

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. УГЛУБЛЕННЫЙ УРОВЕНЬ**

**Документ:** Свидетельство

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

**Срок обучения:** 15 недель

**Объем программы:** 152 ак. ч.

Москва, 2023 г.

## Оглавление

1. ОБЩИЕ ПОЛОЖЕНИЯ.....	3
1.1. Назначение программы.....	3
1.2. Нормативные документы, регламентирующие разработку дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	3
2.1. Цели дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.2. Требования к слушателю дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.3. Срок освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
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.....	17
4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ - ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ.....	18
4.1. Требования к квалификации педагогических кадров, представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса.....	18
4.2. Материально-технические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	18
4.3. Информационные и учебно-методические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	20
4.4. Общие требования к организации образовательного процесса.....	21
5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	21
5.1. Формы контроля знаний и требования к его проведению.....	21
5.2. Критерии оценки знаний слушателей.....	22
5.3. Фонд оценочных средств.....	24

# 1. ОБЩИЕ ПОЛОЖЕНИЯ

## 1.1. Назначение программы

**Дополнительная общеобразовательная программа дополнительная общеразвивающая программа «Тестирование ПО. Углубленный уровень»** направлена на получение практических навыков и теоретических знаний, необходимых для тех, кто хочет больше узнать про тестирование программного обеспечения и познакомиться с инструментами тестирования ПО.

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

# 2. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

## 2.1. Цели дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Основная цель программы – дать слушателям базовые и углубленные знания о тестировании ПО, а также предложить различные инструменты тестирования и рассказать об автоматизации тестирования на популярном языке Python.

### Задачи дисциплины

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

- об основных языках программирования: JavaScript, CSS и Python;
- о тестировании безопасности веб-приложений;
- об автоматизации тестирования на языке программирования Python;
- о различных инструментах для тестирования ПО.

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

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

Возраст слушателей: 18 лет и старше.

## 2.3. Срок освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Нормативный срок освоения программы – 15 недель.

## 2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Объем образовательной программы составляет 152 академических часа. Из них 77 часов - самостоятельная работа, 75 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:

- основные сведения о языках программирования: JavaScript, CSS и Python;
- основы использования Docker;
- особенности использования тестирования API: назначение, процесс, применимость.

### Уметь:

- использовать DevTools популярных браузеров;
- выбирать различные техники тестирования;
- составлять корректные HTML-запросы в Allure;
- создавать проекты на Python для покрытия API тестами в PyCharm.

### Владеть:

- навыками тестирования безопасности веб-приложений;
- навыками создания автотестов для API и веб-приложений на языке Python;
- навыками работы на контейнерной платформе Docker;
- навыками проведения нагрузочного тестирования.

### 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	
<b>1.</b>	<b>Работа с JavaScript и CSS</b>	<b>40</b>	<b>20</b>	<b>20</b>	<b>Зачет</b>
1.1	Введение в JavaScript	4	2	2	Тестирование
1.2	Консоль разработчика	4	2	2	Тестирование
1.3	Типы данных	4	2	2	Тестирование
1.4	Объекты: основы	4	2	2	Тестирование
1.5	Обработка ошибок	4	2	2	Тестирование
1.6	Что такое CSS	4	2	2	Тестирование
1.7	Синтаксис CSS	4	2	2	Тестирование
1.8	Способы подключения CSS	4	2	2	Тестирование
1.9	Селекторы	4	2	2	Тестирование
1.10	Как работать в DevTools	4	2	2	Тестирование
<b>2.</b>	<b>Тестирование безопасности веб-приложений</b>	<b>16</b>	<b>8</b>	<b>8</b>	<b>Зачет</b>
2.1	Основы безопасности веб-приложений	4	2	2	Тестирование
2.2	Управление cookie, сессиями	4	2	2	Тестирование
2.3	SQL инъекции	4	2	2	Тестирование
2.4	Тестирование XSS-уязвимости	4	2	2	Тестирование
<b>3.</b>	<b>Основы языка Python</b>	<b>28</b>	<b>14</b>	<b>14</b>	<b>Зачет</b>
3.1	Типы и структуры данных	4	2	2	Тестирование

3.2	Основные конструкции языка	4	2	2	Тестирование
3.3	Работа со строками	4	2	2	Тестирование
3.4	Работа со списками	4	2	2	Тестирование
3.5	Работа со словарями и коллекциями	4	2	2	Тестирование
3.6	Функции	4	2	2	Тестирование
3.7	Классы и объекты	4	2	2	Тестирование
<b>4.</b>	<b>Автоматизация на Python</b>	<b>24</b>	<b>12</b>	<b>12</b>	<b>Зачет</b>
4.1	Как тестировать веб-приложения и API. Тестовые фреймворки	4	2	2	Тестирование
4.2	Тестирование с помощью PyTest	4	2	2	Тестирование
4.3	Использование фикстур в PyTest	4	2	2	Тестирование
4.4	PyTest	4	2	2	Тестирование
4.5	Разработка автотестов на API	4	2	2	Тестирование
4.6	Разработка автотестов на Веб	4	2	2	Тестирование
<b>5.</b>	<b>CI/CD. Запуск тестов Jenkins</b>	<b>12</b>	<b>6</b>	<b>6</b>	<b>Зачет</b>
5.1	Установка Jenkins	4	2	2	Тестирование
5.2	Разработка bash-скриптов	4	2	2	Тестирование
5.3	Запуск тестов на Jenkins	4	2	2	Тестирование
<b>6.</b>	<b>Автоматизация тестирования при помощи Allure</b>	<b>10</b>	<b>5</b>	<b>5</b>	<b>Зачет</b>
6.1	Основы Allure и общие принципы отчетности	2	1	1	Тестирование
6.2	Возможности Allure и его установка	2	1	1	Тестирование
6.3	Написание тестов и сборка HTML-отчета	2	1	1	Тестирование
6.4	Конфигурация для сборки отчета	2	1	1	Тестирование

6.5	Работа с отчетами	2	1	1	Тестирование
<b>7.</b>	<b>Нагрузочное тестирование</b>	<b>8</b>	<b>4</b>	<b>4</b>	<b>Зачет</b>
7.1	Основы нагрузочного тестирования	4	2	2	Тестирование
7.2	Jmeter	4	2	2	Тестирование
<b>8.</b>	<b>Docker</b>	<b>12</b>	<b>8</b>	<b>4</b>	<b>Зачет</b>
8.1	Введение в Docker	3	2	1	Тестирование
8.2	Основные команды Docker	3	2	1	Тестирование
8.3	Запуск сервисов с помощью Docker Compose	3	2	1	Тестирование
8.4	Разворачивание проекта Docker	3	2	1	Тестирование
<b>ИТОГОВАЯ АТТЕСТАЦИЯ</b>		<b>2</b>	<b>—</b>	<b>2</b>	<b>Зачет</b>
<b>Всего:</b>		<b>152</b>	<b>77</b>	<b>75</b>	<b>—</b>

## 3.2. Содержание

### Тема 1. Работа с JavaScript и CSS

#### 1.1. Введение в JavaScript

- Введение в JavaScript
- Синтаксис JavaScript
- Базовые конструкции (оператор, условие, цикл, функции)
- Практическое задание: нужно сложить два положительных числа. Если хотя бы одно из чисел отрицательное, выводить в консоль сообщение «Отрицательные числа складывать не буду». И создать цикл от 1 до 6 и вывести в консоль только четные числа

#### 1.2. Консоль разработчика

- Консоль разработчика
- Логирование в JavaScript
- Практическое задание: поработать в консоли разработчика и создать html-файл с js-кодом внутри

#### 1.3. Типы данных

- Типы данных в JavaScript
- Преобразование типов данных
- Операции с базовыми типами данных
- Практическое задание:
  1. Петя плохо выучил JavaScript, написал программу и забыл проверить. Найдите ошибки в программе Пети и исправьте их.
  2. Написать свою программу с использованием всех типов данных.
  3. Дополнительно: по аналогии с оператором «Побитовое И» сделайте разбор остальных побитовых операторов.

#### 1.4. Объекты: основы

- Класс, свойства и методы
- Работа с классами
- Практическое задание: Создайте класс Student со свойствами: имя, оценка. Создайте пару студентов и выведите значения (имена и их оценки) в консоль. В созданном классе Student создайте метод, который выводит в консоль «Молодец!», если оценка у студента «5».

#### 1.5. Обработка ошибок

- Ошибки. Основные пользовательские ошибки.
- Перехват и обработка ошибок.
- Генерация ошибок.
- Практическое задание:
  1. Напишите код, вызывающий все типы пользовательских ошибок, рассмотренные в уроке. В консоли разработчика должны быть вызваны все типы ошибок.
  2. Напишите код, который генерирует ошибки всех типов пользовательских ошибок при помощи оператора throw.

#### 1.6. Что такое CSS

- Атрибуты стиля элементов
- Способы встраивания CSS
- Основные селекторы



- Структура CSS-файла
- Практическое задание: создание HTML-файла с текстом и добавление новых фраз с определенными критериями

### 1.7. Синтаксис CSS

- Описание стиля CSS-файле
  - Внешний способ встраивания CSS
  - Основные свойства CSS
  - Практическое задание:
1. Найдите и перечислите все возможные значения-типы атрибута rel (помимо stylesheet). Опишите назначение каждого типа.
  2. Создайте файл стилей с расширением CSS, реализуйте все примеры из урока, используйте все изученные свойства.

### 1.8. Способы подключения CSS

- Порядок применения стиля
- Способы встраивания CSS
- Способы определения стиля элемента
- Практическое задание: найти ошибки в HTML-файле и исправить их.

