона отвечает за совпадение с любым из приведенных значений?

1. %
2. _
3. []
4. [^]

8. 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

9. Какой оператор используется для выведения всех записей из двух таблиц?

1. FULL OUTER JOIN
 2. LEFT JOIN
 3. JOIN
 4. RIGHT JOIN
10. Сколько строк будет у результирующей таблицы, если в исходных таблицах было 7 и 3 значений соответственно?
1. 7
 2. 3
 3. 10
 4. 21
11. Какие данные можно хранить в столбце с типом VARCHAR(256)?
1. целочисленные значения от -2147483648 до 2147483647.
 2. большой объем текста до 65 KB.
 3. строка с максимально возможной длиной 256 символов
 4. любые данные, весом 256 бит.
12. Какой оператор используется для изменения существующих записей в таблице?
1. GROUP BY
 2. UPDATE
 3. CREATE
 4. SELECT
13. 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
14. Чему эквивалентно такое условие $x \diamond ALL (5, 13)$?
1. $x > 13$
 2. $x < 5$
 3. $(x = 5) AND (x = 13)$
 4. $x NOT IN (5, 13)$
15. Какой оператор используется для работы с комбинированными запросами?
1. UNIFY
 2. MERGE
 3. COMBINE
 4. UNION

Тест к уроку 3.1. «Типы и структуры данных»

1. Какой тип данных в Python используется для хранения целых чисел?
 1. float
 2. str
 3. int
 4. bool
2. Какой тип данных в Python используется для хранения текстовой информации?
 1. float
 2. str
 3. int
 4. bool
3. Какой тип данных будет, если сложить целое 5 и дробное 3.14? (22171)

1. float
 2. str
 3. int
 4. bool
4. Какая структура данных является неизменяемой?
1. Список (list)
 2. Словарь (dict)
 3. Кортеж (tuple)
 4. Множество (set)
5. Какую структуру данных можно использовать для хранения элементов в стеке (LIFO - последний вошел, первый вышел)?
1. Список (list)
 2. Кортеж (tuple)
 3. Словарь (dict)
 4. Множество (set)

Тест к уроку 3.2. «Основные конструкции языка»

2. Как объявить константу в Python?

1. var PI = 3.1415
2. const PI = 3.1415
3. final PI = 3.1415
4. PI = 3.1415

1) Что выведет следующий код?

```
x = 5
y = 10
if x > y:
    print("x больше чем y")
else:
    print("x меньше или равно y")
```

1. x больше чем y
2. x меньше чем y
3. x меньше или равно y
4. Ничего из вышеперечисленного

2) Какой будет равно z в результате следующего кода? (22176)

```
x = 10
y = 10
z = 0
if x > y:
    z = 5
elif x == y:
    z = 55
else:
    z = 555

print(z)
```

1. 0
2. 5
3. 55

4. 555

1. Какие типы циклов поддерживаются в Python?
 1. for, while, do-while, repeat-until
 2. for, while, until, foreach
 3. for, while, do-while, continue
 4. for, while
5. Каким оператором можно пропустить текущую итерацию в цикле в Python?
 1. continue
 2. next
 3. skip
 4. ignore

Тест к уроку 3.3. «Работа со строками»

1. Как создать строку в Python?
 1. string = "hello world"
 2. string = 'hello world'
 3. string = """"hello world""""
 4. все вышеперечисленные варианты верны
2. Как добавить строку к другой строке в Python?
 1. str1 = "hello"
str2 = "world"
string = str1 + str2
 2. str1 = "hello"
str2 = "world"
string = str1.append(str2)
 3. str1 = "hello"
str2 = "world"
string = str1.expand(str2)
 4. невозможно добавить строку к другой строке в Python
3. Как получить длину строки в Python?
 1. string = "hello world"
length = string.length()
 2. string = "hello world"
length = len(string)
 3. string = "hello world"
length = string.count()
 4. string = "hello world"
length = string.size()
4. Как проверить, содержится ли подстрока в строке?
 1. string = "hello world"
substring = "world"
if substring in string:
print("Yes")
else:
print("No")
 2. string = "hello world"
substring = "planet"
if substring not in string:
print("No")
else:
print("Yes")
 3. string = "hello world"
substring = "world"

```

if string.has(substring):
    print("Yes")
else:
    print("No")

```

4. только а и b верны

5. Как заменить часть строки на другую строку в Python?

1. `string = "hello world"`
`new_string = string.replace("world", "planet")`
2. `string = "hello world"`
`new_string = string.swap("world", "planet")`
3. `string = "hello world"`
`new_string = string.switch("world", "planet")`
4. НЕВОЗМОЖНО ИЗМЕНИТЬ СТРОКУ в Python

<pre> str1 = "hello" str2 = "world" string = str1 + str2 </pre>	1	<pre> str1 = "hello" str2 = "world" string = str1.append(str2) </pre>	2
<pre> str1 = "hello" str2 = "world" string = str1.expand(str2) </pre>	3	<pre> str1 = "hello" str2 = "world" string = str1.expand(str2) </pre>	4
невозможно добавить строку к другой строке в Python			5

Тест к уроку 3.4. «Работа со списками»

1. Какой метод в Python используется для добавления элемента в конец списка?

1. `add()`
2. `push()`
3. `append()`
4. `insert()`

2. Какой метод в Python используется для удаления последнего элемента из списка?

1. `remove()`
2. `delete()`
3. `pop()`
4. `discard()`

3. Что из следующего верно для списков Python?

1. Список может содержать объекты любого типа
2. Все элементы в списке должны быть одного типа
3. Эти списки одинаковые:

```

4. ['a', 'b', 'c']
   ['c', 'a', 'b']

```

5. Список может содержать одинаковые значения

4. Какой метод в Python используется для сортировки списка в порядке возрастания?

1. `sort()`
2. `arrange()`
3. `order()`
4. `organize()`

5. Какой метод в Python используется для создания нового списка на основе существующего?

