Контур Школа

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.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
повышения квалификации

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ НА PYTHON

Москва, 2025 г.

ОГЛАВЛЕНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная профессиональная программа повышения квалификации «Автоматизация тестирования на Python» (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 02.08.2021 № 531н «Об утверждении профессионального стандарта «Специалист по тестированию в области информационных технологий» (коды A, B).
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 21.04.2015г. № ВК-1013/06);
8. Методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов (утв. Минобрнауки России 22.01.2015г. № ДЛ-1/05вн).

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

Задачи:

- формирование знаний по основам языка программирования Python;
- формирование знаний по автоматизации тестирования ПО;
- формирование умений по автоматизации тестирования с применением языка программирования Python.

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: 78/6/1 (час., нед., мес.).

Трудоемкость программы: 40 академических часов (1 академический час равен 45 мин.) самостоятельного обучения, 38 академических часов работы на образовательной онлайн-платформе (включая аттестацию).

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:

- выполнение процесса тестирования ПО;
- определение и описание тестовых случаев для выполнения процесса тестирования ПО, включая разработку автотестов;
- проведение тестирования ПО по разработанным тестовым случаям.

В результате освоения программы слушатели должны будут:

знать:

- стандарты оформления программного кода для языка программирования Python;
- основы проведения тестирования ПО с использованием языка программирования Python;
- общие принципы автоматизации тестирования;
- особенности использования тестирования API: назначение, процесс, применимость;
- системы автоматизированного тестирования ПО.

Уметь:

- выбирать различные техники тестирования ПО;
- создавать проекты на Python для покрытия API-тестами;
- проводить тестирование безопасности веб-приложений;
- создавать автотесты для API и веб-приложений;
- использовать системы автоматизированного тестирования ПО.

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	52	24	28	Зачет
2	Автотесты на Python	24	16	8	Зачет
3	ИТОГОВАЯ АТТЕСТАЦИЯ	2	–	2	Зачет
–	**Всего:**	**78**	**40**	**38**	–

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**	**52**	**24**	**28**	**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	3	1	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
1.9	Декораторы и перегрузка	3	1	2	Тестирование
1.10	Введение в ООП	4	2	2	Тестирование
1.11	Инкапсуляция и полиморфизм	4	2	2	Тестирование
1.12	Наследование и абстрактные классы	3	1	2	Тестирование
1.13	Дескрипторы	3	1	2	Тестирование
1.14	Датаклассы и метаклассы	4	2	2	Тестирование
2	**Автотесты на Python**	**24**	**16**	**8**	**Зачет**
2.1	Общие принципы автоматизации тестирования	3	2	1	Тестирование
2.2	Уровни автоматизации тестирования	3	2	1	Тестирование
2.3	Модульное тестирование	3	2	1	Тестирование
2.4	Тесты на PyTest	3	2	1	Тестирование
2.5	Разработка тестов на API	3	2	1	Тестирование
2.6	Автотесты на мобильные приложения	3	2	1	Тестирование
2.7	Разработка тестов на Веб	3	2	1	Тестирование
2.8	Pattern Page Object Model	3	2	1	Тестирование
3	**ИТОГОВАЯ АТТЕСТАЦИЯ**	**2**	**–**	**2**	**Зачет**
—	**Всего:**	**78**	**40**	**38**	**–**

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Календарный график обучения является примерным.

Срок освоения программы — 6 недель. Примерный режим занятий: 3 - 5 академических часов в день, 12 - 16 часов в неделю. Промежуточная и итоговые аттестации проводятся согласно графику.

№	Темы / недели	ВР	1	2	3	4	5	6
1	Основы языка Python	**РП**	12	12	4			
		СР	4	4	8	8		
2	Автотесты на Python	**РП**				4	4	
		СР					8	8
3	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 НА PYTHON

Москва, 2025 г.

Цель: овладение знаниями по основам языка программирования Python.

