

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

Утверждаю
Директор АНО ДПО
«Учебный центр СКБ Контур»


Е.Ю. Тerasимова
1 сентября 2025 г.



**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
повышения квалификации**

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ПО

Москва, 2025 г.

Оглавление

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
УЧЕБНЫЙ ПЛАН	5
УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН.....	6
КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК.....	7
Рабочая программа учебной дисциплины «Основы языка Python»	8
Рабочая программа учебной дисциплины «Автоматизация на Python»	11
Рабочая программа учебной дисциплины «Автоматизация тестирования при помощи Allure».....	14
Рабочая программа учебной дисциплины «Работа с логами»	17
ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОГРАММЫ.....	19
Формы аттестации	19
Критерии оценки слушателей	20
Фонд оценочных средств	21
ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ	44
Требования к квалификации педагогических кадров	44
Требования к материально-техническим условиям	45
Требованиям к информационным и учебно-методическим условиям.....	46

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная профессиональная программа повышения квалификации «Автоматизация тестирования ПО» (далее — Программа) разработана специалистами Автономной некоммерческой организации дополнительного профессионального образования «Учебный центр СКБ Контур» (далее — АНО ДПО «Учебный центр СКБ Контур»).

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, формы контроля знаний и требования к его проведению, календарный учебный график и другие материалы, обеспечивающие качество подготовки слушателей.

Образовательная деятельность слушателей предусматривает следующие виды учебных занятий и учебных работ:

- 1) изучение лекционного (теоретического) материала с помощью видеопараграфа и/или текстового параграфа;
- 2) практическое занятие: разбор практических ситуаций, примеров, выполнение практических заданий, решение контрольных вопросов;
- 3) задания для самостоятельной подготовки: домашние задания, контрольные вопросы, тесты, задачи, анализ практических ситуаций, иные формы самостоятельной работы слушателей, с целью глубокого усвоения учебных материалов.

Нормативные документы, регламентирующие разработку программы:

1. Федеральный закон Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
2. Приказ Министерства науки и высшего образования Российской Федерации от 24.03.2025 № 266 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам».
3. Приказ Минобрнауки России от 09.12.2016 № 1547 «Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 «Информационные системы и программирование».
4. Приказ Минтруда России от 02.08.2021г. № 531н «Об утверждении профессионального стандарта «Специалист по тестированию в области информационных технологий».
5. Письмо Минобрнауки России от 22.04.2015 № ВК-1032/06 «О направлении методических рекомендаций» (вместе с «Методическими рекомендациями-разъяснениями по разработке дополнительных профессиональных программ на основе профессиональных стандартов»).
6. Постановление Правительства РФ от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».
7. Методические рекомендации по реализации дополнительных профессиональных программ с использованием дистанционных образовательных технологий, электронного обучения и в сетевой форме (Письмо Минобрнауки России от 21.04.2015 № ВК-1013/06).

8. Методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов, утв. Минобрнауки России 22.01.2015 № ДЛ-1/05вн.

Цель обучения: совершенствование компетенций слушателей, необходимых для профессиональной деятельности по проведению тестирования программного обеспечения (ПО).

Задачи:

- формирование знаний необходимых для проведения оценки качества разрабатываемого программного обеспечения;
- практическая подготовка к подготовке тестовых данных и выполнению тестовых процедур ПО.

Категории слушателей:

- лица, имеющие среднее профессиональное и (или) высшее образование;
- лица, получающие среднее профессиональное и (или) высшее образование.

Возраст слушателей: 18 лет и старше.

Форма и сроки обучения по программе

Форма обучения: заочная, с использованием дистанционных образовательных технологий и/или электронного обучения.

Срок обучения: 76/6 (ак. час., нед.).

Трудоемкость программы: 39 академических часов самостоятельного обучения, 37 академических часов – работа на образовательной онлайн-платформе.

Режим занятий: определяется календарным учебным графиком

Планируемые результаты обучения по программе

Слушатель должен обладать общепрофессиональными компетенциями, включающими в себя способности:

- выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;
- использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.

Слушатель должен обладать профессиональными компетенциями, включающими в себя:

- определение и описание тестовых случаев для выполнения процесса тестирования ПО, включая разработку автотестов (В/01.5);
- проведение тестирования ПО по разработанным тестовым случаям (В/02.5).

В результате освоения программы слушатели должны будут:

Знать:

- основы проведения тестирования ПО с использованием языка программирования Python;
- общие принципы автоматизации тестирования;
- основы автоматизации тестирования при помощи Allure;
- как работать с логами.

Уметь:

- использовать язык программирования Python для тестирования и автоматизации тестирования;
- выбирать различные техники тестирования ПО;
- создавать автотесты;
- работать с логами.

УЧЕБНЫЙ ПЛАН

№ п/п	Наименование разделов и дисциплин	Всего часов	В том числе		Форма контроля
			Самостоятельная работа	Работа на образовательной онлайн-платформе	
1	Основы языка Python	32	16	16	Зачет
2	Автоматизация на Python	20	11	9	Зачет
3	Автоматизация тестирования при помощи Allure	14	8	6	Зачет
4	Работа с логами	8	4	4	Зачет
5	ИТОГОВАЯ АТТЕСТАЦИЯ	2	—	2	Зачет
—	Всего:	76	39	37	—

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Наименование разделов, дисциплин	Всего часов	В том числе		Форма контроля
			Самостоятельная работа	Работа на образовательной онлайн-платформе	
1	Основы языка Python	32	16	16	Зачет
1.1	Переменные и типы данных	4	2	2	Тестирование
1.2	Условный оператор и отладка	4	2	2	Тестирование
1.3	Циклы	4	2	2	Тестирование
1.4	Строки, регулярные выражения, обработка ошибок	4	2	2	Тестирование
1.5	Коллекции	4	2	2	Тестирование
1.6	Функции	4	2	2	Тестирование
1.7	Итераторы, генераторы, анонимные функции	4	2	2	Тестирование
1.8	Файловая система	4	2	2	Тестирование
2	Автоматизация на Python	20	11	9	Зачет
2.1	Общие принципы автоматизации тестирования	3	1	2	Тестирование
2.2	Модульное тестирование	3	2	1	Тестирование
2.3	Тесты на PyTest	4	2	2	Тестирование
2.4	Автотесты на мобильные приложения	3	2	1	Тестирование
2.5	Разработка тестов на Веб	3	2	1	Тестирование
2.6	Паттерн Page Object Model	4	2	2	Тестирование
3	Автоматизация тестирования при помощи Allure	14	8	6	Зачет
3.1	Основы Allure и общие принципы отчетности	2	1	1	Тестирование
3.2	Возможности Allure и его установка	2	1	1	Тестирование