1. copy()
2. clone()
3. update()
4. duplicate()

Тест к уроку 3.5. «Работа со словарями и коллекциями»

1. Как получить список всех ключей в словаре?
 1. dict.get_keys()
 2. dict.keys()
 3. dict.all_keys()
 4. dict.key_list()
2. Что из следующего верно для словарей Python?
 1. Доступ к элементам осуществляется по их положению в словаре.
 2. Доступ к словарям осуществляется по ключу.
 3. Словарь может содержать любой тип объекта, кроме другого словаря.
 4. Словари изменяемы.
2. Какая структура данных Python является неизменяемой?
 1. tuple
 2. list
 3. set
 4. dict
3. Какой метод используется для объединения двух списков в Python?
 1. merge()
 2. join()
 3. append()
 4. extend()
4. Какая структура данных Python использует для хранения пары ключ-значение?
 1. tuple
 2. list
 3. set
 4. dict

Тест к уроку 3.6. «Функции»

1. Что такое функция в Python?
 1. Переменная со значением
 2. Блок кода, который можно вызвать из другого кода
 3. Ошибки в коде
 4. Программный модуль
2. Как объявить функцию в Python?
 1. function myfunction():
 2. myfunction = function():
 3. def myfunction():
 4. declare myfunction as function():
3. Что такое аргументы функции?
 1. Переменные, которые передаются в функцию при ее вызове
 2. Значения переменных, объявленных внутри функции
 3. Алгоритм работы функции
 4. Используемые модули
4. Как в Python можно вернуть значение из функции?
 1. функция возвращает значение неявно

2. с помощью оператора return
 3. при помощи оператора print
 4. значение нельзя вернуть из функции
5. Как можно вызвать функцию в Python?
1. Вызов функции осуществляется при помощи функции myfunction()
 2. Через ключевое слово import
 3. при помощи оператора print
 4. функция вызывается автоматически, как только программа запускается.

Тест к уроку 3.7. «Классы и объекты»

1. Что НЕ является объектом в Python?
 1. Целое число
 2. Функция
 3. Класс
 4. Все сущности в Python являются объектами
2. Что делает оператор pass?
 1. Ничего
 2. Кидает исключение
 3. Пропускает итерацию в цикле
 4. Запускает отладчик
3. С помощью какого ключевого слова создаются классы в Python?
 1. cls
 2. class
 3. klass
 4. type
6. Что означает одиночное нижнее подчеркивание в начале имени атрибута?
 1. Имя атрибута зарезервировано внутренними механизмами языка
 2. Ничего не означает
 3. Атрибут следует считать приватным
 4. Имена любых атрибутов нужно начинать с нижнего подчеркивания
8. Как создать атрибут экземпляра класса в Python?
 1. Объявить в виде поля класса
 2. Объявить переменную внутри конструктора
 3. Создать в конструкторе новое поле для self
 4. В Python есть только атрибуты самого класса

Тест к уроку 3.8. «Объектно-ориентированное программирование (ООП)»

1. Что такое ООП?
 1. Подход к программированию, основанный на объектах и их взаимодействии
 2. Способ организации рабочих процессов
 3. Подход к программированию, при котором описывается ожидаемый результат, а не способ его достижения
 4. Язык программирования
2. Что из перечисленного НЕ является принципом ООП?
 1. Инкапсуляция
 2. Наследование
 3. Полиморфизм
 4. Все является
3. Что такое абстракция В ООП?
 1. Процесс разделения сложной задачи на простые

2. Способ представления упрощенной модели объекта
3. Детально проработанная точная модель объекта и всех его свойств
4. Что такое наследование в ООП?
 1. Процесс адаптации старого кода к новым условиям
 2. Способность абстракции перенимать свойства другой абстракции
 3. Процесс отмирания старого кода
5. Что такое инкапсуляция в ООП?
 - a. Способ объединения множества сущностей в одну
 - b. Изоляция внешнего представления от внутренней логики
 - c. Внедрение части кода на другом языке

Тест к уроку 3.9. «Модификаторы доступа, свойства и дескрипторы»

1. Какие модификаторы доступа есть в Python:
 1. private
 2. protected
 3. private, protected и public
 4. В языке Python нет модификаторов доступа в их привычном понимании
2. С помощью какой переменной можно определить публичный интерфейс модуля?
 1. `__interface__`
 2. `__public__`
 3. `__all__`
 4. `__module__`
3. Для чего используется одиночное нижнее подчёркивание на конце имени:
 1. Не используется в Python
 2. Для обозначения внутреннего интерфейса
 3. Чтобы избежать конфликтов имён
 4. Для переменных внутри функций и методов
4. Какие бывают дескрипторы?
 1. Только данных
 2. Данных и не данных
 3. Только не данных
 4. Ничего из перечисленного
7. Какие процессы доступа к атрибутам позволяет определить property?
 1. Чтение
 2. Запись
 3. Удаление
 4. Все перечисленные

Тест к уроку 3.10. «Использование библиотек»

1. Какой файл используется при импорте пакета?
 1. `__init__.py`
 2. `__module__.py`
 3. `__package__.py`
 4. `__import__.py`
2. Какой из этих вариантов синтаксически правильный?
 1. `from hashlib import sha512`
 2. `from hashlib import sha512 as hash_func`
 3. `from hashlib import *`
 4. Все перечисленные
3. Относительно чего строится путь при относительном импорте?
 1. Корня проекта

2. Рабочего каталога
3. Текущего модуля
4. Исполняемого файла
4. Разные виртуальные окружения позволяют использовать:
 1. Разные версии интерпретатора
 2. Разные версии библиотек
 3. Разные версии интерпретатора и библиотек
 4. Ничего из перечисленного
5. Какой из этих способов импорта не рекомендуется использовать?
 1. `import hashlib`
 2. `from hashlib import sha512`
 3. `from hashlib import sha512 as hash_func`
 4. `from hashlib import *`

Тест к уроку 3.11. «Обработка исключений»

1. От какого класса наследуются абсолютно все исключения в Python?
 1. `Error`
 2. `BaseError`
 3. `Exception`
 4. `BaseException`
2. Какой оператор /нужно использовать чтобы бросить исключение в Python?
 1. `throw`
 2. `raise`
 3. `fire`
 4. `try`
3. Как следует перехватывать исключения?
 1. Указывая конкретные классы исключений
 2. Указывая класс `Exception`
 3. Не указывая никакие классы
 4. Любым из перечисленных способов
4. Как указать несколько классов в одном блоке `except`?
 1. С помощью вертикальной черты `|`
 2. С помощью оператора сложения `+`
 3. Поместить их в кортеж
 4. Нет такой возможности
5. Когда вызывается блок `else`?
 1. Когда в блоке `try` было брошено исключение
 2. Когда исключение обработано в блоке `except`
 3. Когда в блоке `try` не было исключений
 4. Всегда