Требования к результатам освоения дисциплины

В результате обучения дисциплине слушатели должны:

Знать:

- Типы и структуры данных в языке Python
- Особенности работы с разными типами данных, используемых в языке Python
- Основы использования условных операторов
- Циклы в языке Python
- Основные операции со словарем, с коллекцией
- Операции с функциями в Python
- Основы работы с декораторами, передача параметров
- Особенности работы с классами, дата и метаклассами
- Создание и применение дескрипторов, хранение данных в экземплярах дескриптора
- Назначение метаклассов и необходимость их использования

Уметь:

- Разрабатывать программы с использованием основных конструкций языка Python
- Использовать правильные имена переменных и констант, чтобы улучшить читаемость кода
- Работать с разными типами и структурами данных при разработке и тестировании программ на языке Python
- Работать с отладчиком в Pycharm
- Получать доступ к словарям, управлять ими
- Использовать коллекции данных при разработке и тестировании ПО
- Работать с функциями, в том числе с анонимными функциями
- Выполнять работу с декораторами
- Работать с классами, дата и метаклассами
- Использовать дискрипторы при создании классов

Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 52 академических часа (из них самостоятельная работа — 24 ак. часа, работа на образовательной онлайн-платформе — 28 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
- Практическое задание: написать программу, которая будет запрашивать у пользователя ввод трехзначного числа и выводить на экран заданную информацию

Урок 1.2. Условный оператор и отладка

- Условный оператор if-elif-else
- Операторы сравнения
- Примеры использования условных операторов
- Работа с отладчиком в Pycharm
- Практическое задание: написать программу, которая определяет, из какого количества разрядов состоит заданное число

Урок 1.3. Циклы

- Цикл while и бесконечные циклы
- Примеры использования цикла while
- Цикл for и интервалы
- Особенности использования циклов for
- Практическое задание: написать программу, которая просит ввести целое число и генерирует таблицу умножения на это число размером 20 строк на 10 столбцов

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
- Практическое задание: написать программу, которая будет заданную строку печатать задом наперед

Урок 1.5. Коллекции

- Списки как универсальный контейнер
- Методы списков
- Особенности работы со списками. Генераторы списков
- Кортежи и множества
- Работа со словарями
- Практическое задание: написать программу, которая определяет, есть ли в списке повторяющиеся элементы. Если да, то она должна вывести на экран все эти значения по одному разу

Урок 1.6. Функции

- Создание и вызов функций
- Параметры и область видимости
- Возврат значений и функций
- Работа с рекурсивными функциями
- Практическое задание: написать функцию, которая возводит число N в степень M

Урок 1.7. Итераторы, генераторы, анонимные функции

- Понятие итератора и итерируемого объекта
- Понятие генератора
- Генератор-функции
- Анонимные функции
- Работа с анонимными функциями
- Практическое задание: написать лямбда функцию, которая отсортирует список словарей по указанному ключу словаря

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
- Практическое задание: создать отдельный модуль из функций

Урок 1.9. Декораторы и перегрузка

- Понятие декоратора. Области применения
- Работа с декораторами. Передача параметров
- Понятие перегрузки. Перегрузка функций
- Практическая работа: создать декоратор, с помощью которого можно будет сварить кофе с одной любой добавкой и получить финальную стоимость напитка

Урок 1.10. Введение в ООП

- Понятие класса и объект класса
- Создание классов и объектов
- Функции классов. Методы объектов
- Конструктор класса. Инициализация объектов
- Работа с классами. Разбор примера
- Практическая работа: реализовать класс, который позволит хранить деньги на счете

Урок 1.11. Инкапсуляция и полиморфизм

- Приватные методы и свойства
- Свойства property, геттеры и сеттеры
- Статические методы @staticmethod и методы класса @classmethod
- Полиморфизм и перегрузка операторов
- Практическая работа: создать класс Fraction, который будет хранить в себе два целых числа - числитель и знаменатель дроби

Урок 1.12. Наследование и абстрактные классы

- Организация наследования. Родительские и дочерние классы
- Расширение функциональности. Классы super()
- Множественное наследование и миксины
- Абстрактные классы
- Практическая работа: разработать базовый класс Commodity, описывающий товар

Урок 1.13. Дескрипторы

- Понятие и использование дескрипторов
- Дескрипторы данных. Non-data дескрипторы
- Слабые ссылки weakref
- Проблемы хранения данных в дескрипторах
- Практическая работа: создать класс Car в котором реализуются свойства в виде объектов заданных дескрипторов

Урок 1.14. Датаклассы и метаклассы

- Способы создания классов
- Работа метаклассов
- Датаклассы и их назначения
- Создание и использование датаклассов
- Практическая работа: реализовать метакласс, который позволит создать только один экземпляр пользовательского класса

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
«Автотесты на Python»

образовательной программы дополнительного профессионального образования
повышения квалификации

АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ НА PYTHON

Москва, 2025 г.

Цель: применение знаний по автоматизации тестирования на языке программирования Python.