### 1.9. Селекторы

- Селекторы
- CSS-селекторы
- XPath-селекторы
- Функции в XPath-селекторах
- Практическое задание: найти и составить список самых популярных псевдоклассов. Составить выражения для поиска элементов с текстом «параграф» двумя способами: при помощи CSS и при помощи XPath.

### 1.10. Как работать с DevTools

- Chrome DevTools
- Firefox DevTools
- Практическое задание: на любом сайте просмотрите все вкладки в инструментах разработчика, стараясь извлечь максимум полезной информации о веб-приложении. Кратко написать, что полезного вы узнали о сайте при помощи инструментов разработчика.

## Тема 2. Тестирование безопасности веб-приложений

### 2.1. Основы безопасности веб-приложений

- Архитектура веб-приложений
- Работа с формами
- Передача данных с форм
- Практическое задание: Установите приложение bWAPP. Откройте форму HTML Injection — Reflected (GET). Сформируйте произвольный HTML-код и внедрите его в форму. Откройте форму HTML Injection — Stored (Blog). Сформируйте произвольный HTML-код и внедрите его в форму.

## 2.2. Управление cookie, сессиями

- Что такое cookie
- Работа с cookie (удаление, редактирование cookie)
- Практическое задание: в уроке 12, в параграфе 12.1 «Что такое cookies» рассмотрели пример подмены cookie-файла пользователя на cookie-файл администратора, в результате чего пользователь получил доступ к функциям администратора и смог выполнить сброс настроек системы. Сделайте корректное описание дефекта для данной ситуации.

## 2.3. SQL инъекции

- Что такое SQL инъекции
- Принцип действия атаки путем внедрения кода SQL
- Поиск SQL injection
- Практическое задание: в приложении bWAPP нужно вывести значения из таблицы heroes.

## 2.4. Тестирование на XSS-уязвимости

- Что такое XSS-уязвимость
- Как устроена уязвимость
- Чем опасна XSS-уязвимость
- Где искать XSS и как с ней бороться
- Практическое задание: сформировать произвольный JavaScript-код и провести его внедрение в приложении bWAPP. Выполните отраженную XSS-атаку и хранимую XSS-атаку.

## Тема 3. Основы языка Python

### 3.1. Типы и структуры данных

- Типы данных
- Структуры данных
- Практическое задание: написать программу, которая запрашивает у пользователя его имя, возраст, рост и вес. Написать 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.

### 3.2. Основные конструкции языка

- Переменные, константы и оператор присваивания
- Условный оператор в языке Python: if, if-else, if-elif-else
- Оператор условного выражения (тернарный оператор)
- Циклы в языке Python: for, while.
- Практическое задание: создать программу на Python, которая будет вычислять площадь круга. Задать значение радиуса круга в переменную и используйте оператор присваивания для присвоения значения. Использовать константу для хранения значения числа  $\pi$ . Использовать формулу  $S = \pi r^2$  для вычисления площади круга, где  $S$  - площадь,  $r$  - радиус. Вывести результат на экран с помощью функции print().

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

### 3.4. Работа со списками

- Операции со списками
- Списки в Python.
- Индексация списков.
- Вложенность списков.
- Методы списков.
- Практическое задание: написать программу, которая принимает на вход список чисел и выводит на экран два списка: список четных чисел и список нечетных чисел из входного списка.

### 3.5. Работа со словарями и коллекциями

- Определение словаря.
- Доступ к значениям словаря.
- Индексация в словарях.
- Ограничения на ключи.
- Методы словарей.
- Операции с коллекцией. Определение коллекции.
- Практическое задание: написать программу, которая будет работать с информацией о книгах в библиотеке. Написать функцию, на вход которой поступает список целых чисел и которая должна вернуть новый список, содержащий только четные целые числа, отсортированные в порядке возрастания.

### 3.6. Функции

- Использование функций
- Определение функции. Важность функций.
- Вызовы функций и их определение.
- Передача аргументов в функцию.
- Возвращаемое значение.
- Практическое задание: написать программу-калькулятор, которая будет выполнять математические операции (+, -, \*, /) в зависимости от выбора пользователя и выводить результат. Для реализации программы использовать функции.

### 3.7. Классы и объекты

- Классы и объекты
- Классы в Python
- Объекты в Python
- Практическое задание: изменить стандартный объем стакана кофе на 250 мл. Добавить возможность создать стакан сразу наполненным некоторым количеством кофе.