№ п/п	Наименование разделов, дисциплин	Всего часов	В том числе		Форма контроля
			Самостоятельная работа	Работа на образовательной онлайн-платформе	
3.3	Написание тестов и сборка HTML-отчета	3	2	1	Тестирование
3.4	Конфигурация для сборки отчета	3	2	1	Тестирование
3.5	Работа с отчетами	4	2	2	Тестирование
4	Работа с логами	8	4	4	Зачет
4.1	Для чего необходимо логирование. Уровни и типы логирования	4	2	2	Тестирование
4.2	Основы работы с Graylog и Kibana	4	2	2	Тестирование
5	ИТОГОВАЯ АТТЕСТАЦИЯ	2	—	2	Зачет
—	Всего	76	39	37	—

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

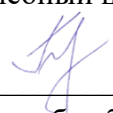
Календарный график обучения является примерным, составляется и утверждается для каждой группы.

Срок освоения программы — 6 недель. Начало обучения — по мере набора группы. Примерный режим занятий: 3 академических часа в день, 12–13 академических часов в неделю. Промежуточная и итоговые аттестации проводятся согласно графику.

№	Наименование модулей / недели	ФР	1	2	3	4	5	6
1	Основы языка Python	РП	12	4				
		СР		8	8			
2	Автоматизация на Python	РП			5	4		
		СР				11		
3	Автоматизация тестирования при помощи Allure	РП					6	
		СР					8	
4	Работа с логами	РП						4
		СР						4
5	Итоговая аттестация	РП						2

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

Утверждаю
Директор АНО ДПО
«Учебный центр СКБ Контур»


Е.Ю. Герасимова
1 сентября 2025 г.



**Рабочая программа учебной дисциплины
«Основы языка Python»**

образовательной программы дополнительного профессионального образования
повышения квалификации

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ПО

Москва, 2025 г.

Цель: освоение знаний по основам языка программирования Python для выполнения разных задач при тестировании программного обеспечения.

Требования к результатам освоения дисциплины

В результате обучения дисциплине слушатели должны:

Знать:

- Типы и структуры данных в языке Python.
- Основные конструкции языка.
- Операции с использованием строк, функции и методы строк.
- Особенности операций со списками.
- Использование арифметических функций, классов и объектов в языке Python.

Уметь:

- Читать код, созданный на языке программирования Python.
- Писать программы на языке Python используя различные типы данных.
- Выполнять работу со словарями и коллекциями.
- Проводить обработку исключений.
- Работать с отладчиком в Pycharm.
- Использовать полученные знания в практической работе.
- Владеть навыками профессионально и эффективно применять на практике приобретенные в процессе обучения знания и умения.

Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 32 часа (из них самостоятельная работа — 16 часов, работа на онлайн-платформе — 16 ак. часов).

Урок 1.1. Переменные и типы данных

- Знакомство, установка Pycharm.
- Понятие переменной, объекта и класса. Знакомство с базовыми функциями.
- Основные арифметические операции. Явные и неявные преобразования.
- Форматированный ввод и вывод.
- Самостоятельная работа.

Урок 1.2. Условный оператор и отладка

- Условный оператор if-elif-else
- Операторы сравнения
- Примеры использования условных операторов
- Работа с отладчиком в Pycharm
- Самостоятельная работа

Урок 1.3. Циклы

- Цикл while и бесконечные циклы.
- Примеры использования цикла while.
- Цикл for и интервалы.
- Особенности использования циклов for.
- Самостоятельная работа.

Урок 1.4. Строки, регулярные выражения, обработка ошибок

- Особенности класса <str>.
- Основные методы строк.
- Регулярные выражения и для чего они нужны.
- Операторы регулярных выражений.
- Обработка ошибок.
- Самостоятельная работа.

Урок 1.5. Коллекции

- Списки как универсальный контейнер.
- Методы списков.
- Особенности работы со списками.
- Генераторы списков.
- Кортежи и множества.
- Работа со словарями.
- Самостоятельная работа.

Урок 1.6. Функции

- Создание и вызов функций.
- Параметры и область видимости.
- Возврат значений и функций.
- Работа с рекурсивными функциями.
- Самостоятельная работа.

Урок 1.7. Итераторы, генераторы, анонимные функции

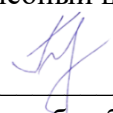
- Понятие итератора и итерируемого объекта.
- Понятие генератора.
- Генератор-функции.
- Анонимные функции.
- Работа с анонимными функциями.
- Самостоятельная работа.

Урок 1.8. Файловая система

- Работа с текстовыми и json файлами.
- Работа с табличными данными.
- Модуль CSV.
- Модули OS и SYS для взаимодействия с операционной системой.
- Создание своих библиотек.
- Самостоятельная работа.

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

Утверждаю
Директор АНО ДПО
«Учебный центр СКБ Контур»


Е.Ю. Герасимова
1 сентября 2025 г.



**Рабочая программа учебной дисциплины
«Автоматизация на Python»**

образовательной программы дополнительного профессионального образования
повышения квалификации

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ПО

Москва, 2025 г.

Цель: освоение знаний по автоматизации тестирования ПО с разными инструментами на Python.

Требования к результатам освоения дисциплины

В результате обучения дисциплине слушатели должны:

Знать:

- Этапы построения процесса автоматизации.
- Основные принципы автоматизации тестирования.
- Основы применения тестового фреймворка PyTest.
- Правила автоматизация тестирования при помощи Selenium.

Уметь:

- Применять тестовый фреймворк PyTest.
- Проводить тестирование с помощью Selenium.
- Использовать полученные знания в практической работе.
- Владеть навыками профессионально и эффективно применять на практике приобретенные в процессе обучения знания и умения.

Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 20 часов (из них самостоятельная работа — 11 ак. часов, работа на онлайн-платформе — 9 ак. часов).

Урок 2.1 Общие принципы автоматизации тестирования

- Принципы автоматизации тестирования.
- Инструменты автоматизации тестирования.
- Уровни автоматизации тестирования.
- Методологии разработки тестов (TDD, BDD).
- Самостоятельная работа.

Урок 2.2. Модульное тестирование

- Модульное тестирование.
- Написание модульных тестов.
- Самостоятельная работа.