Тест к теме 3 «Продвинутый Python»

1. Какой тип данных в Python используется для хранения целых чисел?
 1. `float`
 2. `str`
 3. `int`
 4. `bool`
2. Какая структура данных является неизменяемой?
 1. Список (`list`)
 2. Словарь (`dict`)
 3. Кортеж (`tuple`)

4. Множество (set)

3. Что выведет следующий код?

```
x = 5
y = 10
if x > y:
    print("x больше чем y")
else:
    print("x меньше или равно y")
```

1. x больше чем y
 2. x меньше чем y
 3. x меньше или равно y
 4. Ничего из вышеперечисленного
4. Каким оператором можно пропустить текущую итерацию в цикле в Python?
1. continue
 2. next
 3. skip
 4. ignore
5. Как добавить строку к другой строке в Python?
1. str1 = "hello"
str2 = "world"
string = str1 + str2
 2. str1 = "hello"
str2 = "world"
string = str1.append(str2)
 3. str1 = "hello"
str2 = "world"
string = str1.expand(str2)
 4. невозможно добавить строку к другой строке в Python
6. Как заменить часть строки на другую строку в Python?
1. string = "hello world"
new_string = string.replace("world", "planet")
 2. string = "hello world"
new_string = string.swap("world", "planet")
 3. string = "hello world"
new_string = string.switch("world", "planet")
 4. невозможно изменить строку в Python
7. Какой метод в Python используется для удаления последнего элемента из списка?
1. remove()
 2. delete()
 3. pop()
 4. discard()
8. Какой метод в Python используется для сортировки списка в порядке возрастания?
1. sort()
 2. arrange()
 3. order()
 4. organize()
9. Что из следующего верно для словарей Python?
1. Доступ к элементам осуществляется по их положению в словаре.
 2. Доступ к словарям осуществляется по ключу.
 3. Словарь может содержать любой тип объекта, кроме другого словаря.
 4. Словари изменяемы.

10. Какая структура данных Python использует для хранения пары ключ-значение?
 1. tuple
 2. list
 3. set
 4. dict
11. Как в Python можно вернуть значение из функции?
 1. функция возвращает значение неявно
 2. с помощью оператора return
 3. при помощи оператора print
 4. значение нельзя вернуть из функции
12. Как в Python можно вернуть значение из функции?
 1. функция возвращает значение неявно
 2. с помощью оператора return
 3. при помощи оператора print
 4. значение нельзя вернуть из функции
13. Что НЕ является объектом в Python?
 1. Целое число
 2. Функция
 3. Класс
 4. Все сущности в Python являются объектами
14. Что из перечисленного НЕ является принципом ООП?
 1. Инкапсуляция
 2. Наследование
 3. Полиморфизм
 4. Все является
15. Что такое наследование в ООП?
 1. Процесс адаптации старого кода к новым условиям
 2. Способность абстракции перенимать свойства другой абстракции
 3. Процесс отмирания старого кода
16. С помощью какой переменной можно определить публичный интерфейс модуля?
 1. `__interface__`
 2. `__public__`
 3. `__all__`
 4. `__module__`
17. Какие процессы доступа к атрибутам позволяет определить property?
 1. Чтение
 2. Запись
 3. Удаление
 4. Все перечисленные
18. Разные виртуальные окружения позволяют использовать:
 1. Разные версии интерпретатора
 2. Разные версии библиотек
 3. Разные версии интерпретатора и библиотек
 4. Ничего из перечисленного
19. Как указать несколько классов в одном блоке excerpt?
 1. С помощью вертикальной черты “|”
 2. С помощью оператора сложения “+”
 3. Поместить их в кортеж
 4. Нет такой возможности

Тест к уроку 4.1. «Как тестировать веб- приложения и API. Тестовые фреймворки»

1. Что такое автоматизация тестирования?
 1. Методика тестирования, при которой тестирование проводится специальными инструментами автоматизации
 2. Методика тестирования, призванная уменьшать рутину
 3. Вид тестирования, который нигде не используется
 4. Вид тестирования, который занимает большую часть в тестировании
2. Какого этапа нет в автоматизации тестирования?
 1. Составление сценариев
 2. Поддержка
 3. Маркетинг
 4. Планирование
3. Какого тестового фреймворка не существует?
 1. Behave
 2. Robot Framework
 3. Lettuce
 4. Putest

Тест к уроку 4.2. «Тестирование с помощью Selenium»

1. Что такое Selenium?
 1. Технология, которая экономит ресурсы и время
 2. Позволяет автоматизировать всё, что угодно
 3. Набор инструментов для анализа покрытия кода тестами
 4. Инструменты для автоматизации тестирования
2. Что такое Selenium WebDriver?
 1. Утилита для дебага кода тестов
 2. Утилита для захвата элементов страницы
 3. Инструмент для автоматизации действий браузера
 4. Инструменты для автоматизации тестирования
3. Что произойдет, если запустить этот тест?

```
from selenium import webdriver
driver = webdriver.Chrome()
driver.get("https://google.ru")
driver.close()
```

1. Откроется браузер и произойдет переход по урлу google.ru
 2. Откроется браузер
 3. Откроется браузер и произойдет переход по урлу google.ru, окно браузера не будет закрыто
 4. Ничего, есть ошибка
4. Как можно найти элемент по тегу h2?
 1. driver.find_element_by_tag("h2")
 2. myValue = driver.find_element(By.TAG, "h2")
 3. driver.find_element(By.TAG, "h2")
 4. myValue = driver.find_element(By.TAG_NAME, "h2")

Тест к уроку 4.3. «Тестирование с помощью PyTest»

1. Что такое Assert?
 1. Проверка в тесте
 2. Проверяющая система
 3. Вид тестирования на Python

4. Специальный атрибут в Pytest
2. Для чего нужны ассерты?
 1. Для проверки двух массивов
 2. Для сравнения let, var, const
 3. Для сравнения двух массивов
 4. Для сравнения двух значений
3. Какая функция предназначена для вызова исключения?
 1. Exception
 2. RaiseException
 3. Raises
 4. Throws
4. Что означает ZeroDivisionError?
 1. Исключение общего характера
 2. Ошибка деления на ноль
 3. Множественное исключение
 4. Тип исключения
5. Что означает эта строчка кода?


```
assert func() == 2
```