## Тема 4. Автоматизация на Python

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

- Принципы автоматизации тестирования

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

#### 4.2. 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
```

#### 4.3. 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.

#### 4.4. 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 "dev"`. Результат должен быть в виде кода на языке Python.

#### 4.5. Разработка автотестов на API

- Общие принципы тестирования API
- Что такое API
- Особенности взаимодействия с API
- Процесс тестирования API
- Автоматизация тестирования API с инструментом PyCharm.
- Подготовка окружения
- Тестирование API в PyCharm.
- Практическое задание.
  1. Повторите действия в уроке: напишите тесты на создание и удаление фильма, на получение всех фильмов.
  2. Добавьте `assert` (проверки на каждый тест): проверку ответа API на созданный фильм, проверку ответа удаленного фильма, проверку ответа списка фильмов.  
Результат пришлите в виде кода тестов, скопированных из среды разработки.

#### 4.6. Разработка автотестов на Веб

- Как устроены веб-приложения
- Фреймворки автоматизации тестирования
- Написание тестов на Selenium+Python на веб-приложение

### Тема 5. CI/CD. Запуск тестов в Jenkins

#### 5.1. Установка Jenkins

- Знакомство с Jenkins
- Установка
- Настройка
- Практическое задание: скачать плагины Allure и Labelled pipeline steps. Создать первую job и проверить ее результат. Поискать информацию о том, чем отличается Jenkins от основных аналогов (GitLab CI/CD и Travis CI), и сравнить эти сервисы.

#### 5.2. Разработка bash-скриптов

- Разработка bash-скриптов
- Практическое задание: написать bash-скрипт, который печатает числа от 1 до 10, используя цикл `for`. Написать Bash-скрипт, который предлагает пользователю ввести положительное целое число, а затем распечатайте числа от 1 к этому целому числу с помощью цикла `while`.

#### 5.3. Запуск тестов на Jenkins

- Запуск тестов на Jenkins
- Создание тест-среды, процесс работы с тестами.
- Лучшие практики проведения тестирования в Jenkins.
- Практическое задание: создать тесты в Jenkins используя инструкцию

## **Тема 6. Автоматизация тестирования при помощи Allure**

### **6.1. Основы Allure и общие принципы отчетности**

- Назначение Allure - автоматизация тестирования, создание удобных и простых отчетов.
- Инструменты: Allure Report и Allure Testops.
- Процесс работы с Allure.
- Назначение отчетности в автоматизации. Виды тестирования.
- Инструменты отчетности в Python.
- Практическое задание: опишите, какой на ваш взгляд должна быть идеальная система отчетности по тестам: что в ней должно быть, как может выглядеть, из чего она должна состоять, ее особенности, какие должны быть метрики, например, число пройденных тестов.

### **6.2. Возможности Allure и его установка**

- Подготовка окружения для работы с Allure.
- Возможности Allure для тестирования.
- Установка Allure.
- Практическое задание: изучите разделы документации про возможности allure report. Выделите самое важное, что поняли из этой документации. Подумайте, какие фичи вам кажутся самыми нужными. Опишите назначение каждой фичи. Часть фич была разобрана на занятии, поэтому материалы занятий помогут вам при выполнении практической работы. Результат пришлите в виде текста.

### **6.3. Написание тестов и сборка HTML-отчета**

- Формирование HTML-отчета.
- Подключение зависимостей.
- Связывание проекта с allure report.
- Формирование html-отчета.
- Написание тестов на Python + Selenium.
- Практическое задание: повторите действия в уроке, сформировав отчет по прогону тестов. Критичность сценариев определена следующим образом: CRITICAL — для сценария с поиском дудла, TRIVIAL — для всех остальных. По примеру <https://github.com/allure-framework/allure-python/blob/master/allure-pytest/examples/label/severity/severity.rst> навесьте теги severity на каждый тест. Регенерируйте allure-отчет. Убедитесь, что severity на графике отображается правильно. Результат пришлите в виде скриншота, где видны severity по каждому тесту.

### **6.4. Конфигурация для сборки отчета**

- Системы непрерывной интеграции.
- Установка и настройка Jenkins.
- Установка плагинов. Создание конфигурации в Jenkins.
- Настройка инструментов. Запуск конфигурации.
- Практическое задание:
  1. Установите Jenkins на свой компьютер, используя материалы урока.

2. Настройте Jenkins-конфигурацию, чтобы запускались тесты на Python (проект из предыдущего занятия) и формировался HTML-отчет при помощи Allure Report.
3. Переименуйте проект и задайте описание произвольным образом. Результат пришлите в виде скриншота открытой страницы с проектом в Jenkins.

## 6.5. Работа с отчетами

- Работа с отчетами в Allure.
- Вкладки отчета по прогону тестов: Overview, Categories, Suites, Graphs, Timeline, Behaviors, Packages.
- Выгрузка информации по тестам из вкладок в csv-файл.
- Улучшение html-отчета при помощи специальных маркеров allure (allure-аннотаций).
- Практическое задание:
  1. Навесьте на первые два теста маркеры suite, на третий — sub\_suite, на остальные тесты — parent\_suite.
  2. Пересоберите отчет. В отчете должны появиться изменения. Найдите в отчете хотя бы в одном месте результат примененных изменений.
  3. Результат пришлите в виде трех скриншотов на каждый маркер (первые увиденные места, где было изменение по каждому маркеру).

## Тема 7. Нагрузочное тестирование

### 7.1. Основы нагрузочного тестирования

- Введение в нагрузочное тестирование
- Назначение нагрузочного тестирования
- Основные этапы нагрузочного тестирования, метрики
- Практическое задание: перечислите инструменты по нагрузочному тестированию и кратко расскажите о каждом из них. Познакомьтесь с инструментом Grafana

### 7.2. Jmeter

- Знакомство с Jmeter
- Создание первого теста
- Throughput controller
- Controller thread group
- Проектирование сценариев
- Работа с данными в Jmeter
- Отчеты
- Практическое задание: найдите границу, когда бэкенд будет выдавать 50% и больше неудачных запросов. Найдите три статуса кода этих ошибок и распишите, что они значат.

## Тема 8. Docker

### 8.1. Введение в Docker

- Что такое Docker и Docker image («образ»).
- Как работает Docker.
- Чем может быть полезен Docker. Назначение Docker.
- Практическое задание: 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.

## 8.2. 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.

## 8.3. 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: создать и запустить файл docker compose.

## 8.4. 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
- 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.

### 3.3 Календарный учебный график

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

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

Примерный режим занятий слушатели определяют самостоятельно, но не более установленного срока обучения. Примерный график освоения программы:

№	Темы / недели	ВР	-	с	с	<	с	с	с	с	с	с	с	с	с	с



1.	Работа с JavaScript и CSS	РП	5	5	5	5												
		СР	5	5	5	5												
2.	Тестирование безопасности веб-приложений	РП					8											
		СР					8											
3.	Основы языка Python	РП						7	7									
		СР						7	7									
4.	Автоматизация на Python	РП								6	6							
		СР								6	6							
5.	CI/CD. Запуск тестов Jenkins	РП										6						
		СР										6						
6.	Автоматизация тестирования при помощи Allure	РП											5					
		СР											5					
7.	Нагрузочное тестирование	РП												4				
		СР												4				
8.	Docker	РП													2	2		
		СР													4	4		
9.	<b>Итоговая аттестация</b>	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** на базе высшего образования (специалитета или магистратуры) — 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. Материально-технические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.**

Организация проводит занятия по адресу: г. Москва, ул. Сушевский 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.

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;
- видеочамера (максимальное разрешение видео — не менее 3840 x 2160).