Урок 2.3. Тесты на PyTest

- Обзор Pytest.
- Параметризация и маркировка тестов.
- Самостоятельная работа.

Урок 2.4. Автотесты на мобильные приложения

- Общие принципы автоматизации мобильного тестирования.
- Подготовка окружения и написание приложения.
- Написание тестов на мобильное приложение.
- Самостоятельная работа.

Урок 2.5. Разработка тестов на веб-приложения

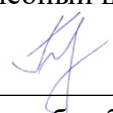
- Как устроены веб-приложения.
- Фреймворки автоматизации тестирования веб-приложений.
- Написание тестов на Selenium в веб-приложении.
- Самостоятельная работа.

Урок 2.6. Паттерн Page Object Model

- Виды паттернов проектирования в автоматизации тестирования.
- Реализация паттерн Page Object в Python.
- Самостоятельная работа.

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

Утверждаю
Директор АНО ДПО
«Учебный центр СКБ Контур»


Е.Ю. Герасимова
1 сентября 2025 г.



Рабочая программа учебной дисциплины
«Автоматизация тестирования при помощи Allure»
образовательной программы дополнительного профессионального образования
повышения квалификации

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ПО

Москва, 2025 г.

Цель: формирование знаний по основам автоматизации тестирования при помощи Allure, по созданию удобных и простых отчетов.

Требования к результатам освоения дисциплины

В результате обучения дисциплине слушатели должны:

Знать:

- Назначение Allure и его инструментов.
- Основы Allure и общие принципы отчетности.
- Возможности Allure.
- Назначение отчетности в автоматизации, виды тестирования.
- Особенности формирования HTML-отчета.
- Создание и настройка конфигурации в Jenkins.

Уметь:

- Устанавливать и готовить окружение для работы с Allure.
- Работать с отчетами в Allure.
- Связывать проект с тестами на Python.
- Настраивать конфигурацию в Jenkins на запуск тестов и формирование html-отчета с помощью allure report.
- Использовать полученные знания в практической работе.
- Владеть навыками профессионально и эффективно применять на практике приобретенные в процессе обучения знания и умения.

Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 14 часов (из них самостоятельная работа — 8 ак. часов, работа на онлайн-платформе — 6 ак. часов).

Урок 3.1. Основы Allure и общие принципы отчетности

- Общие определения, применимость Allure.
- Назначение отчетности в автоматизации.
- Инструменты отчетности на Python.
- Самостоятельная работа.

Урок 3.2. Возможности Allure и его установка

- Подготовка окружения.
- Установка Allure.
- Возможности Allure.
- Самостоятельная работа.

Урок 3.3. Написание тестов и сборка HTML-отчета

- Написание тестов на Python+Selenium.
- Формирование HTML-отчета.
- Самостоятельная работа.

Урок 3.4. Конфигурация для сборки отчета

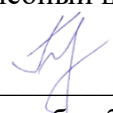
- Установка Jenkins.
- Настройка конфигурации Allure.
- Самостоятельная работа.

Урок 3.5. Работа с отчетами

- Работа с отчетами.
- Улучшение отчета.
- Самостоятельная работа.

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

Утверждаю
Директор АНО ДПО
«Учебный центр СКБ Контур»


Е.Ю. Герасимова
1 сентября 2025 г.



**Рабочая программа учебной дисциплины
«Работа с логами»**

образовательной программы дополнительного профессионального образования
повышения квалификации

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ПО

Москва, 2025 г.

Цель: формирование базовых компетенций по работе с системным логированием, включая навыки чтения, анализа логов и работы с централизованными системами логирования.

Требования к результатам освоения дисциплины

В результате обучения дисциплине слушатели должны:

Знать:

- Определение и назначение логирования.
- Основные уровни логирования.
- Типы логов и их назначение.
- Принципы работы с журналами событий.
- Архитектура и возможности систем Graylog и Kibana.
- Базовые конструкции регулярных выражений.
- Компоненты стека ELK и их взаимодействие.

Уметь:

- Проводить анализ логов различных уровней для выявления проблем.
- Настраивать базовые параметры логирования.
- Использовать консоль для просмотра и анализа логов.
- Работать с централизованными системами сбора логов.

Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 8 ак. часов (из них самостоятельное изучение материала — 4 ак. часов, работа на онлайн-платформе — 4 ак. часов).

Урок 4.1. Для чего необходимо логирование. Уровни и типы логирования

- Что такое логирование.
- Уровни и типы логирования.
- Чтение и интерпретация логов.
- Чтение логов через консоль.
- Самостоятельная работа.

Урок 4.2. Основы работы с Graylog и Kibana

- Централизованные системы сбора логов.
- Регулярные выражения (Regex).
- Обзор стека технологий ELK.
- Установка и практическая работа со стеком ELK.
- Что такое Graylog.
- Самостоятельная работа.

ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОГРАММЫ

Формы аттестации

Объектами оценивания выступают:

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

Текущий контроль знаний включает в себя контроль за учебной работой слушателей и проверку качества знаний и умений, которыми слушатели овладели на определенном этапе обучения. Текущий контроль знаний проводится посредством выполнения упражнений на практических занятиях и в иных формах, установленных преподавателем, а также в форме проверочного тестирования на протяжении всего обучения по программе.

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

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

Слушатель допускается к итоговой аттестации после изучения дисциплин Программы в объеме, предусмотренном для самостоятельных занятий и занятий на образовательной онлайн-платформе, и после сдачи тестов промежуточной аттестации.

Слушателям, освоившим образовательную программу повышения квалификации «Автоматизация тестирования ПО» и успешно прошедшим итоговую аттестацию, выдается **удостоверение о повышении квалификации** установленного образца с указанием названия программы, календарного периода обучения, длительности обучения в академических часах.

Для проведения промежуточной и итоговой аттестации Программы разработан фонд оценочных средств, который включает типовые задания, тесты и методы контроля, позволяющие оценить знания, умения и уровень приобретенных компетенций. Фонд оценочных средств соответствует целям и задачам программы подготовки специалиста, учебному плану и обеспечивает оценку качества общепрофессиональных и профессиональных компетенций, приобретаемых слушателями.