 1. Проверка функции на null
 2. Проверяется то, что не выбрасывается исключение
 3. Ассертится функция с функцией
 4. Проверяется, что возвращаемое значение функции равно 2

Тест к уроку 4.4. «Использование фикстур в PyTest»

1. Что такое фикстура?
 1. Функция, которая запускается до или после тестовых функций
 2. Это то, что пишут не в файле conftest.py
 3. Это вид автоматизированного тестирования
 4. Это параметр функции pytest
2. Что делает эта конструкция?


```
@pytest.fixture
```

 - a. Скипает тесты
 - b. Помечает тесты красными
 - c. Создает фикстуру
 - d. Помечает тестовую функцию фиксурой
3. Какой параметр позволяет запустить тесты с выводом в консоль?
 1. Console
 2. C
 3. S
 4. K
4. Что нужно сделать, чтобы глобально включить фикстуру
 1. Это невозможно
 2. Вместо `@pytest.fixture()` написать `@pytest.mark.globalFixture()`
 3. В `@pytest.fixture()` внутри скобок добавить `Global`
 4. В `@pytest.fixture()` внутри скобок добавить `autouse=True`
5. Какое ключевое слово в фикстуре позволяет разграничить то, что выполнится до и после тестовых функций?
 1. After
 2. Before
 3. Teardown

4. Yield

Тест к уроку 4.5. «PyTest»

1. Что такое параметризация?
 1. Запуск теста с одним или несколькими параметрами
 2. Сущность, которая передается тесту
 3. Вариант входных данных
 4. Тест, которому передали несколько параметров
2. Какой маркер нужно использовать для параметризации тестов?
 1. Params
 2. P
 3. Parameter
 4. Parametrize
3. Что такое маркировка?
 1. Нанесение кода на товар
 2. Навешивание уникального кода на функцию
 3. Процесс навешивания маркеров на тесты
 4. Процесс написания теста
4. Для чего нужен маркер skip?
 1. Чтобы разрешить запуск теста
 2. Беспольный маркер
 3. Для того, чтобы украсить тесты
 4. Чтобы не запускать тест
5. Какой файл нужно создать для хранения собственных маркеров?
 1. Pytest.py
 2. Python.py
 3. Python.ini
 4. Pytest.ini

Тест к уроку 4.6. «Автотесты на мобильные приложения на Python»

1. При помощи какого инструмента искались элементы в уроке?
 1. Appium Inspector
 2. Appium Server GUI
 3. Appium
 4. Inspector
2. Для чего нужен эмулятор?
 1. Для эмуляции действий Desktop-приложений
 2. Чтобы эмулировать поведение в браузере
 3. Для запуска тестов
 4. Чтобы запускать мобильные приложения
3. Что делает этот код?

```
driver.find_element(AppiumBy.ID, 'com.google.android.calculator:id/op_add').click()
```

 1. Кликает по элементу
 2. Находит элемент по классу
 3. Находит элемент по ID и кликает по нему
 4. Код не работает
4. Где создавался Android-эмулятор в уроке?
 1. В Pycharm
 2. В Appium
 3. В Android Studio
 4. В Device Manager

5. Какие параметры нужно указать Appium Server'у?

1. Url сервера
2. Имя устройства
3. Версию Android
4. Url сервера и настройки приложения

Тест к уроку 4.7. «Автоматизация тестирования API»

1. Какая цель у API?

1. Предоставление функциональности пользователям, либо программам
2. Описание интерфейса приложения
3. Реализация функциональности
4. Тестирование программного интерфейса

2. Какого вида запросов не существует?

1. POST
2. GET
3. PATCH
4. POSTING

3. Какие сущности в Postman используются для обращения к API?

1. Элементы
2. Классы
3. Запросы
4. Тесты

4. Для чего нужен токен?

1. Токен не требуется
2. Для проверки на уязвимости API
3. Для обеспечения безопасности тестов на API
4. Для разграничения доступа пользователей к API

5. Какой маркер используется для параметризации тестов?

1. Mark
2. Parameter
3. Params
4. Parametrize

Тест к уроку 4.8. «Паттерн Page Object Model»

1. Что такое паттерн проектирования?

1. Способ организации кода тестов
2. Построение тестовой инфраструктуры
3. Способ написания теста
4. Правило наименования теста

2. Какой паттерн позволяет задать несколько входных данных для одного теста?

1. Сборка окружения
2. ObjectMother
3. Page Object
4. Параметризация

3. Согласно какому паттерну тест должен быть из трех блоков?

1. Page Object
2. Параметризация
3. AAA
4. AAB

4. Какой паттерн позволяет отделить одни тесты от других?

1. Разборка окружения

2. Параметризация
3. Логирование
4. Маркировка
5. Какого паттерна не существует?
 1. Параметризация
 2. Наименование тестов
 3. Маркировка
 4. Виртуализация

Тест к теме 4. «Углубленное изучение: автоматизация тестирования»

1. Что такое автоматизация тестирования?
 1. Методика тестирования, при которой тестирование проводится специальными инструментами автоматизации
 2. Методика тестирования, призванная уменьшать рутину
 3. Вид тестирования, который нигде не используется
 4. Вид тестирования, который занимает большую часть в тестировании
2. Что такое Selenium?
 1. Технология, которая экономит ресурсы и время
 2. Позволяет автоматизировать всё, что угодно
 3. Набор инструментов для анализа покрытия кода тестами
 4. Инструменты для автоматизации тестирования
3. Что произойдет, если запустить этот тест?


```
from selenium import webdriver
driver = webdriver.Chrome()
driver.get("https://google.ru")
driver.close()
```

 1. Откроется браузер и произойдет переход по urlу google.ru
 2. Откроется браузер
 3. Откроется браузер и произойдет переход по urlу google.ru, окно браузера не будет закрыто
 4. Ничего, есть ошибка
4. Что такое Assert?
 1. Проверка в тесте
 2. Проверяющая система
 3. Вид тестирования на Python
 4. Специальный атрибут в Pytest
5. Какая функция предназначена для вызова исключения?
 1. Exception
 2. RaiseException
 3. Raises
 4. Throws
6. Что такое фикстура?
 1. Функция, которая запускается до или после тестовых функций
 2. Это то, что пишут не в файле conftest.py
 3. Это вид автоматизированного тестирования
 4. Это параметр функции pytest
7. Что нужно сделать, чтобы глобально включить фикстуру
 1. Это невозможно
 2. Вместо @pytest.fixture() написать @pytest.mark.globalFixture()
 3. В @pytest.fixture() внутри скобок добавить Global