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

1. Трансляция вебинара в режиме реального времени.
2. Хранение, систематизация записей вебинаров, с предоставлением слушателям возможности просмотра записи онлайн.
3. Хранение, систематизация и доступ к скачиванию материалов учебных программ.
4. Напоминание слушателям 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 слушателей.
12. Техническая доступность услуги не менее 99,8% времени.
13. Устойчивость при проведении вебинара при единовременном 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 видеосоинформации на персональные компьютеры слушателей реализована при скорости интернет-соединения не менее 134 кбит/с.

Основные функции системы дистанционного обучения 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.

## **Программное обеспечение для обеспечения обучения с использованием дистанционных образовательных технологий, электронного обучения:**

Система дистанционного обучения.

Наличие библиотеки учебных материалов в электронном виде.

### **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 [Зяц 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**

Дистанционные образовательные технологии применяются частично.

В учебном процессе могут использоваться следующие организационные формы учебной деятельности:

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

### **5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

#### **5.1. Формы контроля знаний и требования к его проведению**

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

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** - процедура, проводимая с целью установления уровня знаний, слушателей с учетом прогнозируемых результатов обучения и требований к результатам освоения образовательной программы.

**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, который демонстрирует качество полученных навыков, определяет уровень усвоения слушателями учебного и практического материала и охватывает все содержание, установленное соответствующей дополнительной общеобразовательной программой - дополнительной общеразвивающей программой.

Итоговая аттестация проводится в форме тестирования, в том числе с использованием технических возможностей системы СДО.

**Лицам, успешно прошедшим итоговую аттестацию, выдается Свидетельство.**

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

При разработке оценочных средств для контроля качества изучения дисциплин учтены все виды связей между включенными в них знаниями, умениями, навыками, позволяющие установить качество сформированных у слушателей компетенций по видам деятельности и степень общей готовности выпускников к применению знаний на практике.

В качестве оценочных средств для проведения текущего контроля и аттестации в целом по всем дисциплинам учебного плана используются как инновационные формы контроля, так и классические.

Фонды оценочных средств соответствуют целям и задачам дополнительной общеобразовательной программы, учебному плану и обеспечивают оценку качества компетенций, приобретаемых слушателям.

## 5.2. Критерии оценки знаний слушателей

<b>Предмет оценивания (компетенции)</b>	<b>Объект оценивания (навыки)</b>	<b>Показатель оценки (знания, умения)</b>
Работа на языках программирования: JavaScript, CSS и Python	<ul style="list-style-type: none"> <li>- Применение знаний по основам языков программирования для выполнения разных задач при тестировании программного обеспечения</li> <li>- Понимание разницы в функционале языков программирования JavaScript, CSS и Python</li> </ul>	<p><b>Знания:</b></p> <ul style="list-style-type: none"> <li>- Основы языков программирования JavaScript, CSS и Python</li> <li>- Базовые концепции программирования, их применение при тестировании ПО</li> </ul> <p><b>Умения:</b></p> <ul style="list-style-type: none"> <li>- Читать код созданный на языках программирования: JavaScript, CSS и Python</li> <li>- Соблюдать минимальные стандарты оформления кода</li> </ul>
Проведение тестирования безопасности веб-	- Понимание уязвимости веб-приложений	<p><b>Знания:</b></p> <ul style="list-style-type: none"> <li>- Основы безопасности веб-</li> </ul>

приложений	- Проведение проверки веб-приложений на наличие уязвимости	приложений - Управление cookie, сессиями <b>Умение:</b> - Проверять веб-приложения на наличие уязвимости - Работать с cookie (удаление, редактирование cookie) - Выполнять поиск и работу с SQL injection, XSS
Осуществление автоматизации тестирования на языке программирования Python	- Автоматизации тестирования на языке программирования Python	<b>Знания:</b> - Основы тестирования веб-приложения и API. - Особенности применение тестового фреймворка Pytest <b>Умения:</b> - Тестирование с помощью PyTest - Разработка автотестов на API - Разработка автотестов на Веб
Использование различных инструментов для тестирования ПО	- Применение при тестировании ПО различных инструментов	<b>Знания:</b> - Основы работы с Docker, Jenkins, Allure - Основы работы с инструментами по нагрузочному тестированию <b>Умения:</b> - Применять при тестировании ПО инструменты Docker - Создавать тесты в Jenkins - Формировать отчеты в Allure - Использовать инструмент по нагрузочному тестированию Jmeter

**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 продемонстрировали 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 продемонстрировали неудовлетворительный уровень знаний и умений слушателя. Менее 80% правильных ответов при решении тестов.

### 5.3. Фонд оценочных средств

#### Тест к уроку «Введение в JavaScript»

1. Какие ключевые слова существуют в JavaScript?
  1. function, if, typeof
  2. javascript, js, language
  3. importer, exporter, how
  4. start, finish, continue
2. Какие типы данных не существуют в JavaScript?
  1. Boolean, Number, String
  2. Null, Object, BigInt
  3. Undefined, Symbol, BigInt
  4. Byte, Long, Char
3. Когда код выполнится без ошибок?
  1. `const test;`  
`console.log(test);`
  2. `var test = 'test';`  
`console.log(tost);`
  3. `const test = 'test';`  
`console.log(test);`
  4. `var test;`  
`console.log(test);`
4. Выберите верное утверждение
  1. JavaScript создан на основе Java
  2. JavaScript поддерживает объектно-ориентированное, компонентно-ориентированное и обобщенное программирование (тут неверное - компонентно-ориентированное)
  3. JavaScript - язык со статической и слабой типизацией
  4. JavaScript поддерживает объектно-ориентированное, функциональное и событийно-ориентированное программирование
5. Когда код выполнится без ошибок?
  1. `var a1, a2 <= 3;`  
`if (a1 <= a2) {`  
`}`
  2. `var a1, a2 = 3;`  
`if a1 <= a2 {`  
`console.log(a2);`  
`}`
  3. `a1, a2 = 3;`



```
    if (a1 <= a2){  
        console.log(a2);  
    }  
4. var a1 = 2;  
   var a2 = 3;
```

#### Тест к уроку «Консоль разработчика»

1. Какую клавишу нужно нажать в Chrome для открытия консоли разработчика?
  1. F12
  2. F11
  3. F10
  4. F9
2. Какие виды сообщений можно увидеть в консоли разработчика?
  1. Debug, Error, Message
  2. Fatal, Error, Log
  3. Message, Info, Log
  4. Error, Warning, Info
3. Как называется объект для доступа к средствам отладки?
  1. Debug
  2. Log
  3. Console
  4. Error
4. Выберите верное утверждение:
  1. Чтобы писать программы
  2. Чтобы посмотреть куки
  3. Для редактирования исходного кода страницы
  4. Для отладки и поиска ошибок
5. Как вызвать вид сообщения «Ошибка» в консоли браузера?
  1. `var error = "error";  
 console.log(error);`
  2. `console.log(new Error);`
  3. `console.log("error");`
  4. `console.error("error");`

#### Тест к уроку «Типы данных»

1. Что будет выведено в консоли?

```
var v1 = Symbol('value2');  
var v2 = Symbol('value2');  
var r = v1 === v2;  
console.log(r);
```

  1. false
  2. true
  3. null
  4. Будет ошибка
2. Для чего нужны типы данных в JavaScript?
  1. Чтобы разграничить диапазоны возможных значений и операций
  2. Чтобы продемонстрировать структуры данных
  3. Для отладки и тестирования программы
  4. Для обозначения операций и операндов
3. Что будет выведено в консоли?