Требования к результатам освоения дисциплины

В результате обучения дисциплине слушатели должны:

Знать:

- Особенности тестирования веб-приложений и API
- Этапы построения процесса автоматизации
- Основные принципы автоматизации тестирования
- Основы применения тестового фреймворка Pytest
- Правила автоматизация тестирования при помощи Selenium
- Особенности составления автотестов для мобильных приложений
- Принципы автоматизации тестирования API
- Написание тестов с применением паттерна Page Object Model

Уметь:

- Применять тестовый фреймворк Pytest
- Проводить тестирование с помощью Selenium
- Тестировать API в Postman
- Проводить автоматизацию тестирования API на Python
- Использовать в кодовых базах тестирования веб-UI паттерн Page Object Model
- Писать тесты на мобильном и веб-приложении
- 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 24 академических часа (из них самостоятельная работа — 16 ак. часов, работа на образовательной онлайн-платформе — 8 ак. часов).

Урок 2.1. Общие принципы автоматизации тестирования

- Принципы автоматизации тестирования
- Инструменты автоматизации тестирования
- Практическая работа: провести сравнительный анализ инструментов автоматизации модульного тестирования PyTest и unittest

Урок 2.2. Уровни автоматизации тестирования

- Уровни автоматизации тестирования
- Методологии разработки тестов (TDD, BDD)
- Практическая работа: сравнить изученные формы пирамид тестирования и уровни автоматизации тестирования

Урок 2.3. Модульное тестирование

- Модульное тестирование
- Написание модульных тестов
- Практическая работа: выполнить реализацию вычитания, деления и умножения и покрыть ее тестами

Урок 2.4. Тесты на PyTest

- Обзор Pytest
- Параметризация и маркировка тестов
- Практическая работа: параметризовать значения типов данных: списки (lists) и кортежи (tuples)

Урок 2.5. Разработка тестов на API

- Общие принципы тестирования API
- Автоматизация тестирования API в Postman
- Автоматизация тестирования API на Python
- Практическая работа: создать PUT-запрос в Postman на обновление любого фильма

Урок 2.6. Автотесты на мобильные приложения

- Общие принципы автоматизации мобильного тестирования
- Подготовка окружения и написание приложения
- Написание тестов на мобильное приложение
- Практическая работа: реализовать тесты на вычитание чисел, деление и умножение

Урок 2.7. Разработка тестов на веб

- Как устроены веб-приложения
- Фреймворки автоматизации тестирования веб-приложений
- Написание тестов на Selenium в веб-приложении
- Практическая работа: написать тест на веб-приложение

Урок 2.8. Pattern Page Object Model

- Виды паттернов проектирования в автоматизации тестирования
- Реализация паттерна Page Object на Python
- Практическая работа: реализовать в тесте более умное ожидание вместо wait.sleep (чтобы тест дожидался появления элементов, которые могут не успевать загрузиться)

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.

Лицам, освоившим образовательную программу повышения квалификации по теме «Автоматизация тестирования на Python» 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, трудовые функции)	**Объект оценивания (навыки, трудовые действия)**	**Показатель оценки (знания, умения)**
– Выполнение процесса тестирования ПО (A/03.4); – определение и описание тестовых случаев для выполнения процесса тестирования ПО, включая разработку автотестов (B/01.5); – проведение тестирования ПО по разработанным тестовым случаям (B/02.5)	– Выполнение тестовых процедур на тестовых данных; – написание/настройка программ для автоматизированного тестирования ПО (при необходимости); – разработка автоматизированных тестов, в том числе для проверки информационной безопасности разрабатываемого ПО; – проведение автоматизированного тестирования ПО при необходимости; – оптимизация тестовых наборов	**Знания:** – Особенности мобильного тестирования. – Особенности тестирования веб-приложений и API. – Этапы построения процесса автоматизации. – Основные принципы автоматизации тестирования. – Основы применения тестового фреймворка Pytest. – Правила автоматизация тестирования при помощи Selenium. – Особенности составления автотестов для мобильных приложений на Python. – Типы и структуры данных в языке Python. Основные конструкции языка. – Операции с использованием строк, функции и методы строк. – Модификаторы доступа, свойства и дескрипторы. – Особенности обработки исключений. – Особенности работы с классами, дата и метаклассами. – Создание и применение дескрипторов, хранение данных в экземплярах дескриптора. **Умения:** – Писать код на языке Python. – Создавать автотесты для API, мобильных и веб-приложений. – Работать с PyTest и Selenium для создания автотестов. – Использовать функции для решения сложных задач по оптимизации кода и ускорению приложений. – Создавать иерархию классов с помощью наследования. – Использовать основные функции для работы с данными. – Строить сложную логику на классах с использованием метаклассов и дата-классов.