Критерии оценки слушателей

Предмет оценивания (компетенции)	Объект оценивания (навыки)	Показатель оценки (знания, умения)
— Определение и описание тестовых случаев для выполнения процесса тестирования ПО, включая разработку автотестов. — Проведение тестирования ПО по разработанным тестовым случаям	— Определение и описание тестовых случаев для выполнения процесса тестирования ПО, включая разработку автотестов. — Проведение тестирования ПО по разработанным тестовым случаям	Знания: — Типы и структуры данных в языке Python. — Основные конструкции языка Python. — Операции с использованием строк, функции и методы строк на Python. — Особенности операций со списками на Python. — Использовать арифметические функции, классы и объекты в языке Python. — Этапы построения процесса автоматизации — Основы применения тестового фреймворка PyTest. — Основы Allure и общие принципы отчетности. Возможности Allure. — Назначение отчетности в автоматизации, виды тестирования. — Особенности формирования HTML-отчета. — Основные команды Docker. — Определение и назначение логирования. — Архитектура и возможности систем Graylog и Kibana Умения: — Писать программы на языке Python используя различные типы данных. — Выполнять работу со словарями и коллекциями. — Проводить обработку исключений. — Применять тестовый фреймворк PyEest. — Устанавливать и готовить окружение для работы с Allure. — Работать с отчетами в Allure. — Связывать проект с тестами на Python. — Проводить анализ логов различных уровней для выявления проблем

Оценка качества освоения учебных модулей проводится в процессе промежуточной аттестации в форме зачета.

Оценка	Критерии оценки
Зачтено	Оценка « зачтено » выставляется слушателю, если он твердо знает материал курса, грамотно и по существу использует его, не допуская существенных неточностей в ответе на тестовые вопросы, правильно применяет теоретические положения при решении практических вопросов, владеет необходимыми навыками и приемами их выполнения. Не менее 80% правильных ответов при решении тестов.
Не зачтено	Оценка « не зачтено » выставляется слушателю, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями решает практические вопросы или не справляется с ними самостоятельно. Менее 80% правильных ответов при решении тестов.

Оценка качества освоения учебной программы проводится в процессе итоговой аттестации в форме тестирования.

Оценка (стандартная)	Требования к знаниям
Зачтено	Оценка « зачтено » выставляется слушателю, продемонстрировавшему твердое и всестороннее знание материала, умение применять полученные в рамках занятий практические навыки и умения. Достижения за период обучения и результаты текущей аттестации демонстрировали отличный уровень знаний и умений слушателя. Не менее 80% правильных ответов при решении тестов.
Не зачтено	Оценка « не зачтено » выставляется слушателю, который в недостаточной мере овладел теоретическим материалом по дисциплине, допустил ряд грубых ошибок при выполнении практических заданий, а также не выполнил требований, предъявляемых к промежуточной аттестации. Достижения за период обучения и результаты текущей аттестации демонстрировали неудовлетворительный уровень знаний и умений слушателя. Менее 80% правильных ответов при решении тестов.

Оценочные материалы

Тема 1. Основы языка Python

Урок 1.1. Переменные и типы данных

1. Данные какого типа возвращает функция `input()`?
 1. Целого
 2. Вещественного
 3. Строку
 4. Булев тип
2. С помощью какого оператора можно изменить порядок вычисления в Python?
 1. Скобки
 2. Умножение
 3. Возведение в степень
 4. Остаток от деления
3. Какой параметр надо изменить, чтобы функция `print()` вывела переданные ей аргументы в столбик?
 1. `end`
 2. `sep`
 3. `arg`
 4. `next`
4. Чтобы результат вычисления не сильно исказился, какой функцией лучше преобразовывать вещественное число в целое?
 1. `floor()`
 2. `ceil()`
 3. `int()`
 4. `round()`
5. Каким образом можно получить значение типа `int` после выполнения операции деления?
 1. Делить только целые числа
 2. Использовать целочисленное деление
 3. Взять модуль получившегося числа
 4. Такое невозможно

Урок 1.2 Условный оператор и отладка

Сколько условий можно проверить внутри одной условной конструкции?

1. 1
2. 3
3. Любое количество
4. 4

2. В чем важное отличие операторов = и ==?
 1. = присваивает значения, == сравнивает
 2. = — менее точный, == — более точный
 3. Отличий нет, это один и тот же оператор
 4. В количестве черточек
3. Что делает оператор pass?
 1. Пропускает выполнение текущей строки
 2. Пропускает выполнение следующей строки
 3. Выполняет строку самой последней
 4. Ничего, это просто пустая команда
4. Сколько вложенных блоков вы можете создать внутри другого вложенного блока?
 1. Сколько угодно
 2. 1
 3. 2
 4. 3
5. С помощью какого оператора можно проверить одновременное выполнение нескольких условий?
 1. and
 2. +
 3. or
 4. not

Урок 1.3. Циклы

1. В чем отличие оператора break от continue?
 1. break прерывает цикл, а continue переносит на следующую итерацию
 2. break можно использовать без цикла, а continue только в цикле
 3. break не позволяет вернуться в цикл после прерывания, а continue позволяет
 4. Отличий нет
2. Будет ли выполняться код, расположенный после цикла while?
 1. Не будет, так как while всегда работает бесконечно
 2. Будет только в случае если указать break в цикле
 3. Будет только если есть оператор continue в цикле
 4. Будет когда, while закончит свою работу
3. В чем принципиальное отличие цикла for от while?
 1. Количество повторений цикла for всегда известно перед его выполнением
 2. Количество повторений цикла for никогда не зависит от ввода пользователя
 3. В цикле for всегда обязательно указание условия выхода
 4. Оператор break в цикле for никогда не нужен

4. Какая последовательность чисел будет сгенерирована в `range(1,15, 5)` ?

1. 1, 5, 15
2. 1, 6, 11
3. 6, 11, 15
4. 10, 15, 20

5. Что можно использовать вместо функции `range()` в цикле `for`?

1. Любое число
2. Любую другую функцию
3. Ничего кроме `range()`
4. Любую последовательность или функцию ее возвращающую

Урок 1.4. Строки, регулярные выражения, обработка ошибок

1. Можно ли размещать длинный текст на несколько строк в программе?

1. True
2. False

2. Какова основная цель применения сырых строк?

1. Преобразование из числа в строку
2. Форматированный вывод содержимого строки на экран
3. Удаление пробелов в строке
4. Выводить и обрабатывать строку как она есть

3. Как можно изменить содержимое строки?

1. Только если создадим новую строку на основании старой
2. Обратиться к символу по номеру элемента и записать новый на его место
3. Использовать методы строк
4. Использовать цикл для перебора строки посимвольно

4. Какой метод модуля `re` позволит найти все точки в тексте?