```
var v = true;  
v = false;  
v + true;  
console.log(v);
```

  1. 0

2. true
3. false
4. 1
4. Какие типы данных не существуют в JavaScript?
  1. Boolean, Number, String
  2. Null, Object, BigInt
  3. Undefined, Symbol, BigInt
  4. Byte, Long, Char
5. В каком случае используется тип данных BigInt?
  1. var bi = 5;
  2. var x = 5.05;
  3. let y = 5B;
  4. var bi = 5n;

Тест к уроку «Объекты: основы»

1. Для чего нужны методы класса?
  1. Для совершения действий над объектом
  2. Для хранения информации об объекте
  3. Для создания экземпляра класса
  4. Для вызова действия внутри другого класса
2. Что будет выведено в консоль?
 

```
const car = class MyCar {
  model = "Volvo";
}
var car1 = new MyCar();
console.log(car1.model);
```

  1. Ничего
  2. 'MyCar' has already been declared
  3. Volvo
  4. MyCar is not defined
3. Выберите корректную (-ые) часть (-и) кода:
  1. var my = new My("value");
  2. var myValue = new My("key", "value");
  3. Все корректные
  4. this.something = something;
4. Выберите верное утверждение:
  1. Нельзя создать несколько методов в одном классе
  2. Класс – это генератор
  3. Класс – это класс!
  4. Класс – это функция
5. Когда код выполнится без ошибок?
  1. class M{constructor(n){this.n=n;}}
 

```
console.log(k.n);
```
  2. class M{constructor(n){this.n=n;}}
 

```
console.log(new m("n"));
```
  3. class M{constructor(nn){this.nn=nn;}}
 

```
var s = new M("test it!");
console.log(s.nn);
```
  4. class M{constructor(n){this.n=n;}}
 

```
var k = new M("n");
console.log(k.n);
```

Тест к уроку «Обработка ошибок»

1. Какого типа ошибок не существует в JavaScript?
  1. ExceptionError

2. TypeError
3. URIError
4. ReferenceError
2. Что будет выведено в консоль?  
`decodeURIComponent('%test%');`
  1. Ничего
  2. %test%
  3. test
  4. Uncaught URIError: URI malformed
3. Что будет выведено в консоль?  
`const myError = new Error('Общая пользовательская ошибка!');`  
`console.log(myError.name);`
  1. Общая пользовательская ошибка!
  2. Ошибка
  3. Error
  4. Ничего
4. Выберите верное утверждение:
  1. Направления ошибок в JS – встроенные и системные
  2. Направления ошибок в JS – пользовательские и системные
  3. Направления ошибок в JS – встроенные и логические
  4. Направления ошибок в JS – пользовательские и встроенные
5. Какой код вызовет ошибку SyntaxError?
  1. `var opechatka = ['1', '2']  
delete opechatka[0];`
  2. `var abrakadabra = ['1', '2'];  
delete arr[0];`
  3. `var arr = ['1', '2'];  
delete arr[0];`
  4. `var arr = ['1', '2'];  
delite arr[0];`

#### Тест к уроку «Что такое CSS»

1. Какие скобки используются для задания тегов?
  1. Треугольные
  2. Фигурные
  3. Круглые
  4. Квадратные
2. Что такое атрибут?
  1. Устаревший элемент, который уже не используется
  2. Стил для экрана монитора
  3. То же самое, что тег
  4. То, что указывается внутри скобок тега
3. Какие значения не могут быть у атрибута media?
  1. all, tv, braille
  2. speech, print, screen
  3. tty, tv, tti
  4. projection, handheld, screen
4. Выберите верное утверждение:
  1. CSS нужен, чтобы разметить HTML-теги на странице
  2. CSS нужен, чтобы задать правила для обработки исключений
  3. CSS – это custom style sheets
  4. CSS нужен, чтобы задать цвет, расположение элементов
5. Какого атрибута не существует у тега style?
  1. type

2. media
3. Все перечисленные
4. mime

Тест к уроку «Синтаксис CSS»

1. Какого вида селектор используется в этом коде?

```
span {  
  color: blue;  
}
```

1. Селектор элемента
2. Селектор класса
3. Селектор идентификатора
4. Универсальный селектор

2. Что такое font-size в представленном CSS-файле?

```
p {  
  font-size: 26px;  
}
```

1. Селектор
2. Правило
3. Значение
4. Свойство

3. Какого значения не может быть у свойства color?

1. #fff
2. teal
3. krasnij
4. rgb(255,0,0)

4. Выберите верное утверждение:

1. CSS нужен, чтобы разметить HTML-теги на странице
2. CSS нужен, чтобы задать правила для обработки исключений
3. CSS – это custom style sheets
4. CSS нужен, чтобы описать внешний вид HTML-документа при помощи тегов

5. Каким образом можно выполнить внешнее встраивание CSS?

1. <style>...</style>
2. <link href="stylesheet" ref="css/style.css">
3. Такого встраивания не существует
4. <link rel="stylesheet" href="css/style.css">

Тест к уроку «Способы подключения CSS»

1. Сайт состоит из сотни страниц. Какой способ подключения предпочтительнее использовать? (22352)

1. Внешнее встраивание
2. Внутреннее встраивание
3. Встроенное встраивание
4. Правило @import

2. Какой в итоге будет цвет у текста?

```
<html>  
<head>  
  <style>  
    html { color: yellow; }  
    body { color: purple; }  
    p { color: black; }  
  </style>  
</head>  
<body>  
  <div style="color: grey;"><span style="color: red;">Текст</span></div>
```

- ```
</body>
</html>
```
1. Серый
  2. Чёрный
  3. Жёлтый
  4. Красный
3. Какого значения не может быть у свойства color?
    1. #fff
    2. teal
    3. whiteblue
    4. rgb(255,0,0)
  4. Выберите верное утверждение:
    1. CSS нужен, чтобы разметить HTML-теги на странице
    2. CSS нужен, чтобы задать правила для обработки исключений
    3. CSS – это custom style sheets
    4. CSS нужен, чтобы описать внешний вид HTML-документа при помощи тегов
  5. Какой тег используется для внутреннего встраивания CSS?
    1. link
    2. rel
    3. href
    4. style

#### Тест к уроку «Селекторы»

1. Какого вида селектора не существует?
  1. Math
  2. XPath
  3. CSS
  4. Универсальный
2. Что такое XPath?
  1. Express Path Language
  2. X Path Language
  3. XML-разметка документа
  4. Язык запросов к элементам страницы
3. Какой XPath не найдет элемент на странице?

```
<div class="test">
  <p>Параграф единственный и неповторимый!</p>
</div>
```

  1. /html/body/div/p
  2. //div/p
  3. body/div/p
  4. //body/\*/p
4. Что не умеет XPath, что умеет CSS?
  1. Искать по селектору идентификатора
  2. Искать по селектору класса
  3. Искать по универсальному селектору
  4. Искать по селектору псевдокласса
5. Что чего придуманы псевдоклассы?
  1. Чтобы применять к элементам стили, которые не входят в обычные CSS-стандарты
  2. Чтобы применять стили к элементам, находящимся на одном уровне вложенности
  3. Чтобы применять стили ко всем элементам на странице сразу

4. Чтобы применять стили к элементам, находящимся в определённом состоянии или порядке

Тест к уроку «Как работать в DevTools»

1. Как открыть консоль разработчика?
  1. Нажать Ctrl+Shift+I
  2. Нажать Ctrl+Shift+L
  3. Нажать F2
  4. Нажать на меню-троеточие и выбрать Вспомогательные инструменты
2. Какие вкладки не скрыты по умолчанию?
  1. Регистратор, Статистика производительности
  2. Быстрый исходный код, Поиск
  3. Анимации, Покрытие
  4. Сеть, Источники
3. Где найти справку по сочетаниям клавиш?
  1. Правый верхний угол – кнопка-троеточие – Дополнительные настройки – Сочетания клавиш
  2. Нажать F1
  3. Правый верхний угол – меню-троеточие - Сочетания клавиш
  4. Меню-троеточие – Настройки – Сочетания клавиш
4. Выберите неверное утверждение
  1. Chrome DevTools используется для тестирования и отладки кода
  2. Инструменты разработчика помогают находить ошибки
  3. DevTools можно открыть клавишей F12
  4. DevTools используется для автоматического написания кода
5. Что нельзя делать с сочетаниями клавиш?
  1. Добавлять
  2. Удалять
  3. Изменять
  4. Изменять порядок

Тест к теме «Работа с JavaScript и CSS»

1. Когда код выполнится без ошибок?
  1. 

```
const test;  
console.log(test);
```
  2. 

```
var test = 'test';  
console.log(tost);
```
  3. 

```
const test = 'test';  
console.log(test);
```
  4. 

```
var test;  
console.log(test);
```
2. Какие виды сообщений можно увидеть в консоли разработчика?
  1. Debug, Error, Message
  2. Fatal, Error, Log
  3. Message, Info, Log
  4. Error, Warning, Info
3. Что будет выведено в консоли?