Предмет оценивания (компетенции, трудовые функции)	Объект оценивания (навыки, трудовые действия)	Показатель оценки (знания, умения)
		– Создавать проект на Python с тестами, спроектированными согласно паттерну Page Object Model. – Использовать полученные знания в практической работе. – Владеть навыками профессионально и эффективно применять на практике приобретенные в процессе обучения знания и умения.

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.

Фонд оценочных средств

Оценочные материалы

ТЕСТОВЫЕ ВОПРОСЫ

Тема 1

Тест к уроку 1.1

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 всего 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

Тест к уроку 1.2

1. 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. Сколько значений содержит булев тип?
    1. 1
    2. 2
    3. 3
    4. Бесконечное количество
4. 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
5. Сколько вложенных блоков вы можете создать внутри другого вложенного блока?
    1. Сколько угодно
    2. 1
    3. 2
    4. 3
6. С помощью какого оператора можно проверить одновременное выполнение нескольких условий?
    1. and
    2. +
    3. or
    4. not
7. Где логичнее всего поставить точку останова?
    1. В начале программы
    2. В конце программы
    3. В месте предполагаемой ошибки
    4. За несколько строк до места предполагаемой ошибки

Тест к уроку 1.3

1. В каком цикле мы можем реализовать выход по условию?
    1. В цикле for
    2. В цикле while
    3. В обоих циклах
    4. Выход из цикла по условию реализовать нельзя
2. 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
3. В какой ситуации нужны бесконечные циклы?
    1. Когда неизвестно условие выхода из цикла
    2. Когда неизвестно в какой момент выполнится условие выхода из цикла
    3. Когда известно точное количество итераций цикла
    4. Когда нам необходимо вызывать еще один цикл в текущем цикле
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. Найдите неверное выражение
    1. Счетчик цикла не является обязательным для циклов
    2. Цикл while может заменить цикл for во всех случаях
    3. Оператор continue заставляет python игнорировать команды стоящие после него
    4. Цикл for может заменить цикл while во всех случаях

Тест к уроку 1.4

1. Какие типы кавычек лучше использовать при создании строк в Python
    1. Одинарные
    2. Двойные
    3. Любые, главное, чтобы были одинаковыми
    4. Тройные
2. Можно ли размещать длинный текст на несколько строк в программе?
    1. True
    2. False

3. Как мы можем вывести на экран слово в кавычках?

1. Можем экранировать внутренние кавычки символом /
2. Любой вариант
3. Можем использовать другой тип кавычек внутри основной строки
4. Можем конкатенировать отдельно слово с кавычками

4. 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. Вывод и обработка строки как она есть

5. Какой метод позволит заменить все символы ‘o’ на ‘0’?

1. s.split(‘o’)
2. s.replace(‘o’,’0’)
3. s.join(‘0’)
4. s.strip(‘o’)

6. Как можно изменить содержимое строки?

1. Создать новую строку на основе 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

7. 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

Тест к уроку 1.5

1. Какие объекты можно хранить в списках?

1. Только объекты, принадлежащие одному классу
2. Только строки
3. Любые
4. Только числа

2. 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

3. 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

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 отсутствии ключа ошибки не будет
3. Работает быстрее
4. Не изменяет словарь

5. Какой результат вернет метод items() у словаря?

1. Общий список всех ключей и значений
2. Набор ключей
3. Набор значений
4. Набор кортежей «ключ — значение»

Тест к уроку 1.6

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. Сколько максимально рекурсивных вызовов может выполнить функция?

1. 995
2. 1000
3. 10
4. Будет выполнять, пока не заполнится стек

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

Тест к уроку 1.7

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. Сколько раз мы можем посчитать сумму чисел, получаемых с помощью одного генератора?

1. Бесконечное количество раз
2. Сможем считать пока не получим исключение Memory Error
3. Только два раза
4. Только один раз

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. Сколько выражений может выполнять анонимная функция?

1. Произвольное количество
2. Только одно
3. Ни одного, так как у функции нет имени
4. Только два

5. Как чаще всего используются лямбда-функции?