1. `re.find('.')`
2. `re.findall('.')`
3. `re.sub('.')`
4. `re.split('.')`

5. Как мы можем заменить стандартный вывод сообщения об ошибке на свой?

1. Вызвать функцию `print()` в блоке `try`
2. Вызвать функцию `print()` в блоке `except`
3. Нельзя заменять стандартные сообщения об ошибках
4. Вызвать функцию `print()` в блоке `finally`

Урок 1.5. Коллекции

1. Какой номер индекса необходимо использовать первым, если выводить список по одному элементу с конца?
 1. -1
 2. 0
 3. 1
 4. Нельзя обращаться сразу к последнему элементу списка
2. В чем отличие между удалением элементов через `pop()` и `remove()`?
 1. `pop()` удаляет по индексу, а `remove()` — по значению
 2. `pop()` работает быстрее, чем `remove()`
 3. `pop()` возвращает удаленный элемент, а `remove()` — нет
 4. Все варианты верны
3. С помощью какой функции можно отсортировать элементы в множестве?
 1. Множество — неупорядоченная последовательность, а значит, сортировать его нельзя
 2. Метод `sort()`
 3. Нет готовой функции, необходимо писать собственную
 4. Воспользоваться готовыми алгоритмами (сортировка пузырьком, методом вставки и т.п.)
4. В чем преимущество метода `get()` перед оператором выбора `[]` при обращении к значению словаря?
 1. Преимущества нет, результат будет всегда одинаковый
 2. В том, что в случае отсутствия ключа ошибки не будет
 3. Работает быстрее
 4. Не изменяет словарь
5. Что произойдет, если по не существующему в словаре ключу запишем новое значение?
 1. Программа выдаст исключение
 2. Словарь останется неизменным
 3. Создастся ключ со значением `None`
 4. Создастся новая пара «ключ — значение»

Уроку 1.6. Функции

1. Как создать функцию с параметрами, которые не обязательно передавать при вызове?
 1. Сделать все параметры со значениями по умолчанию
 2. Присвоить значения параметрам внутри тела функции
 3. Так сделать нельзя
 4. Присвоить значения параметрам до вызова функции

2. Что будет если при вызове функции передать свое значение в параметр, для которого определено значение по умолчанию?
 1. Останется значение указанное по умолчанию
 2. Значение заменится на указанное
 3. Ошибка вызова функции
 4. Переданное значение будет следующим в списке параметров
3. Что делает оператор global?
 1. Указывает, что будет использована переменная, объявленная в глобальной области видимости
 2. Изменяет тип области видимости функции на глобальный
 3. Создает переменную в глобальной области видимости
 4. Создает глобальную функцию, доступную для любых программ
4. Что будет, если мы присвоим новое имя функции?
 1. Можно будет вызывать функцию как по новому, так и по старому имени
 2. Можно будет использовать только новое имя
 3. Исключение, функции после создания переименовывать нельзя
 4. Ничего, функция будет доступна только по старому имени
5. Что нужно сделать, первым делом, чтобы рекурсия не была бесконечной
 1. Выполнить рекурсивный вызов
 2. Продумать и реализовать способ завершения рекурсии
 3. Вернуть значение из рекурсивной функции
 4. Постараться избежать рекурсии и реализовать обычным способом

Урок 1.7. Итераторы, генераторы, анонимные функции

1. В чем отличие итератора от итерируемого объекта?
 1. Итератор — это часть итерируемого объекта
 2. Итератор нельзя использовать в цикле
 3. Элементы итератора можно получить только один раз
 4. Отличий нет
2. Какой объект будет меньше по размеру?
 1. Список четных чисел от 1 до 500 полученный с помощью генератора списков
 2. Генератор всех чисел от 1 до 10000
 3. Список нечетных чисел от 1 до 500, полученный с помощью цикла for
 4. Список всех чисел от 1 до 500, полученный с помощью цикла while?
3. В каких случаях будет выполняться код после оператора yield?
 1. При каждом вызове функции генератора начиная со второго раза
 2. Будет выполняться всегда
 3. Никогда не выполнится, так как yield возвращает значение из функции
 4. Выполнится только один раз при первом вызове

4. Чем `yield` отличается от `return`?

1. `yield` запоминает состояние функции, чтобы при следующем вызове продолжить выполнять код функции дальше
2. `return` возвращает конкретное значение, `yield` ничего не возвращает
3. `yield` может использоваться не только в функциях
4. Ничем. Это абсолютно одинаковые операторы

5. Найдите неверное утверждение.

1. Лямбда-функцию можно превратить в обычную присвоив ей имя
2. Лямбда-функции ничего не возвращают, так как в них отсутствует оператор `return`
3. Лямбда-функции можно возвращать как результат работы
4. Лямбда-функции могут принимать только один параметр

Урок 1.8. Файловая система

1. Как можно открыть файл одновременно на чтение и запись?

1. Так сделать нельзя
2. Использовать режим `'a'`
3. Использовать режим `'w+'` или `'r+'`
4. Использовать режим `'x'`

2. В чем отличие методов `json.dump()` и `json.dumps()`?

1. `dumps()` - для записи в файл `json`, `dump()` - для его чтения
2. `dumps()` конвертирует данные в строку, `dump()` - записывает как есть
3. Отличий нет
4. `dumps()` возвращает результат операции, `dump()` - не возвращает

3. Для чего указывать параметр `fieldnames` при записи CSV с использованием класса `DictWriter`?

1. Чтобы явно задать порядок следования столбцов
2. Чтобы можно было указать свои наименования столбцов
3. Чтобы не нарушать типы данных в столбцах
4. Чтобы преобразовать значения столбцов в строки

4. В чем разница работы `os.listdir(".")` и `os.walk(".")`?