```
var v1 = Symbol('value2');  
var v2 = Symbol('value2');  
var r = v1 == v2;  
console.log(r);
```

  1. false
  2. true
  3. null
  4. Будет ошибка

4. В каком случае используется тип данных BigInt?
1. `var bi = 5;`
  2. `var x = 5.05;`
  3. `let y = 5B;`
  4. `var bi = 5n;`
5. Что будет выведено в консоль?
- ```
const car = class MyCar {
  model = "Volvo";
}
var car1 = new MyCar();
console.log(car1.model);
```
1. Ничего
  2. 'MyCar' has already been declared
  3. Volvo
  4. MyCar is not defined
6. Когда код выполнится без ошибок?
1. `class M{constructor(n){this.n=n;}} console.log(k.n);`
  2. `class M{constructor(n){this.n=n;}} console.log(new m("n"));`
  3. `class M{constructor(nn){this.nn=nn;} var s = new M("test it!"); console.log(s.nn);`
  4. `class M{constructor(n){this.n=n;}} var k = new M("n"); console.log(k.n);`
7. Какого типа ошибок не существует в JavaScript?
1. `ExceptionError`
  2. `TypeError`
  3. `URIError`
  4. `ReferenceError`
8. Какой код вызовет ошибку `SyntaxError`?
1. `var opechatka = ['1', '2'] delete opechatka[0];`
  2. `var abrakadabra = ['1', '2']; delete arr[0];`
  3. `var arr = ['1', '2']; delete arr[0];`
  4. `var arr = ['1', '2']; delite arr[0];`
9. Что такое атрибут?
1. Устаревший элемент, который уже не используется
  2. Силь для экрана монитора
  3. То же самое, что тег
  4. То, что указывается внутри скобок тега
10. Какого атрибута не существует у тега `style`?
1. `type`
  2. `media`
  3. Все перечисленные
  4. `mime`
11. Что такое `font-size` в представленном CSS-файле?
- ```
p {
  font-size: 26px;
}
```

1. Селектор
  2. Правило
  3. Значение
  4. Свойство
12. Каким образом можно выполнить внешнее встраивание CSS?
1. `<style>...</style>`
  2. `<link href="stylesheet" rel="css/style.css">`
  3. Такого встраивания не существует
  4. `<link rel="stylesheet" href="css/style.css">`
13. Какой в итоге будет цвет у текста?
- ```

<html>
  <head>
    <style>
      html { color: yellow; }
      body { color: purple; }
      p { color: black; }
    </style>
  </head>
  <body>
    <div style="color: grey;"><span style="color: red;">Текст</span></div>
  </body>
</html>

```
1. Серый
  2. Чёрный
  3. Жёлтый
  4. Красный
14. Для чего придуманы псевдоклассы?
1. Чтобы применять к элементам стили, которые не входят в обычные CSS-стандарты
  2. Чтобы применять стили к элементам, находящимся на одном уровне вложенности
  3. Чтобы применять стили ко всем элементам на странице сразу
  4. Чтобы применять стили к элементам, находящимся в определённом состоянии или порядке
15. Что нельзя делать с сочетаниями клавиш?
1. Добавлять
  2. Удалять
  3. Изменять
  4. Изменять порядок

#### Тест к уроку «Основы безопасности веб-приложений»

1. Какая атака является отраженной (reflected)?
  1. Атака при которой внедренный код отправляется на сервер, сразу возвращается в исходном виде на клиента пользователя, где и выполняется
  2. Атака при которой внедренный код отправляется на сервер и там сохраняется. Внедренный код выполняется при обращении пользователя к странице, отображающей его
  3. Атака, которую удалось отразить сотрудникам кибербезопасности.
  4. Атака, которая негативно отразилась на пользователе или организации.
2. Цель внедрения HTML-кода на страницу?
  1. Ввести пользователя в заблуждение, собрать личные данные пользователя.
  2. Коррекция интерфейса веб-приложения
  3. Сбор или изменений информации на сервере
  4. Расширение функциональности веб-приложения



3. Цель внедрения SQL-кода на страницу?
  1. Отображение недостающей информации на веб-странице
  2. Сбор или изменение информации в базе данных
  3. Расширение функциональности веб-приложения
  4. Введение пользователя в заблуждение.
4. Какая атака является хранимой (stored)?
  1. Атака, которая проведена киберпреступниками не до конца и находится в состоянии хранения. И в необходимый момент может быть продолжена
  2. Атака, которую сохранили сотрудники кибербезопасности для рассказа внукам
  3. Атака при которой внедренный код отправляется на сервер и там сохраняется. Внедренный код выполняется при обращении пользователя к странице, отображающей вредоносный код.
  4. Атака, которой подвергся пользователь и хранит как воспоминание
5. Что такое OWASP Top 10?
  1. Список наиболее популярных уязвимостей, которые подвергаются атакам киберпреступников
  2. Список наиболее популярных ошибок в веб-приложениях
  3. Организация, занимающаяся вопросами безопасности веб-приложений
  4. Рейтинг наиболее успешных атак на веб-приложения

#### Тест к уроку «Управление cookie, сессиями»

1. Что такое сессионные cookie-файлы?
  1. Cookie-файлы, постоянно хранящиеся в браузера для поддержания активной сессии пользователя, вне зависимости его реальной работы. При открытии страницы браузера сессия пользователя продолжается
  2. Cookie-файлы, предназначенные для сопровождения сессии рекламной компании
  3. Cookie-файлы, действующие в рамках сессии работы пользователя в браузере. После закрытия браузера эти cookie-файлы удаляются
  4. Cookie-файлы, создающиеся при каждом запросе пользователя к системе, для постоянного обновления сессии пользователя на сервере
2. Какие флаги связаны с безопасностью cookie-файлов?
  1. Secure, HttpOnly, SameSite
  2. Secure, Priority, SameSite
  3. Priority, Domain, Path
  4. HttpOnly, Domain, Expires
3. За что отвечает флаг secure в cookie-файлах?
  1. Таким флагом помечаются защищенные сервером cookie-файлы. Только сервер их может изменить.
  2. Флаг отвечает за передачу cookie-файлов исключительно по защищенному протоколу HTTPS. Cookie-файлы с этим флагом запрещено передавать по протоколу HTTP.
  3. Таким флагом помещаются cookie-файлы, которые обеспечивают безопасность пользователя. Для удобства их нахождения и работы с ними.
  4. Флаг отвечает за дополнительную защиту cookie-файла на уровне браузера.
4. Где можно увидеть cookie-файлы пользователя?
  1. В DevTools браузера на вкладке Приложение в разделе Файлы cookie
  2. В DevTools браузера на вкладке Сеть в деталях запроса
  3. В DevTools браузера на вкладке Консоль, с помощью команды `document.cookie`

#### 4. Все ответы верны

#### Тест к уроку «SQL инъекции»

1. Что обычно является целью внедрения SQL-кода?
  1. Получение закрытых данных из базы данных на сервере
  2. Получение персональной информации пользователя из браузера
  3. Изменение кода приложения на сервере
  4. Нарушение функционирования веб-приложения
2. Каково предназначение оператора UNION в SQL?
  1. Добавляет новые записи в базу данных
  2. Удаляет записи из базы данных
  3. Оператор объединяет результаты выполнения запросов SELECT
  4. Изменяет права пользователя в базе данных
3. Что является возможностью внедрения SQL-кода?
  1. Желание злоумышленника внедрить SQL-код в веб-приложение
  2. Отсутствие проверки корректности пользовательского ввода
  3. Ошибки при разработке веб-приложения
  4. Некачественное выполнение тестирования веб-приложения
4. Каким самым простым способом можно проверить наличие возможности внедрения SQL-кода в поле?
  1. Изучить html-код описания поля
  2. Изучить http-запрос, в котором передается значение поля
  3. Изучить программный код, в котором обрабатывается значение поля
  4. Ввести в поле формы символ кавычки - < ' >

#### Тест к уроку «Тестирование XSS-уязвимости»

1. Какая цель проведения XSS атаки?
  1. Сбор данных о пользователе при помощи кода
  2. Выполнение от имени пользователя произвольного JavaScript-кода
  3. Получение защищенных данных от сервера с помощью внедрения кода
  4. Нанести вред пользователю и организации
2. Как выполняется хранимая XSS атака?
  1. Пользователь открывает ссылку содержащую JS-код, который выполняется в браузере пользователя.
  2. JavaScript-код располагается в веб-приложении, пользователь выполняет локальное действие и это запускает вредоносный код на выполнение.
  3. JavaScript отправляется на сервер и там сохраняется в хранилище среди данных, пользователь обращается за данными, получает в браузере данные и сохраненный JS-код, который выполняется в браузере пользователя.
3. Как выполняется отраженная XSS атака?
  1. JavaScript-код располагается в web-приложении, пользователь выполняет локальное действие и это запускает вредоносный код на выполнение.
  2. JavaScript отправляется на сервер и там сохраняется в хранилище среди данных, пользователь обращается за данными, получает в браузере данные и сохраненный JS-код, который выполняется в браузере пользователя.
  3. JavaScript-код отправляется на сервер, там он преобразуется (отражается) и отправляется пользователю для выполнения
  4. Успешно отраженная сотрудниками безопасности атака.
4. Где выполняется XSS атака?
  1. На сервере в веб-сервере
  2. На сервере в базе данных
  3. В браузере пользователя
  4. В операционной системе пользователя
5. Чем опасна XSS атака?

1. XSS атака ограничивается возможностями языка JavaScript и знаниями злоумышленника
2. Раскрытием персональных данных пользователя
3. Кражей ключей доступа у пользователя к системе
4. Обманом пользователя и подменой функциональности системы

Тест к теме «Тестирование безопасности веб-приложений»

1. Какая атака является отраженной (reflected)?
  1. Атака при которой внедренный код отправляется на сервер, сразу возвращается в исходном виде на клиента пользователя, где и выполняется
  2. Атака при которой внедренный код отправляется на сервер и там сохраняется. Внедренный код выполняется при обращении пользователя к странице, отображающей его
  3. Атака, которую удалось отразить сотрудникам кибербезопасности.
  4. Атака, которая негативно отразилась на пользователе или организации.
2. Цель внедрения HTML-кода на страницу?
  1. Ввести пользователя в заблуждение, собрать личные данные пользователя.
  2. Коррекция интерфейса веб-приложения
  3. Сбор или изменений информации на сервере
  4. Расширение функциональности веб-приложения
3. Что такое OWASP Top 10?
  1. Список наиболее популярных уязвимостей, которые подвергаются атакам киберпреступников
  2. Список наиболее популярных ошибок в веб-приложениях
  3. Организация, занимающаяся вопросами безопасности веб-приложений
  4. Рейтинг наиболее успешных атак на веб-приложения
4. Какие флаги связаны с безопасностью cookie-файлов?
  1. Secure, HttpOnly, SameSite
  2. Secure, Priority, SameSite
  3. Priority, Domain, Path
  4. HttpOnly, Domain, Expires
5. Где можно увидеть cookie-файлы пользователя?
  1. В DevTools браузера на вкладке Приложение в разделе Файлы cookie
  2. В DevTools браузера на вкладке Сеть в деталях запроса
  3. В DevTools браузера на вкладке Консоль, с помощью команды `document.cookie`
  4. Все ответы верны
6. Что обычно является целью внедрения SQL-кода?
  1. Получение закрытых данных из базы данных на сервере
  2. Получение персональной информации пользователя из браузера
  3. Изменение кода приложения на сервере
  4. Нарушение функционирования веб-приложения
7. Что является возможностью внедрения SQL-кода?
  1. Желание злоумышленника внедрить SQL-код в веб-приложение
  2. Отсутствие проверки корректности пользовательского ввода
  3. Ошибки при разработке веб-приложения
  4. Некачественное выполнение тестирования веб-приложения
8. Какая цель проведения XSS атаки?
  1. Сбор данных о пользователе при помощи кода
  2. Выполнение от имени пользователя произвольного JavaScript-кода
  3. Получение защищенных данных от сервера с помощью внедрения кода
  4. Нанести вред пользователю и организации
9. Как выполняется отраженная XSS атака?

1. JavaScript-код располагается в web-приложении, пользователь выполняет локальное действие и это запускает вредоносный код на выполнение.
  2. JavaScript отправляется на сервер и там сохраняется в хранилище среди данных, пользователь обращается за данными, получает в браузере данные и сохраненный JS-код, который выполняется в браузере пользователя.
  3. JavaScript-код отправляется на сервер, там он преобразуется (отражается) и отправляется пользователю для выполнения
  4. Успешно отраженная сотрудниками безопасности атака
10. Чем опасна XSS атака?
1. XSS атака ограничивается возможностями языка JavaScript и знаниями злоумышленника
  2. Раскрытием персональных данных пользователя
  3. Кражей ключей доступа у пользователя к системе
  4. Обманом пользователя и подменой функциональности системы

#### Тест к уроку «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?
  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 «Основные конструкции языка»

1. Как объявить константу в Python?
  1. var PI = 3.1415
  2. const PI = 3.1415
  3. final PI = 3.1415
  4. PI = 3.1415
2. 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

3. Какой будет равно z в результате следующего кода?