4. В `@pytest.fixture()` внутри скобок добавить `autouse=True`
8. Что такое параметризация?
 1. Запуск теста с одним или несколькими параметрами
 2. Сущность, которая передается тесту
 3. Вариант входных данных
 4. Тест, которому передали несколько параметров
9. Что такое маркировка?
 1. Нанесение кода на товар
 2. Навешивание уникального кода на функцию
 3. Процесс навешивания маркеров на тесты
 4. Процесс написания теста
10. При помощи какого инструмента искались элементы в уроке?
 1. Appium Inspector
 2. Appium Server GUI
 3. Appium
 4. Inspector
11. Что делает этот код?
`driver.find_element(AppiumBy.ID, 'com.google.android.calculator:id/op_add').click()`
 1. Кликает по элементу
 2. Находит элемент по классу
 3. Находит элемент по ID и кликает по нему
 4. Код не работает
12. Какая цель у API?
 1. Предоставление функциональности пользователям, либо программам
 2. Описание интерфейса приложения
 3. Реализация функциональности
 4. Тестирование программного интерфейса
13. Какие сущности в Postman используются для обращения к API?
 1. Элементы
 2. Классы
 3. Запросы
 4. Тесты
14. Что такое паттерн проектирования?
 1. Способ организации кода тестов
 2. Построение тестовой инфраструктуры
 3. Способ написания теста
 4. Правило наименования теста
15. Согласно какому паттерну тест должен быть из трех блоков?
 1. Page Object
 2. Параметризация
 3. AAA
 4. AAB

Тема «Управление командой»

Тест к уроку 5.1. «Кто такой тимлид команды и как им стать»

1. Кто такой тимлид?
 1. специалист, который переводит бизнес требования пользователей в технические требования для разработчиков
 2. специалист, который составляет тестовую стратегию и проводит тестирование продукта

3. специалист, отвечающий за стратегию разработки и подбор тех команды
 4. специалист, отвечающий за процесс сборки и доставки изменений кода от локальных файлов в различные окружения
2. Что входит в обязанности тимлида?
 1. создание архитектуры приложения
 2. проведение ревью кода и менторинга
 3. разбираться в технологии скрам
 4. планировать бюджет проекта
 3. Какие навыки необходимы тимлиду?
 1. Планирование разработки
 2. Навыки подбора и управления командой
 3. Экспертиза в определенном стеке технологий
 4. Все вышеперечисленное
 4. Как тестировщик может стать тимлидом?
 1. Репортить как можно больше багов в день
 2. Самостоятельно изучать стек разработки, развиваться из ручного тестирования в написание автотестов
 3. Развиваться только в лидерских навыках, проходить мастер классы и тренинги на лидерство
 4. Все вышеперечисленное

Тест к уроку 5.2. «Приоритизация задач и умение сказать – Нет»

1. Зачем нужно расставлять приоритеты на разработку функциональности?
 1. чтобы ваша любимая задача была выполнена самой первой
 2. чтобы задача от руководства была выполнена самой первой
 3. чтобы выявить самые важные задачи для успеха продукта и взять их в работу в первую очередь, учитывая при этом сложность выполнения
 4. чтобы выявить самые простые задачи и взять их в работу в первую очередь, отложив самые сложные на потом
2. Какой из кейсов нужно взять в работу в первую очередь?
 1. срочный запрос от представителя отдела продаж
 2. запрос от генерального директора
 3. функциональность, которая принесет 10% прибыли, но в дальнейшем потерю 60% клиентов
 4. запрос о критической ошибке с продакшена
3. Как рассчитать оценку приоритета по методу RICE?
 1. $R \times I \times C \times E$
 2. $(R \times I) / (C \times E)$
 3. $(R \times I \times C) / E$
 4. $(I \times C \times E) / R$

Тест к уроку 5.3. «Постановка целей команде и контроль за исполнением»

1. Выберите вариант, где указаны все критерии хорошо поставленной цели?
 1. конкретная, измеримая, достижимая
 2. соответствие стратегии компании и ограниченность по времени
 3. конкретная, измеримая, достижимая, в соответствии с глобальной стратегией компании и ограничена во времени
 4. конкретная, актуальная, простая, в соответствии с глобальной стратегией компании и ограничена во времени
2. Поможет ли прогрессу проекта, вопрос от менеджера об успехах каждые полчаса?

1. Да
 2. Нет
3. Как можно объективно оценить прогресс достижения цели?
1. поговорить со всеми сотрудниками и спросить их личные ощущения
 2. наблюдать за работой сотрудников и оценить прогресс
 3. использовать системы управления проектами с указаниями там задач и их сроков
 4. подсчитать количество выполненных задач каждым из сотрудников

Тест к уроку 5.6. «Развитие и мотивация команды»

1. Какой вопрос стоит задать сотруднику на встрече по постановке личной цели?
 1. Когда вы собираетесь рожать?
 2. Что вы делали вчера вечером?
 3. Какие навыки вы используете в работе больше остальных?
 4. Куда бы вы хотели уйти работать?
2. Будет ли мотивация сотрудников результативной, если собирать обратную связь раз в семь лет?
 1. Да
 2. Нет
3. Что из перечисленного является подходом к развитию команды?
 1. Признание достижений
 2. Создание положительной рабочей обстановки
 3. Развитие лидерских качеств
 4. Все вышеперечисленное

Тест к уроку 5.7. «Решение конфликтов»

1. Какие из причин возникновения конфликтов относятся к профессиональной деятельности?
 1. Ссора между коллегами из-за футбольного матча
 2. Недобросовестное выполнение обязательств
 3. Конфликт между более взрослыми и молодыми представителями команды
 4. Негативное отношение к коллеге из-за личной неприязни
2. Что из перечисленного поможет избежать развития конфликта?
 1. Не скрывать наличие проблемы
 2. Регулярно общаться со своими сотрудниками, получать обратную связь
 3. Заботиться о том, чтобы все члены команды могли комфортно работать вместе
 4. Все вышеперечисленное
3. Что из перечисленного не является хорошей стратегией решения конфликта?
 1. инициация дискуссии между конфликтующими сторонами
 2. защита своего любимого сотрудника, даже если он неправ
 3. сохранение нейтралитета, фокусирование на фактах
 4. поиск решения проблемы безотносительно того, кто виноват