1. `listdir` возвращает список, `walk` возвращает генератор
2. `listdir` позволяет узнать содержимое текущего каталога, `walk` — содержимое текущего каталога и всех вложенных
3. результат `listdir` можно использовать несколько раз. Результат `walk` только один
4. Все вышеперечисленное

5. С какого элемента списка мы начинаем чтение параметров, переданных в sys.args?
1. С первого
 2. Со второго
 3. С третьего
 4. С четвертого

Тест темы 1. Основы языка Python

1. Данные какого типа возвращает функция input()?
1. Целого
 2. Вещественного
 3. Строку
 4. Булев тип
2. Сколько условий можно проверить внутри одной условной конструкции?
1. 1
 2. 3
 3. Любое количество
 4. 4
3. В чем отличие оператора break от continue?
1. break прерывает цикл, а continue переносит на следующую итерацию
 2. break можно использовать без цикла, а continue только в цикле
 3. break не позволяет вернуться в цикл после прерывания, а continue позволяет
 4. Отличий нет
4. Как создать функцию с параметрами, которые не обязательно передавать при вызове?
1. Сделать все параметры со значениями по умолчанию
 2. Присвоить значения параметрам внутри тела функции
 3. Так сделать нельзя
 4. Присвоить значения параметрам до вызова функции
5. С какого элемента списка мы начинаем чтение параметров, переданных в sys.args?
1. С первого
 2. Со второго
 3. С третьего
 4. С четвертого

Тема 2. Автоматизация на Python

Урок 2.1. Общие принципы автоматизации тестирования

1. Что такое автоматизация тестирования?

1. Методика тестирования, при которой тестирование проводится специальными инструментами автоматизации
2. Методика тестирования, призванная уменьшать рутину
3. Вид тестирования, который нигде не используется
4. Вид тестирования, который занимает большую часть в тестировании

2. 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. Выберите верное утверждение

1. Сервисные тесты — это модульные тесты
2. TDD — это temp driven development
3. Интеграционные тесты ниже сквозных
4. Сервисные тесты — это интеграционные тесты

4. Какие существуют техники экстремального программирования?

1. Хорошее программирование, парное ревью
2. Нестабильное кодирование, экстремальное программирование
3. Парное кодирование, эмоциональное ревью
4. Парное программирование, экстремальное ревью

Какой методологии не существует?

1. KDT
2. CDT
3. DDT
4. TTT

Урок 2.2. Модульное тестирование

1. Что такое unittest?

1. Библиотека для модульного тестирования на Python
2. Внешний фреймворк автоматизации тестирования
3. Вид тестирования, который подлежит автоматизации
4. Модульное тестирование

2. Выберите неверное утверждение

1. Модульное тестирование нужно для упрощения отладки
2. Модульные тесты нужны для проведения регрессионного тестирования
3. Хорошие модульные тесты могут заменять документацию
4. Модульные тесты никогда не заменят документацию

3. Какие принципы модульного тестирования рассмотрены в уроке?

1. Написание тестов до релиза
2. Написание тестов после релиза
3. Тесты до и после реализации
4. Тесты до и после релиза

4. Что использовалось в уроке для первичного вывода чисел из реализации?

1. Дебаг, отладка и переписывание реализации
2. Вопросы аналитику
3. Тесты на unittest
4. Print и for in

5. Что такое модульное тестирование?

1. Тесты на всю систему в целом
2. Тесты между отдельными модулями
3. Самый верхний уровень в пирамиде тестирования
4. Самый первый уровень в пирамиде тестирования

Урок 2.3. Тесты на PyTest

1. Что такое Assert?

1. Проверка в тесте
2. Проверяющая система
3. Вид тестирования на Python
4. Специальный атрибут в Pytest

2. Что такое Pytest?

1. Pytest — это тестовая обвязка
2. Pytest — это среда для написания тестов
3. Pytest — это написание тестов на Python
4. Pytest — это тестовый фреймворк на Python

3. Что означает ZeroDivisionError?

1. Исключение общего характера
2. Ошибка деления на ноль
3. Множественное исключение
4. Тип исключения

4. 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
5. Что такое маркировка?
 1. Нанесение кода на товар
 2. Навешивание уникального кода на функцию
 3. Процесс навешивания маркеров на тесты
 4. Процесс написания теста

Урок 2.4. Автотесты на мобильные приложения

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

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

 1. Кликает по элементу
 2. Находит элемент по классу
 3. Находит элемент по ID и кликает по нему
 4. Код не сработает
4. Где создавался Android-эмулятор в уроке?
 1. В Pycharm
 2. В Appium
 3. В Android Studio
 4. В Device Manager
5. Какие параметры нужно указать Appium Server?
 1. Url сервера
 2. Имя устройства
 3. Версию Android
 4. Url сервера и настройки приложения

Урок 2.5. Разработка тестов на веб-приложения

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

Урок 2.6. Паттерн Page Object Model

1. Что такое паттерн проектирования?
 1. Способ организации кода тестов
 2. Построение тестовой инфраструктуры
 3. Способ написания теста
 4. Правило наименования теста
2. Какой паттерн позволяет задать несколько входных данных для одного теста?
 1. Сборка окружения
 2. ObjectMother
 3. Page Object
 4. Параметризация

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

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

4. Какой паттерн позволяет отделить одни тесты от других?

1. Разборка окружения
2. Параметризация
3. Логирование
4. Маркировка

5. Какого паттерна не существует?

1. Параметризация
2. Наименование тестов
3. Маркировка
4. Виртуализация

Тест темы 2. 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

2. Что такое модульное тестирование?

1. Тесты на всю систему в целом
2. Тесты между отдельными модулями
3. Самый верхний уровень в пирамиде тестирования
4. Самый первый уровень в пирамиде тестирования

3. Что такое Pytest?

1. Pytest — это тестовая обвязка
2. Pytest — это среда для написания тестов
3. Pytest — это написание тестов на Python
4. Pytest — это тестовый фреймворк на Python

4. Для чего нужен эмулятор?

1. Для эмуляции действий Desktop-приложений
2. Чтобы эмулировать поведение в браузере
3. Для запуска тестов
4. Чтобы запускать мобильные приложения

5. Что такое паттерн проектирования?