```
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
4. 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?  
(22178)
1. continue
  2. next
  3. skip
  4. ignore

Тест к уроку «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 1 и 2 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

#### Тест к уроку «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:

`['a', 'b', 'c']`

4. 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 «Работа со словарями и коллекциями»

2. 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.
3. Какая структура данных Python является неизменяемой?
  1. tuple
  2. list
  3. set
  4. dict
4. Какой метод используется для объединения двух списков в Python?
  1. merge()
  2. join()
  3. append()
  4. extend()
5. Какая структура данных Python использует для хранения пары ключ-значение?
  1. tuple
  2. list
  3. set
  4. dict

Тест к уроку «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 «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
4. 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
5. 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 теме «Основы языка Python»

1. Какой тип данных в Python используется для хранения текстовой информации?
  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. 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. 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 создания нового списка на основе существующего?
  1. copy()
  2. clone()
  3. update()
  4. duplicate()
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. 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
  12. 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
  13. Что делает оператор pass?
    1. Ничего
    2. Кидает исключение
    3. Пропускает итерацию в цикле
    4. Запускает отладчик
  14. 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 «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. Выберите верное утверждение
  1. Автоматизация всегда экономит ресурсы и время
  2. Красные тесты – это неплохо, если их немного
  3. Перезапуск тестов нужен, чтобы не тестировать вручную
  4. Автоматизация экономит время и ресурсы
3. Какого этапа нет в автоматизации тестирования?
  1. Составление сценариев
  2. Поддержка
  3. Маркетинг
  4. Планирование
4. Выберите верное утверждение
  1. Хорошие тесты могут быть документацией
  2. Нестабильные тесты – это плохо
  3. Автоматизированное тестирование – это автоматизация тестирования
  4. Все утверждения верные
5. Какого тестового фреймворка не существует?
  1. Behave
  2. Robot Framework
  3. Lettuce
  4. Putest

Тест к уроку «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 «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
```

    1. Скипает тесты
    2. Помечает тесты красными
    3. Создает фикстуру
    4. Помечает тестовую функцию фикстурой
  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 «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 «Разработка автотестов на API»

1. Какая самая популярная среда разработки под Python?
  1. PyCharm
  2. Notepad++
  3. Sublime text
  4. Visual Studio
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. Из каких частей состоит запрос?
  1. Из параметров
  2. Из тела запроса
  3. Из url'a и payload'a
  4. Заголовки, url, параметры и сам запрос

#### Тест у уроку «Разработка автотестов на Веб»

1. Из каких частей состоит веб-приложение?

1. Клиент и сервер
  2. Клиент и база данных
  3. База данных и сервер
  4. Только серверной
  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. Для чего нужен этап планирования в автоматизации?
    1. Чтобы определить функциональность
    2. Для описания сценариев
    3. Для определения приоритетности тестов
    4. Для проектирования и разработки тестов
  4. Какой фреймворк использован на практическом занятии?
    1. Robot Framework
    2. Selenium Grid
    3. Cypress
    4. Selenium
  5. Для чего нужны сценарии автоматизации?
    1. Для описания функциональности
    2. Для тестирования тестов
    3. Для поддержки тестов
    4. Для описания шагов теста
- Тест к теме «Автоматизация на Python»
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
  3. Что такое Assert?
    1. Проверка в тесте
    2. Проверяющая система
    3. Вид тестирования на Python
    4. Специальный атрибут в Pytest
  4. 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
  6. 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`
  7. Какой маркер нужно использовать для параметризации тестов?
    1. Params
    2. P
    3. Parameter
    4. Parametrize
  6. Какой файл нужно создать для хранения собственных маркеров?
    1. Pytest.py
    2. Python.py

3. Python.ini
4. Pytest.ini
7. Какая самая популярная среда разработки под Python?
  1. PyCharm
  2. Notepad++
  3. Sublime text
  4. Visual Studio
8. Какие сущности в Postman используются для обращения к API?
  1. Элементы
  2. Классы
  3. Запросы
  4. Тесты
9. Из каких частей состоит веб-приложение?
  5. Клиент и сервер
  6. Клиент и база данных
  7. База данных и сервер
  8. Только серверной
10. Какой фреймворк использован на практическом занятии?
  5. Robot Framework
  6. Selenium Grid
  7. Cypress
  8. Selenium

#### Тест к уроку «Установка Jenkins»

1. Плагины могут?
  1. Добавить новые функции в Jenkins
  2. Интегрировать Jenkins с другими инструментами
  3. Помочь автоматизировать различные части процесса сборки и развертывания
  4. Все варианты ответов
2. Что первое нужно сделать для настройки плагинов?
  1. Нажать кнопку "Управление плагинами" в левой боковой панели
  2. Войти на сервер Дженкинса и перейти на страницу "Управление Дженкинсом"
  3. Войти во вкладку "Установлено" для управления плагинами
  4. Скачать плагины из Docker
3. Pipeline скрипт прописывается в ?
  1. Dockerfile
  2. .yaml
  3. .Jnks
  4. Jenkinsfile
4. Скрипт пайплайна это?
  1. Последовательность шагов, в которых прописаны этапы и действия
  2. Команда для запуска pipeline
  3. Плагин для Jenkins.
  4. Расширение для Jenkinsfile
5. Заключительным действием при создании первой работы в Jenkins Будет?
  1. Создание скрипта pipeline
  2. Настройка URL-адреса для хранилища исходного кода
  3. Сохранение конфигурации и запуск тестовой сборки
  4. Занесение проекта в Jenkinsfile

#### Тест к уроку «Разработка bash-скриптов»

1. Скрипт Bash ...?

1. Содержит ряд команд, выполняемых интерпретатором оболочки Bash.
  2. Используется для итерации последовательности значений или элементов
  3. Является выражением, которое оценивает до true или false
  4. Является файлом с расширением .scr
2. Скрипты Bash используются для?
    1. Для описания последовательности действий
    2. Для написания кода сложных программ
    3. Автоматизации повторяющихся задач или для упрощения сложных команд
    4. Для более удобной работы в CLI
3. Скрипты Bash - это текстовые файлы с расширением?
    1. .scr
    2. .bash
    3. .yml
    4. .sh
4. В Bash-скриптах, циклы используются для?
    1. Для упрощенного повторения действий
    2. Для циклического использования программы
    3. Для описания всех команд в терминале
    4. Итерации набора команд многократно, либо определенное количество раз
5. Основными циклами будут?
    1. bash и while
    2. while и for
    3. for и bin
    4. bin и bash

#### Тест к уроку «Запуск тестов на Jenkins»

1. "Jenkins предоставляет мощную платформу для тестирования..." какой вид тестирования сюда не подходит?
  1. Unit тесты
  2. Интеграционные тесты
  3. Тестирование окружения
  4. Приемочное тестирование
2. Из чего состоит тестовая среда в Jenkins?
  1. Тестовых систем, включая JUnit, selenium, Cucumber
  2. Плагинов и инструментов для анализа результатов испытаний
  3. Пайплайнов и скриптов
  4. Одного или нескольких серверов, или виртуальных машин
3. Работа с тестами в Jenkins включает в себя?
  1. Создание и выполнение автоматизированных тестов
  2. Создания плагинов для Jenkins
  3. Настройку Jenkins
  4. Регрессивное тестирование
4. Какая тестовая система не поддерживается Jenkins?
  1. JUnit
  2. Selenium
  3. Cucumber
  4. Все поддерживаются
5. Что не является лучшей практикой организации тестирования в Jenkins?
  1. Отчеты о тестирование
  2. Прогон тест-кейсов через TMS системы
  3. Использование плагинов
  4. Использование распараллеливания

#### Тест к теме «CI/CD. Запуск тестов Jenkins»

1. Что первое нужно сделать для настройки плагинов?
  1. Нажать кнопку "Управление плагинами" в левой боковой панели
  2. Войти на сервер Дженкинса и перейти на страницу "Управление Дженкинсом"
  3. Войти во вкладку "Установлено" для управления плагинами
  4. Скачать плагины из Docker
2. Pipeline скрипт прописывается в ?
  1. Dockerfile
  2. .yaml
  3. .Jnks
  4. Jenkinsfile
3. Скрипт Bash ...?
  1. Содержит ряд команд, выполняемых интерпретатором оболочки Bash.
  2. Используется для итерации последовательности значений или элементов
  3. Является выражением, которое оценивает до true или false
  4. Является файлом с расширением .scr
4. Скрипты Bash - это текстовые файлы с расширением?
  1. .scr
  2. .bash
  3. .yaml
  4. .sh
5. Работа с тестами в Jenkins включает в себя?
  1. Создание и выполнение автоматизированных тестов
  2. Создания плагинов для Jenkins
  3. Настройку Jenkins
  4. Регрессивное тестирование
6. Что не является лучшей практикой организации тестирования в Jenkins?
  1. Отчеты о тестирование
  2. Прогон тест-кейсов через TMS системы
  3. Использование плагинов
  4. Использование распараллеливания

#### Тест к уроку «Основы Allure и общие принципы отчетности»

1. Для чего нужен Allure?
  1. Для создания удобных и простых отчетов
  2. Для отчетности по тестировщикам
  3. Для отчетов по пройденным тестам
  4. Для управления тестированием, приоритезации тест-кейсов
2. Зачем нужны системы управления тестированием?
  1. Для создания тестовых сценариев
  2. Для прогона тестов
  3. Для определения, какие тесты нужно запускать в первую очередь, какие позже
  4. Для хранения тестовых сценариев, для запуска тестов, для определения очередности выполнения сценариев
3. Кто такой автоматизатор?
  1. Тестировщик, который тестирует задачи
  2. Роль, которая пишет тесты
  3. Отдельный человек в команде, который пишет и поддерживает тесты
  4. Разработчик, который реализует функциональность приложения
4. Какого блока не может быть в отчете о тестировании?
  1. Для информации
  2. Протестировано
  3. Число пройденных тестов



4. Число непротестированных задач
5. Какие инструменты отчетности сравнивались в уроке?
  1. Allure TestOps, TestRail
  2. Allure, TestOps, Report
  3. Allure Report, TestIT, Pytest
  4. Allure Report, TestIT, TestRail

Тест к уроку «Возможности Allure и его установка»

1. Какие составляющие необходимы для подготовки окружения для работы с Allure?
  1. Pycharm, Java, Python
  2. Pytest, Java, C#
  3. Java, C#, JavaScript
  4. Python, Python IDE, Java
2. Какой командой можно проверить установку Python?
  1. python check
  2. python v
  3. python settings
  4. python --version
3. Что такое scoop?
  1. Функциональность теста в уроке
  2. Интерфейс powershell'a
  3. Утилита, при помощи которой можно установить allure
  4. Вкладка в allure-отчете
4. В какой вкладке github'a можно скачать zip-архив allure report'a?
  1. Allure framework
  2. Code
  3. Pull requests
  4. Releases
5. Какой командой можно создать пустой allure-отчет?
  1. Allure --createEmptyReport
  2. Allure empty
  3. Allure create
  4. Allure serve

Тест к уроку «Написание тестов и сборка HTML-отчета»

1. Что делается в этой строке?
 