Тест к теме 5. «Управление командой»

1. Кто такой тимлид?
 1. специалист, который переводит бизнес требования пользователей в технические требования для разработчиков
 2. специалист, который составляет тестовую стратегию и проводит тестирование продукта
 3. специалист, отвечающий за стратегию разработки и подбор тех команды

4. специалист, отвечающий за процесс сборки и доставки изменений кода от локальных файлов в различные окружения
2. Какие навыки необходимы тимлиду?
 1. Планирование разработки
 2. Навыки подбора и управления командой
 3. Экспертиза в определенном стеке технологий
 4. Все вышеперечисленное
3. Зачем нужно расставлять приоритеты на разработку функциональности?
 1. чтобы ваша любимая задача была выполнена самой первой
 2. чтобы задача от руководства была выполнена самой первой
 3. чтобы выявить самые важные задачи для успеха продукта и взять их в работу в первую очередь, учитывая при этом сложность выполнения
 4. чтобы выявить самые простые задачи и взять их в работу в первую очередь, отложив самые сложные на потом
4. Как рассчитать оценку приоритета по методу RICE?
 1. $R \times I \times C \times E$
 2. $(R \times I) / (C \times E)$
 3. $(R \times I \times C) / E$
 4. $(I \times C \times E) / R$
5. Выберите вариант, где указаны все критерии хорошо поставленной цели?
 1. конкретная, измеримая, достижимая
 2. соответствие стратегии компании и ограниченность по времени
 3. конкретная, измеримая, достижимая, в соответствии с глобальной стратегией компании и ограничена во времени
 4. конкретная, актуальная, простая, в соответствии с глобальной стратегией компании и ограничена во времени
6. Как можно объективно оценить прогресс достижения цели?
 1. поговорить со всеми сотрудниками и спросить их личные ощущения
 2. наблюдать за работой сотрудников и оценить прогресс
 3. использовать системы управления проектами с указаниями там задач и их сроков
 4. подсчитать количество выполненных задач каждым из сотрудников
7. Какой вопрос стоит задать сотруднику на встрече по постановке личной цели?
 1. Когда вы собираетесь рожать?
 2. Что вы делали вчера вечером?
 3. Какие навыки вы используете в работе больше остальных?
 4. Куда бы вы хотели уйти работать?
8. Что из перечисленного является подходом к развитию команды?
 1. Признание достижений
 2. Создание положительной рабочей обстановки
 3. Развитие лидерских качеств
 4. Все вышеперечисленное
9. Какие из причин возникновения конфликтов относятся к профессиональной деятельности?
 1. Ссора между коллегами из-за футбольного матча
 2. Недобросовестное выполнение обязательств
 3. Конфликт между более взрослыми и молодыми представителями команды
 4. Негативное отношение к коллеге из-за личной неприязни
10. Что из перечисленного не является хорошей стратегией решения конфликта?
 1. инициация дискуссии между конфликтующими сторонами

2. защита своего любимого сотрудника, даже если он неправ
3. сохранение нейтралитета, фокусирование на фактах
4. поиск решения проблемы безотносительно того, кто виноват

Тема 6. Agile в командах

Тест к уроку 6.1. «Что такое Agile - подход и зачем он нужен бизнесу»

1. В чем состоит главный недостаток водопадной модели разработки ПО?
 1. Она не предоставляет возможности для внесения изменений в процессе разработки
 2. Она не является эффективной при работе в команде
 3. Она не подходит для больших проектов
 4. Она не позволяет быстро реагировать на изменения требований
 5. Она не гарантирует качество конечного продукта
2. Какие из нижеперечисленных видов проектов наиболее подходят для использования водопадной модели разработки ПО?
 1. Маленькие и простые проекты
 2. Сложные проекты, требующие многих итераций
 3. Проекты, где необходимо быстро реагировать на изменения требований
 4. Проекты, где разработка происходит в условиях высокой неопределенности
 5. Большие проекты, где участвует много команд
3. Что представляет собой V-образная модель разработки ПО?
 1. Модель, в которой все этапы процесса разработки выполняются последовательно
 2. Модель, в которой весь процесс разработки разбит на отдельные этапы, которые выполняются параллельно
 3. Модель, в которой каждый этап процесса разработки тесно связан с этапом тестирования
 4. Модель, в которой процесс разработки выполняется в циклическом порядке
 5. Модель, в которой разработка ПО осуществляется с помощью специальных инструментов.
4. В чем состоит основное отличие инкрементно-итеративной модели разработки ПО от водопадной модели?
 1. Инкрементно-итеративная модель - разработка постепенно с улучшениями, водопадная - линейный процесс разработки.
 2. Инкрементно-итеративная модель не предусматривает спецификации требований, в отличие от водопадной модели.
 3. Водопадная модель - параллельная разработка на нескольких этапах, инкрементно-итеративная - последовательное выполнение этапов.
 4. Инкрементно-итеративная модель - работа с заказчиком и обратная связь, водопадная - не предусматривает такой связи.
5. Что из перечисленного ниже не относится к ценностям Agile-манифеста?
 1. Работающий продукт важнее исчерпывающей документации.
 2. Сотрудничество с заказчиком важнее согласования условий контракта.
 3. Следование плану важнее готовности к изменениям.
 4. Люди и взаимодействие важнее процессов и инструментов.