1. Как самостоятельные функции
2. Как элементы последовательностей
3. Как композиции функций
4. Как параметры для функций высшего порядка

Тест к уроку 1.8

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. При каком условии можно не закрывать файл после выполнения операций
    1. Если файл открывается с использованием оператора with
    2. Если файл был открыт внутри функции
    3. Файл нужно всегда закрывать самостоятельно
    4. Если файл открывался только для чтения
3. Какой символ может быть использован в качестве разделителя в CSV файлах?
    1. Только запятая
    2. Только запятая, но если она есть в данных, то точку
    3. Точка. Если есть точки в данных, то запятая
    4. Любой, но предпочтительно использовать запятую
4. С какого элемента списка мы начинаем чтение параметров, переданных в sys.args?
    1. С первого
    2. Со второго
    3. С третьего
    4. С четвертого
5. Каким образом мы можем изменить имя импортируемого объекта?
    1. Изменять нельзя
    2. Воспользоваться функцией os.path.basename()
    3. Воспользоваться функцией os. rename()
    4. Использовать оператор as

Тест к уроку 1.9

1. Что подразумевается под термином «замыкание»?
    1. Возможность вызова функций внутри других функций
    2. Возможность вернуть вызов функции
    3. Функция, которая запоминает значения из своей внешней области видимости, даже если эта область уже недоступна
    4. Функция, которая возвращает лямбду-функцию

2. В чем заключается основной смысл декораторов?

1. В возможности изменять поведение функции без изменения ее кода
2. В возможности вызвать несколько функций одновременно
3. В возможности изменить имя функции
4. В возможности измерять время работы функций

3. В какой последовательности применяются декораторы к функции, когда их несколько?

1. В обратном от порядка их указания в коде (сверху вниз)
2. В порядке следования (снизу вверх)
3. Все одновременно
4. Применяется только один декоратор, который объявлен ранее всех

4. В чем смысл перегрузки?

1. В возможности создать функцию, в которую можно передавать разное количество параметров
2. В возможности работы операторов и функций с разным количеством параметров и разными типами данных
3. В возможности функции по-разному реагировать на разные данные
4. В возможности функции возвращать несколько объектов

5. Каким образом можно проверить принадлежность объекта к определенному классу?

1. С помощью функции isinstance()
2. С помощью функции type()
3. Сравнить результат работы функции type с одним из существующих классов
4. Все вышеперечисленное

Тест к уроку 1.10

1. Что подразумевается под термином ООП?

1. Способность Python создавать разные пространства имен для объектов
2. Название всех классов, объектов, их методов и свойств
3. Способ написания кода при котором задачи решаются через классы и их объекты
4. Возможность разделять объекты на методы и свойства

2. Каким образом мы можем изменить содержимое словаря __dict__

1. Обратившись напрямую по ключу, в виде имени атрибута
2. Обратившись через класс по ключу, в виде имени атрибута
3. __dict__ является неизменяемым словарем
4. Через дот-нотацию, указав новое значение атрибуту

3. Что будет, если класс вызвать как функцию?

1. Вернется экземпляр класса
2. Исключение - классы не являются вызываемыми объектами
3. Вернется экземпляр класса, или None если в классе будет отсутствовать return
4. Ничего не произойдет

4. Сколько параметров по умолчанию принимает функция класса?

1. Один параметр - ссылку на объект у которого она будет вызвана
2. Два параметра - ссылку на объект и пользовательские параметры в виде *args
3. Ни одного параметра
4. *args и **kwargs

5. В чем основное назначение метода __init__()?

1. Создать экземпляр класса
2. Связать класс с объектами
3. Инициализировать пространство имен класса
4. Присвоить свойствам объекта указанные значения или реализовать логику проверки данных для объекта

Тест к уроку 1.11

1. Что подразумевается под термином “Инкапсуляция”?

1. Создание методов и свойств класса доступных только для чтения
2. Переопределение методов класса с целью возможности выполнения математических операций с объектами
3. Процесс скрытия методов и свойств класса
4. Возможность использовать объекты класса в качестве ключей словаря

2. В чем польза Name Mangling?

1. Позволяет создавать несколько имен одному свойству
2. Делает класс read-only
3. Делает свойства объектов доступными только для чтения
4. Защищает приватные свойства от случайной перезаписи другим значением

3. В чем разница между классом property и декоратором @property ?

1. Разницы нет. Декоратор - просто синтаксический сахар использования класса
2. Класс позволяет объявить и сеттеры и геттеры. Декоратор - только геттеры
3. Декоратор можно использовать только для одного свойства класса. Класс property - для любого количества
4. Декоратор позволяет создавать свойства только для чтения. Класс property - не позволяет так делать

4. В чем особенность методов класса?