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

Тема 3. Автоматизация тестирования при помощи Allure

Урок 3.1. 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 протестированных 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 3.2 «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 scoor?
 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

Урок 3.3. 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

### **Урок 3.4. 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

## Урок 3.5. Работа с отчетами

### 1. Какой вкладки не существует в allure-отчете?

1. Environments (правильный ответ)
2. Packages
3. Overview
4. Suites

### 2. Что может быть внутри Epic?

1. Epic
2. Suite
3. Sub Suite
4. Feature, story

### 3. В какой вкладке нет кнопки для выгрузки в csv?

1. Suites
2. Behaviors
3. Packages
4. Во всех есть

### 4. 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

### 5. Какие бывают маркеры в allure?

1. Tag, system, normal
2. Critical, normal, Blocker
3. Step, testcase, body
4. Title, description, severity

### **Тест темы 3. Автоматизация тестирования при помощи Allure**

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. 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
3. Какая команда используется для генерации allure-отчета?
  1. allure start
  2. allure start server
  3. allure serve
  4. allure serve
4. Зачем нужна система непрерывной интеграции?
  1. Для минимизации ручных действий
  2. Для автоматизации приложений
  3. Чтобы сделать проект с автотестами
  4. Чтобы заменить инженеров
5. Какие бывают маркеры в allure?
  1. Tag, system, normal
  2. Critical, normal, Blocker
  3. Step, testcase, body
  4. Title, description, severity

### **Тема 4. Работа с логами**

#### **Урок 4.1. Для чего необходимо логирование. Уровни и типы логирования**

1. Согласны ли вы с тем, что логирование в приложениях не является обязательным?
  1. Да, согласны, это на усмотрение разработчика. Логи не так важны, как само приложение
  2. Нет, не согласны, это неотъемлемая часть при разработке ПО
2. Согласны ли вы с тем, что все уровни логирования во всех приложениях одинаковы?
  1. Да, согласны
  2. Нет, не согласны

3. Какой уровень логирования дает более подробную информацию?
  1. Error
  2. Trace
  3. Warning
4. Сохраняется ли информация о клиенте в логах веб-сервера Apache?
  1. Да, всегда сохраняется
  2. Нет, никогда не сохраняется
  3. Это опциональная информация, в зависимости от конфигурации может логироваться
5. Можно ли переключать уровень логирования?
  1. Нет, уровень всегда должен оставаться неизменным на всех окружениях
  2. Да, уровень логирования может переключаться по необходимости
  3. Уровень логирования регулярно переключается на тестовых стендах автоматически

### **Тест к уроку 4.2 «Основы работы с Graylog и Kibana»**

1. Для чего предназначены централизованные системы, такие как ELK или Greylog?
  1. Для централизованной отправки уведомлений
  2. Для сбора и анализа логов
  3. Для централизованного анализа
  4. Для сбора метрик
2. В каком формате хранятся данные в Elasticsearch?
  1. В строковом формате
  2. В формате JSON
  3. Могут храниться в любом формате
  4. В виде индексов
3. Из каких четырех основных компонентов состоит Greylog?
  1. Graylog Server, Graylog Web Interface, Elasticsearch, MongoDB
  2. Logstash, MongoDB, Kibana, Side Collector
  3. Beat, Logstash, Elasticsearch, Kibana
  4. Google Cloud Platform, Graylog Server, Web Interface, Side Collector
4. Для чего могут использоваться регулярные выражения? Выберите несколько вариантов.
  1. Для поиска подстрок в тексте
  2. Для поиска шаблонов
  3. Для поиска подтекста и его подмены
  4. Для удаления необходимого текста

5. Что такое GROK в файлах конфигурации Logstash?

1. Фильтр
2. Встроенный анализатор спецсимволов для парсинга логов
3. Конструктор, который собирает логи в один файл и перенаправляет в Elasticsearch
4. Хранилище

#### **Тест темы 4. Работа с логами**

1. Какой уровень логирования дает более подробную информацию?

1. Error
2. Trace
3. Warning

2. Сохраняется ли информация о клиенте в логах веб-сервера Apache?

1. Да, всегда сохраняется
2. Нет, никогда не сохраняется
3. Это опциональная информация, в зависимости от конфигурации может логироваться

3. Можно ли переключать уровень логирования?

1. Нет, уровень всегда должен оставаться неизменным на всех окружениях
2. Да, уровень логирования может переключаться по необходимости
3. Уровень логирования регулярно переключается на тестовых стендах автоматически

4. Для чего предназначены централизованные системы, такие как ELK или Greylog?

1. Для централизованной отправки уведомлений
2. Для сбора и анализа логов
3. Для централизованного анализа
4. Для сбора метрик

5. В каком формате хранятся данные в Elasticsearch?

1. В строковом формате
2. В формате JSON
3. Могут храниться в любом формате
4. В виде индексов

#### **Итоговое тестирование**

1. Данные какого типа возвращает функция input()?

1. Целого
2. Вещественного
3. Строку
4. Булев тип



2. Сколько условий можно проверить внутри одной условной конструкции?
  1. 1
  2. 3
  3. Любое количество
  4. 4
3. В чем отличие оператора break от continue?
  1. break прерывает цикл, а continue переносит на следующую итерацию
  2. break можно использовать без цикла, а continue только в цикле
  3. break не позволяет вернуться в цикл после прерывания, а continue позволяет
  4. Отличий нет
4. Как создать функцию с параметрами, которые не обязательно передавать при вызове?
  1. Сделать все параметры со значениями по умолчанию
  2. Присвоить значения параметрам внутри тела функции
  3. Так сделать нельзя
  4. Присвоить значения параметрам до вызова функции
5. С какого элемента списка мы начинаем чтение параметров, переданных в sys.args?
  1. С первого
  2. Со второго
  3. С третьего
  4. С четвертого
6. 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
7. Что такое модульное тестирование?
  1. Тесты на всю систему в целом
  2. Тесты между отдельными модулями
  3. Самый верхний уровень в пирамиде тестирования
  4. Самый первый уровень в пирамиде тестирования
8. Что такое Pytest?
  1. Pytest — это тестовая обвязка
  2. Pytest — это среда для написания тестов
  3. Pytest — это написание тестов на Python
  4. Pytest — это тестовый фреймворк на Python

9. Для чего нужен эмулятор?
1. Для эмуляции действий Desktop-приложений
  2. Чтобы эмулировать поведение в браузере
  3. Для запуска тестов
  4. Чтобы запускать мобильные приложения
10. Что такое паттерн проектирования?
1. Способ организации кода тестов
  2. Построение тестовой инфраструктуры
  3. Способ написания теста
  4. Правило наименования теста
11. Для чего нужен Allure?
1. Для создания удобных и простых отчетов
  2. Для отчетности по тестирующим
  3. Для отчетов по пройденным тестам
  4. Для управления тестированием, приоритезации тест-кейсов
12. 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
13. Какая команда используется для генерации allure-отчета?
1. allure start
  2. allure start server
  3. allure serve
  4. allure serve
14. Зачем нужна система непрерывной интеграции?
1. Для минимизации ручных действий
  2. Для автоматизации приложений
  3. Чтобы сделать проект с автотестами
  4. Чтобы заменить инженеров
15. Какие бывают маркеры в allure?
1. Tag, system, normal
  2. Critical, normal, Blocker
  3. Step, testcase, body
  4. Title, description, severity