Тест к уроку 6.2. «Роль Scrum - мастера в команде»

1. Какие роли определены в команде Scrum?
 1. Владелец продукта, разработчик, тестировщик
 2. Владелец продукта, scrum-мастер, команда разработки
 3. Руководитель, разработчик, аналитик

2. Какое совещание проводится ежедневно в Scrum?
 1. Демонстрация
 2. Планирование спринта
 3. Daily Standup
3. Как называется совещание, на котором обсуждаются проблемы, возникшие в процессе работы над проектом?
 1. Демонстрация
 2. Ретроспектива
 3. Планирование спринта
4. Что такое спринт в Scrum?
 1. Итерация, продолжительностью от 1 до 4 недель, во время которой команда работает над определенным количеством задач
 2. Этап, на котором владелец продукта определяет требования к продукту
 3. Совещание, на котором определяются задачи на следующий день
5. Какой из этих принципов не относится к особенностям разработки по Scrum?
 1. Принцип гибкости
 2. Принцип итеративности
 3. Принцип управления рисками

Тест к уроку 6.3. «Насколько Agile влияет на удовольствие от работы»

1. Какой из ритуалов Scrum помогает улучшить взаимопонимание с заказчиком и получить от него обратную связь по результатам выполненной работы?
 1. Daily Standup
 2. Демонстрация
 3. Ретроспектива
 4. Планирование
1. Кто из участников Scrum ответственен за организацию эффективной коммуникации в команде?
 1. Владелец продукта
 2. Скрам-мастер
 3. Разработчик
 4. Тестирующий
2. Что из перечисленного оказывает большее влияние на удовольствие от работы в Agile-команде?
 1. Тщательный контроль со стороны владельца продукта
 2. Точные дедлайны для всей команды
 3. Увеличение автономии участников команды
 4. Регулярные тимбилдинги (неформальные встречи команды)
3. Какой из инструментов позволяет лучше визуализировать работу команды с текущими задачами?
 1. Доска задач
 2. Бэклог спринта
 3. Чат команды в Slack
 4. Стикеры на окне
4. Почему проведение ежедневных встреч с командой важно для успешной работы Agile-команды?
 1. Встречи помогают скрам-мастеру общаться с командой и регулировать конфликты, как только они возникли
 2. Встречи помогают уточнять задачи, увеличивать прозрачность процесса и делиться возникающими проблемами

3. Встречи нужны, чтобы владелец продукта был в курсе прогресса команды и мог эффективно управлять командой
4. Встречи не влияют на эффективность работы команды и могут быть заменены отчетом, который рассылается по почте.

Тест к уроку 6.4. «Анти-Паттерны Agile – команд»

1. Что из перечисленного относится к анти-паттернам Agile-команд?
 1. Ежедневные встречи для обсуждения прогресса команды
 2. Водопадные спринты
 3. Демонстрация готового инкремента заказчику
 4. Планирование бэклога спринта
2. Какая система оценки задач чаще всего используется в Scrum?
 1. В часах
 2. В баллах
 3. В Story points
 4. В Bucket System
3. Бэклог продукта в Scrum это?
 1. Набор функций, которые будут реализованы в текущем спринте
 2. Отчет о выполненных задачах по окончании спринта
 3. Набор функций, которые должны быть реализованы в процессе разработки
 4. Готовый продукт, который команда демонстрирует заказчику
4. Какова продолжительность ежедневных встреч Команды Разработки (Daily Scrum)?
 1. Не более 5 минут
 2. Не более 15 минут
 3. Не более 30 минут
 4. Время не ограничено
5. Для чего используется Burndown график?
 1. Для планирования задач на текущий спринт
 2. Для урегулирования конфликтов в команде
 3. Для формирования требований
 4. Для отслеживания прогресса команды в рамках спринта

Тест к теме 6. Agile в командах

1. В чем состоит главный недостаток водопадной модели разработки ПО?
 1. Она не предоставляет возможности для внесения изменений в процессе разработки
 2. Она не является эффективной при работе в команде
 3. Она не подходит для больших проектов
 4. Она не позволяет быстро реагировать на изменения требований
 5. Она не гарантирует качество конечного продукта
2. Что представляет собой V-образная модель разработки ПО?
 1. Модель, в которой все этапы процесса разработки выполняются последовательно
 2. Модель, в которой весь процесс разработки разбит на отдельные этапы, которые выполняются параллельно
 3. Модель, в которой каждый этап процесса разработки тесно связан с этапом тестирования
 4. Модель, в которой процесс разработки выполняется в циклическом порядке
 5. Модель, в которой разработка ПО осуществляется с помощью специальных инструментов.
3. Какие роли определены в команде Scrum?

1. Владелец продукта, разработчик, тестировщик
2. Владелец продукта, scrum-мастер, команда разработки
3. Руководитель, разработчик, аналитик
4. Как называется совещание, на котором обсуждаются проблемы, возникшие в процессе работы над проектом?
 1. Демонстрация
 2. Ретроспектива
 3. Планирование спринта
5. Какой из ритуалов Scrum помогает улучшить взаимопонимание с заказчиком и получить от него обратную связь по результатам выполненной работы?
 1. Daily Standup
 2. Демонстрация
 3. Ретроспектива
 4. Планирование
6. Что из перечисленного оказывает большее влияние на удовольствие от работы в Agile-команде?
 1. Тщательный контроль со стороны владельца продукта
 2. Точные дедлайны для всей команды
 3. Увеличение автономии участников команды
 4. Регулярные тимбилдинги (неформальные встречи команды)
7. Почему проведение ежедневных встреч с командой важно для успешной работы Agile-команды
 1. Встречи помогают скрам-мастеру общаться с командой и регулировать конфликты, как только они возникли
 2. Встречи помогают уточнять задачи, увеличивать прозрачность процесса и делиться возникающими проблемами
 3. Встречи нужны, чтобы владелец продукта был в курсе прогресса команды и мог эффективно управлять командой
 4. Встречи не влияют на эффективность работы команды и могут быть заменены отчетом, который рассылается по почте.
8. Что из перечисленного относится к анти-паттернам Agile-команд?
 1. Ежедневные встречи для обсуждения прогресса команды
 2. Водопадные спринты
 3. Демонстрация готового инкремента заказчику
 4. Планирование бэклога спринта
9. Бэклог продукта в Scrum это?
 1. Набор функций, которые будут реализованы в текущем спринте
 2. Отчет о выполненных задачах по окончании спринта
 3. Набор функций, которые должны быть реализованы в процессе разработки
 4. Готовый продукт, который команда демонстрирует заказчику
10. Для чего используется Burndown график?
 1. Для планирования задач на текущий спринт
 2. Для урегулирования конфликтов в команде
 3. Для формирования требований
 4. Для отслеживания прогресса команды в рамках спринта

Итоговое тестирование

1. Что такое Docker?
 1. Виртуальная машина
 2. Контейнерная платформа
 3. Облачный репозиторий
 4. Система управления базой данных