```
driver.get("https://www.google.com/doodles")
```

  1. Переход по ссылке
  2. Получение драйвера
  3. Инициализация драйвера браузера
  4. Переход по дудлам
2. Что нужно сделать, чтобы исчезло красное подчеркивание в PyCharm?
  1. Скачать библиотеку по работе с PyCharm
  2. Удалить слово с красным подчеркиванием
  3. Навести мышкой на подчеркивание и кликнуть по нему
  4. Навести мышкой на подчеркнутое слово, установить / импортировать недостающий пакет
3. Что означает следующая строка кода?
 

```
langChooser = driver.find_element(By.ID, "lang-chooser")
```

  1. Драйвер записывается в переменную langChooser
  2. Находится элемент по идентификатору
  3. Находится элемент по идентификатору и записывается в переменную
  4. Ищется последний доступный элемент на странице
4. Какие зависимости нужны для связывания проекта с тестами и allure report?
  1. allure-pytest, allure-python-comons
  2. allure

3. allure-pytest, allure
4. allure-python-commons, allure-pytest
5. Какая команда используется для генерации allure-отчета?
  1. allure start
  2. allure start server
  3. allure serve
  4. allure serve

Тест к уроку «Конфигурация для сборки отчета»

1. Зачем нужна система непрерывной интеграции?
  1. Для минимизации ручных действий
  2. Для автоматизации приложений
  3. Чтобы сделать проект с автотестами
  4. Чтобы заменить инженеров
2. Какие примеры CI приведены на занятии?
  1. Jenkins, Teamcity, Gitlab CI
  2. Bamboo, Jenkins, Teamcity, Gitlab CI
  3. Jenkins, Gitlab CI, Circle CI, Bamboo
  4. Jenkins, Teamcity, Bamboo, Gitlab CI
3. Что такое Jenkins?
  1. Утилита
  2. Сайт
  3. Служба
  4. Скрипт
4. Какой порт по умолчанию у Jenkins?
  1. 80
  2. 100
  3. 88
  4. 8080
5. Что такое ShiningPanda?
  1. Сияющая панда
  2. Система непрерывной интеграции
  3. Система непрерывной доставки
  4. Плагин в Jenkins для работы с python

Тест к уроку «Работа с отчетами»

1. Какой вкладки не существует в allure-отчете?
  1. Environments
  2. Packages
  3. Overview
  4. Suites
1. Что может быть внутри Epic?
  1. Epic
  2. Suite
  3. Sub Suite
  4. Feature, story
2. В какой вкладке нет кнопки для выгрузки в csv?
  1. Suites
  2. Behaviors
  3. Packages
  4. Во всех есть
3. Для чего нужны trend-графики?
  1. Для деления тестов по статусам
  2. Для сбора метрик тестов
  3. Для красоты

4. Чтобы показать прогоны во временном интервале
4. Какие бывают маркеры в allure?
  1. Tag, system, normal
  2. Critical, normal, Blocker
  3. Step, testcase, body
  4. Title, description, severity

Тест к теме «Автоматизация тестирования при помощи Allure»

1. Зачем нужны системы управления тестированием?
  1. Для создания тестовых сценариев
  2. Для прогона тестов
  3. Для определения, какие тесты нужно запускать в первую очередь, какие позже
  4. Для хранения тестовых сценариев, для запуска тестов, для определения очередности выполнения сценариев
2. Какого блока не может быть в отчете о тестировании?
  1. Для информации
  2. Протестировано
  3. Число пройденных тестов
  4. Число протестированных задач
3. Какой командой можно проверить установку Python?
  1. python check
  2. python v
  3. python settings
  4. python --version
4. В какой вкладке github'a можно скачать zip-архив allure report'a?
  1. Allure framework
  2. Code
  3. Pull requests
  4. Releases
5. Что нужно сделать, чтобы исчезло красное подчеркивание в PyCharm?
  1. Скачать библиотеку по работе с PyCharm
  2. Удалить слово с красным подчеркиванием
  3. Навести мышкой на подчеркивание и кликнуть по нему
  4. Навести мышкой на подчеркнутое слово, установить / импортировать недостающий пакет
6. Какая команда используется для генерации allure-отчета?
  1. allure start
  2. allure start server
  3. allure serve
  4. allure serve
7. Какие примеры CI приведены на занятии?
  1. Jenkins, Teamcity, Gitlab CI
  2. Bamboo, Jenkins, Teamcity, Gitlab CI
  3. Jenkins, Gitlab CI, Circle CI, Bamboo
  4. Jenkins, Teamcity, Bamboo, Gitlab CI
8. Какой порт по умолчанию у Jenkins?
  1. 80
  2. 100
  3. 88
  4. 8080
9. В какой вкладке нет кнопки для выгрузки в csv?
  1. Suites
  2. Behaviors

3. Packages
  4. Во всех есть
10. Какие бывают маркеры в allure?
1. Tag, system, normal
  2. Critical, normal, Blocker
  3. Step, testcase, body
  4. Title, description, severity

Тест к уроку «Основы нагрузочного тестирования»

1. В чем цель нагрузочного тестирования?
  1. Узнать максимальную нагрузку на сервер
  2. В выявлении уязвимых мест
  3. В создании кейсов по нагрузке
  4. Для отчетности
2. Одной из основных целей нагрузочного тестирования для QA будет?
  1. Поддержка бизнес-задач
  2. Масштабируемость
  3. Гибкость
  4. Проверка ресурсоемкости
3. Какая последовательность будет правильной?
  1. Выполнение теста, Анализ результатов и отчетность, Тест-дизайн, Планирование
  2. Планирование, Выполнение теста, Тест-дизайн, Анализ результатов и отчетность
  3. Планирование, Выполнение теста, Анализ результатов и отчетность, Тест-дизайн
  4. Планирование, Тест-дизайн, Выполнение теста, Анализ результатов и отчетность
4. Для чего нужны метрики?
  1. Для измерения и оценки прибыльности проекта
  2. Для измерения и оценки производительности программного обеспечения
  3. Для оценки производительности программного обеспечения
  4. Для оптимизации программного обеспечения
5. Что не показывают метрики?
  1. Процент ошибок
  2. Пропускная способность
  3. Процент использования графического процессора
  4. Время загрузки страницы

Тест к уроку «Jmeter»

1. Что можно выполнить с помощью JMeter из нижеперечисленных?
  1. Смотреть логи
  2. Делать запросы в базу данных
  3. Записывать действия пользователей
  4. Создавать bash-скрипты
2. Какой протокол не поддерживает JMeter?
  1. HTTP
  2. RDP
  3. FTP
  4. JDBC
3. С каким фреймворком не совместим JMeter?
  1. Jenkins
  2. Selenium WebDriver
  3. Kafka
  4. Angular

4. Какой отчет можно назвать дефолтным(базовым) в JMeter?
  1. Отчет в форме иерархического дерева.
  2. CSV-отчеты
  3. Графический отчет
  4. Сводный отчет
5. Какого типа контролера нет в thread group?
  1. Loop
  2. If
  3. While
  4. Performance

Тест к теме «Нагрузочное тестирование»

1. В чем цель нагрузочного тестирования?
  1. Узнать максимальную нагрузку на сервер
  2. В выявлении уязвимых мест
  3. В создании кейсов по нагрузке
  4. Для отчетности
2. Какая последовательность будет правильной?
  1. Выполнение теста, Анализ результатов и отчетность, Тест-дизайн, Планирование
  2. Планирование, Выполнение теста, Тест-дизайн, Анализ результатов и отчетность
  3. Планирование, Выполнение теста, Анализ результатов и отчетность, Тест-дизайн
  4. Планирование, Тест-дизайн, Выполнение теста, Анализ результатов и отчетность
3. Что не показывают метрики?
  1. Процент ошибок
  2. Пропускная способность
  3. Процент использования графического процессора
  4. Время загрузки страницы
4. Какой протокол не поддерживает JMeter?
  1. HTTP
  2. RDP
  3. FTP
  4. JDBC
5. Какой протокол не поддерживает JMeter?
  1. HTTP
  2. RDP
  3. FTP
  4. JDBC

Тест к уроку «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 «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 gun`
  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 «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. .cnps

#### Тест к уроку «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 развертывание проекта в Docker?
  1. Установка Docker
  2. Проверка Docker
  3. Обновление Docker
  4. Настройка Docker
3. Команда “docker push”?
  1. Переместить образ в реестр контейнеров
  2. Скачает образ из реестра контейнеров
  3. Создаст пустой dockerfile
  4. Покажет информацию о запущенных контейнерах
4. После перемещение 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 «Docker для тестировщиков»

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. Чтобы развернуть приложение требующего прямого доступа к физической сети, мы воспользуемся сетью?
  1. Оверлейные сети
  2. Сети Macvlan
  3. Хост-сети
  4. Мостовые сети
5. Что не является сервисом в Docker Compose?
  1. Контейнер
  2. Порт
  3. Том
  4. Среда
6. Какое расширение используется для Compose файлов?
  1. .docker
  2. .compose
  3. .yaml
  4. .comps
7. Что не входит в план развертывание проекта в Docker?
  1. Установка Docker
  2. Проверка Docker
  3. Обновление Docker
  4. Настройка Docker
8. Команда “docker push”?
  1. Переместить образ в реестр контейнеров
  2. Скачает образ из реестра контейнеров
  3. Создаст пустой dockerfile
  4. Покажет информацию о запущенных контейнерах
9. После перемещение образа в репозиторий, следует шаг?
  1. Это заключительный шаг
  2. Развертывание контейнера
  3. Настройка Docker
  4. Создание локальной сети
10. 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

### Итоговое тестирование

1. Когда код выполнится без ошибок?
  1. var a1, a2 <= 3;  
if (a1 <= a2) {  
}  
}
  2. var a1, a2 = 3;  
if a1 <= a2 {



- ```

        console.log(a2);
    }
3. a1, a2 = 3;
    if (a1 <= a2){
        console.log(a2);
    }
4. var a1 = 2;
    var a2 = 3;
    if (a1 <= a2){
        console.log(a2);
    }

```
2. Как вызвать вид сообщения «Ошибка» в консоли браузера?
    1. `var error = "error"`  
`console.log(error)`
    2. `console.log(new Error)`
    3. `console.log("error")`
    4. `console.error("error")`
  3. В каком случае используется тип данных BigInt?
    1. `var bi = 5;`
    2. `var x = 5.05;`
    3. `let y = 5B;`
    4. `var bi = 5n;`
  4. Какого атрибута не существует у тега `style`?
    1. `type`
    2. `media`
    3. Все перечисленные
    4. `mime`
  5. Сайт состоит из сотни страниц. Какой способ подключения предпочтительнее использовать?
    1. Внешнее встраивание
    2. Внутреннее встраивание
    3. Встроенное встраивание
    4. Правило `@import`
  6. Где найти справку по сочетаниям клавиш?
    1. Правый верхний угол – кнопка-троеточие – Дополнительные настройки – Сочетания клавиш
    2. Нажать F1
    3. Правый верхний угол – меню-троеточие - Сочетания клавиш
    4. Меню-троеточие – Настройки – Сочетания клавиш
  7. Цель внедрения HTML-кода на страницу?
    1. Ввести пользователя в заблуждение, собрать личные данные пользователя
    2. Коррекция интерфейса web-приложения
    3. Сбор или изменений информации на сервере
    4. Расширение функциональности web-приложения
  8. Какие флаги связаны с безопасностью cookie-файлов?
    1. `Secure`, `HttpOnly`, `SameSite`
    2. `Secure`, `Priority`, `SameSite`
    3. `Priority`, `Domain`, `Path`
    4. `HttpOnly`, `Domain`, `Expires`
  9. Что является возможностью внедрения SQL-кода?
    1. Желание злоумышленника внедрить SQL-код в web-приложение
    2. Отсутствие проверки корректности пользовательского ввода
    3. Ошибки при разработке web-приложения
    4. Некачественное выполнение тестирования web-приложения

10. Как выполняется отраженная XSS атака?

1. JavaScript-код располагается в web-приложении, пользователь выполняет локальное действие и это запускает вредоносный код на выполнение.
2. JavaScript отправляется на сервер и там сохраняется в хранилище среди данных, пользователь обращается за данными, получает в браузере данные и сохраненный JS-код, который выполняется в браузере пользователя.
3. JavaScript-код отправляется на сервер, там он преобразуется (отражается) и отправляется пользователю для выполнения
4. Успешно отраженная сотрудниками безопасности атака

11. 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

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

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

13. Как объявить функцию в Python?

1. function myfunction():
2. myfunction = function():
3. def myfunction():
4. declare myfunction as function():

14. С помощью какого ключевого слова создаются классы в Python?

1. cls
2. class
3. klass
4. type

15. Какая функция предназначена для вызова исключения?

1. Exception
2. RaiseException
3. Raises
4. Throws

16. Для чего нужен маркер skip?

1. Чтобы разрешить запуск теста
2. Беспольный маркер
3. Для того, чтобы украсить тесты
4. Чтобы не запускать тест

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

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

18. На каком языке построен Jenkins?

1. Python

2. JavaScript
  3. Java
  4. C#
19. Работа с тестами в Jenkins включает в себя?
    1. Создание и выполнение автоматизированных тестов
    2. Создания плагинов для Jenkins
    3. Настройку Jenkins
    4. Регрессивное тестирование
  20. Какого блока не может быть в отчете о тестировании?
    1. Для информации
    2. Протестировано
    3. Число пройденных тестов
    4. Число протестированных задач
  21. Какие зависимости нужны для связывания проекта с тестами и allure report?
    1. allure-pytest, allure-python-comons
    2. allure
    3. allure-pytest, allure
    4. allure-python-commons, allure-pytest
  22. Что такое ShiningPanda?
    1. Сияющая панда
    2. Система непрерывной интеграции
    3. Система непрерывной доставки
    4. Плагин в Jenkins для работы с python
  23. Для чего нужны trend-графики?
    1. Для деления тестов по статусам
    2. Для сбора метрик тестов
    3. Для красоты
    4. Чтобы показать прогоны во временном интервале
  24. Одной из основных целей нагрузочного тестирования для QA будет?
    1. Поддержка бизнес-задач
    2. Масштабируемость
    3. Гибкость
    4. Проверка ресурсоемкости
  25. Какой протокол не поддерживает JMeter?
    1. HTTP
    2. RDP
    3. FTP
    4. JDBC
  26. 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
  27. 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
  28. 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
29. Что не является сервисом в Docker Compose?
1. Контейнер
  2. Порт
  3. Том
  4. Среда
30. 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