16. Какой уровень логирования дает более подробную информацию?
1. Error
  2. Trace
  3. Warning
17. Сохраняется ли информация о клиенте в логах веб-сервера Apache?
1. Да, всегда сохраняется
  2. Нет, никогда не сохраняется
  3. Это опциональная информация, в зависимости от конфигурации может логироваться
18. Можно ли переключать уровень логирования?
1. Нет, уровень всегда должен оставаться неизменным на всех окружениях
  2. Да, уровень логирования может переключаться по необходимости
  3. Уровень логирования регулярно переключается на тестовых стендах автоматически
19. Для чего предназначены централизованные системы, такие как ELK или Greylog?
1. Для централизованной отправки уведомлений
  2. Для сбора и анализа логов
  3. Для централизованного анализа
  4. Для сбора метрик
20. В каком формате хранятся данные в Elasticsearch?
1. В строковом формате
  2. В формате JSON
  3. Могут храниться в любом формате
  4. В виде индексов

## **ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ**

Организация, осуществляющая образовательную деятельность, реализующая дополнительную профессиональную программу, укомплектована квалифицированными кадрами. Уровень квалификации работников организации, осуществляющей образовательную деятельность, реализующей дополнительную профессиональную программу, соответствует квалификационным характеристикам по соответствующей должности.

**Для реализации Программы задействован следующий кадровый потенциал:**

- **Преподаватели учебных дисциплин** — обеспечивают реализацию образовательного процесса.
- **Административный персонал** — обеспечивает условия для эффективной работы педагогического коллектива, осуществляет контроль и текущую организационную работу.
- **Информационно-технологический персонал** — обеспечивает функционирование информационной структуры (включая ремонт техники, оборудования, макетов иного технического обеспечения образовательного процесса, поддержание сайта КонтурШколы и т.п.).

### **Требования к квалификации педагогических кадров**

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

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

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

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

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

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

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

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

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

## Требования к материально-техническим условиям

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

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

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

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

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

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

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

1. Трансляция вебинара в режиме реального времени.
2. Хранение, систематизация записей вебинаров, с предоставлением участникам возможности просмотра записи онлайн.
3. Хранение, систематизация и доступ к скачиванию материалов учебных программ.
4. Напоминание участникам о предстоящем вебинаре за 1 час до начала мероприятия.
5. Использование защищенных соединений, передача и прием видео и звука по протоколам RTMP(S) или аналогичным.
6. Управление качеством и разрешением передаваемого/принимаемого видео вплоть до разрешения HD 720p на каждого участника мероприятия (адаптивный стриминг).
7. Обмен короткими текстовыми сообщениями (чат).
8. Осуществление записи мероприятий в формате, не требующем конвертации для проигрывания (mp4, AVI, WMA и т.д.).
9. Система регистрации на вебинар.
10. Техническое сопровождение проведения вебинара.
11. Отображение числа участников.
12. Техническая доступность услуги не менее 99,8% времени.

13. Устойчивость при проведении вебинара при одновременном подключении до 3000 участников.
14. Возможность участия пользователей на вебинарах в браузерах Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari с установленным плагином Adobe Flash Player.
15. Передача аудио- и видеoinформации на персональные компьютеры участников реализована при скорости интернет-соединения не менее 134 кбит/с.

Основные функции программы Контур.Школа:

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

### **Требованиям к информационным и учебно-методическим условиям**

Образовательная 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. Азиф М. Python для гиков. Пер. с англ. — СПб.: БХВ-Петербург, 2024. — 432 с.
2. Яворски М. Python. Лучшие практики и инструменты. 4-е изд. — СПб.: Питер, 2024. — 592 с.:
3. Марченко А. Л. Python: большая книга примеров. — М.: Издательство Московского университета, 2023. — 361, [1] с.

4. Старолетов С. М. Основы тестирования программного обеспечения: учебное пособие для СПО. — Лань; 2023. — 192 с.
5. Портянкин И. Программирование Cloud Native. Микросервисы, Docker и Kubernetes. — Издательские решения, 2022. — 45 с.
6. Куликов С. Тестирование программного обеспечения. Базовый курс. — ЕРАМ Systems, 2023. — 301 с.
7. Алпатов А. Н., Лобанов А. А., Лобанова Ю. С. Тестирование и отладка программного обеспечения: учебное пособие. — Киров. МЦИТО, 2021. — 58 с.
8. Назина О. Что такое тестирование. Курс молодого бойца. — БХВ, 2022. — 592 с.
9. Маурицио Аниче. Эффективное тестирование программного обеспечения. — ДМК-Пресс; 2023. — 370 с.

#### **Периодические издания**

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

#### **Интернет-ресурсы**

1. Автоматизация тестирования // Studfile.net, 2023. —  
URL: <https://studfile.net/preview/16543182/>
2. Автоматизированное тестирование // GitHub, 2023. —  
URL: <https://gist.github.com/codedokode/a455bde7d0748c0a351a>
3. Большой учебник по тестированию // Testengineer.ru, 2023. —  
URL: <https://testengineer.ru/bolshoj-uchebnik-po-testirovaniyu/#test-design>
4. Как тестируют мобильные приложения? // Qualitica, 2024. —  
URL: <https://qualitica.ru/blog/mobile-testing?ysclid=llxhvfkut7353766387>
5. Лучшие практики автоматизации // QA-Bible, 2024. —  
URL: [https://vladislavermeev.gitbook.io/qa\\_bible/avtomatizaciya-beta/luchshie-praktiki-avtomatizacii](https://vladislavermeev.gitbook.io/qa_bible/avtomatizaciya-beta/luchshie-praktiki-avtomatizacii)
6. Техническая документация по SQLServer от Microsoft // Microsoft, 2023. —  
URL: <https://docs.microsoft.com/ru-ru/sql/sql-server/?view=sql-server-ver15>
7. Упражнение по отработке навыков пользования SQL // Sql-ex.ru, 2023. —  
URL: <https://www.sql-ex.ru/>
8. Учебник «Основы Python с нуля» // Okpython.net, 2023. —  
URL:  
[https://okpython.net/python/python\\_uchebnik/python\\_uchebnik.html?yclid=15535983162980302847](https://okpython.net/python/python_uchebnik/python_uchebnik.html?yclid=15535983162980302847)