2. Что необходимо, чтобы переместить образы docker в хранилище?
 1. Использовать команду docker run
 2. Сжать их в размере меньше чем 2 mb
 3. Остановить все запущенные контейнеры
 4. Пометить их именем хранилища и версией образа
3. Что не является сервисом в Docker Compose?
 1. Контейнер
 2. Порт
 3. Том
 4. Среда
4. Что не входит в план развертывания проекта в Docker?
 1. Установка Docker
 2. Проверка Docker
 3. Обновление Docker
 4. Настройка Docker
5. Где присутствует Docker Compose?
 1. В настройках Docker
 2. В развертывании контейнера
 3. В перемещении образа в репозиторий
 4. Нигде, это отдельный файл со своими настройками
6. Что такое SQL?
 1. язык структурированных запросов
 2. язык гипертекстовой разметки
 3. это язык таблицы стилей
 4. язык веб-программирования
7. Что будет результатом данного запроса?
 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 записи, в которых продолжительность фильмов от полутора до двух часов
8. 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
9. Сколько строк будет у результирующей таблицы, если в исходных таблицах было 7 и 3 значений соответственно?
 1. 7
 2. 3
 3. 10
 4. 21
10. Чему эквивалентно такое условие $x \in \text{ALL}(5, 13)$?
 1. $x > 13$
 2. $x < 5$
 3. $(x = 5) \text{ AND } (x = 13)$
 4. $x \text{ NOT IN } (5, 13)$
11. Какая структура данных является неизменяемой?
 1. Список (list)
 2. Словарь (dict)

3. Кортеж (tuple)
4. Множество (set)
12. Как добавить строку к другой строке в Python?
 1. `str1 = "hello"`
`str2 = "world"`
`string = str1 + str2`
 2. `str1 = "hello"`
`str2 = "world"`
`string = str1.append(str2)`
 3. `str1 = "hello"`
`str2 = "world"`
`string = str1.expand(str2)`
 4. невозможно добавить строку к другой строке в Python
13. Что из следующего верно для словарей Python?
 1. Доступ к элементам осуществляется по их положению в словаре.
 2. Доступ к словарям осуществляется по ключу.
 3. Словарь может содержать любой тип объекта, кроме другого словаря.
 4. Словари изменяемы.
14. Что из перечисленного НЕ является принципом ООП?
 1. Инкапсуляция
 2. Наследование
 3. Полиморфизм
 4. Все является
15. Какие процессы доступа к атрибутам позволяет определить property?
 1. Чтение
 2. Запись
 3. Удаление
 4. Все перечисленные
16. Что такое Selenium?
 1. Технология, которая экономит ресурсы и время
 2. Позволяет автоматизировать всё, что угодно
 3. Набор инструментов для анализа покрытия кода тестами
 4. Инструменты для автоматизации тестирования
17. Что такое фикстура?
 1. Функция, которая запускается до или после тестовых функций
 2. Это то, что пишут не в файле `conftest.py`
 3. Это вид автоматизированного тестирования
 4. Это параметр функции `pytest`
18. Что делает этот код?


```
driver.find_element(AppiumBy.ID, 'com.google.android.calculator:id/op_add').click()
```

 1. Кликает по элементу
 2. Находит элемент по классу
 3. Находит элемент по ID и кликает по нему
 4. Код не работает
19. Какие сущности в Postman используются для обращения к API?
 1. Элементы
 2. Классы
 3. Запросы
 4. Тесты
20. Согласно какому паттерну тест должен быть из трех блоков?
 1. Page Object
 2. Параметризация

3. AAA
 4. AAB
21. Кто такой тимлид?
1. специалист, который переводит бизнес требования пользователей в технические требования для разработчиков
 2. специалист, который составляет тестовую стратегию и проводит тестирование продукта
 3. специалист, отвечающий за стратегию разработки и подбор тех команды
 4. специалист, отвечающий за процесс сборки и доставки изменений кода от локальных файлов в различные окружения
22. Зачем нужно расставлять приоритеты на разработку функциональности?
1. чтобы ваша любимая задача была выполнена самой первой
 2. чтобы задача от руководства была выполнена самой первой
 3. чтобы выявить самые важные задачи для успеха продукта и взять их в работу в первую очередь, учитывая при этом сложность выполнения
 4. чтобы выявить самые простые задачи и взять их в работу в первую очередь, отложив самые сложные на потом
23. Выберите вариант, где указаны все критерии хорошо поставленной цели?
1. конкретная, измеримая, достижимая
 2. соответствие стратегии компании и ограниченность по времени
 3. конкретная, измеримая, достижимая, в соответствии с глобальной стратегией компании и ограничена во времени (правильный ответ)
 4. конкретная, актуальная, простая, в соответствии с глобальной стратегией компании и ограничена во времени
24. Какой вопрос стоит задать сотруднику на встрече по постановке личной цели?
1. Когда вы собираетесь рожать?
 2. Что вы делали вчера вечером?
 3. Какие навыки вы используете в работе больше остальных?
 4. Куда бы вы хотели уйти работать?
25. Что из перечисленного не является хорошей стратегией решения конфликта?
1. инициация дискуссии между конфликтующими сторонами
 2. защита своего любимого сотрудника, даже если он неправ
 3. сохранение нейтралитета, фокусирование на фактах
 4. поиск решения проблемы безотносительно того, кто виноват
26. В чем состоит главный недостаток водопадной модели разработки ПО?
1. Она не предоставляет возможности для внесения изменений в процессе разработки
 2. Она не является эффективной при работе в команде
 3. Она не подходит для больших проектов
 4. Она не позволяет быстро реагировать на изменения требований
 5. Она не гарантирует качество конечного продукта
27. Какие роли определены в команде Scrum?
1. Владелец продукта, разработчик, тестировщик
 2. Владелец продукта, scrum-мастер, команда разработки
 3. Руководитель, разработчик, аналитик
28. Какой из ритуалов Scrum помогает улучшить взаимопонимание с заказчиком и получить от него обратную связь по результатам выполненной работы?
1. Daily Standup
 2. Демонстрация
 3. Ретроспектива
 4. Планирование

29. Что из перечисленного относится к анти-паттернам Agile-команд?
1. Ежедневные встречи для обсуждения прогресса команды
 2. Водопадные спринты
 3. Демонстрация готового инкремента заказчику
 4. Планирование бэклога спринта
30. Для чего используется Burndown график?
1. Для планирования задач на текущий спринт
 2. Для урегулирования конфликтов в команде
 3. Для формирования требований
 4. Для отслеживания прогресса команды в рамках спринта