1. Они не имеют доступа к атрибутам ни класса ни объектов
2. Они имеют доступ только к свойствам и функциям класса
3. Они могут создавать другие классы
4. С ними можно определить несколько конструкторов

5. Какие свойства могут принимать участие при формировании результата, возвращаемого из метода __hash__()?

1. Приватные свойства
2. Только свойства класса
3. только свойства объектов
4. Неизменяемые свойства

Тест к уроку 1.12

1. В чем заключается основной принцип наследования?

1. Можно создать методы и свойства, которые будут только у заранее определенных дочерних классов
2. Можно скрыть реализацию методов в родительском классе
3. Дочерние классы получают все функции и свойства родительского класса
4. Родительские классы получают доступ ко всем методам и свойствам дочерних классов

2. Представьте такую иерархию классов. Что выведет данный код?

```
class Food:
  pass

class Fruits(Food):
  pass

class Apple(Fruits):
  pass

a = Apple()
print(isinstance(a, Food))
```

1. False
2. True
3. TypeError
4. None

3. В чем преимущество функции super() перед указанием напрямую родительского класса?

1. Позволяет получить доступ к пространству имен родительского класса
2. Изолирует содержимое родительского класса от дочернего класса
3. Создает дополнительный объект для взаимодействия с родительским классом
4. Избавляет от зависимости от имени родительского класса

4. В каких случаях нужно применять миксины?

1. Когда у всех дочерних классов есть одинаковые методы
2. Когда есть общий функционал разных классов, не требующий сложной реализации
3. Когда создаем много дочерних классов
4. Когда применяем множественное наследование

5. Чем абстрактный класс отличается от обычного?

1. Нельзя создавать объекты абстрактного класса
2. Методы абстрактных классов не имеют конкретной реализации
3. Абстрактные методы обязательно должны быть определены во всех дочерних классах без исключения
4. Все вышеперечисленное

Тест к уроку 1.13

1. Когда нужно использовать дескрипторы?

1. Когда количество классов превышает три
2. Когда у класса много разных свойств
3. Когда у класса много свойств с одинаковым поведением
4. Когда у класса много методов

2. В чем отличие data-дескрипторов от non-data дескрипторов?

1. Non-data дескрипторы только для чтения, data-дескрипторы — для чтения и записи
2. Non-data дескрипторы поддерживают только метод __get__, data-дескрипторы — методы __get__ и __set__
3. Non-data дескрипторы не поддерживают изменение значений пользователем, data-дескрипторы поддерживают
4. Все вышеперечисленное

3. Где должны храниться значения свойств объекта при использовании дескрипторов?

1. В экземплярах пользовательского класса
2. В экземплярах дескриптора
3. В пользовательском классе
4. В классе дескриптора

4. Что происходит с объектом — слабой ссылкой, когда удаляется объект, на который она указывала?

1. Слабая ссылка полностью удаляется
2. Слабая ссылка изменяет состояние на dead
3. Слабая ссылка продолжает хранить ссылку на удаленный объект
4. Вместо слабой ссылки начинает возвращаться значение None

5. Какое основное назначение метода __set_name__?

1. Для обеспечения обращения к свойствам объекта всегда через методы дескриптора
2. Одинаковая обработка входных данных при создании разных свойств разных классов
3. Для сохранения данных всегда только в локальном словаре __dict__
4. Для создания слабых ссылок на свойства объектов в дескрипторе

Тест к уроку 1.14

1. Что на самом деле представляет собой type()?

1. Отдельная функция, выводящая тип данных объекта
2. Объект метакласса
3. Класс, объектами которого являются пользовательские классы
4. Метод класса, возвращающий родительский класс

2. Как можно создать свойство для метакласса, доступное для всех дочерних классов?

1. self.property = 50
2. cls.property = 50
3. property = 50
4. dict[‘property’] = 50

3. В чем заключается основное назначение классов данных?

1. В простой реализации хранения данных объектов
2. В реализации дополнительных обработок данных
3. В простой реализации хранения методов класса
4. В раздельном использовании методов и свойств классов

4. Почему нельзя создавать неинициализированные поля в мутабельных классах данных?

1. Потому что в классах данных все поля должны быть инициализированы
2. Потому что неинициализированные поля подразумевают изменение значения свойства, которое запрещено
3. Потому что нарушается порядок объявления параметров
4. Потому что вычисленное значение будет некорректным

5. Как можно изменять правила обработки отдельных полей класса данных?

1. Если наследоваться от метакласса
2. Если создать поле, являющееся объектом другого класса
3. Изменять поведение отдельных полей нельзя
4. С помощью параметров функции field()

Тест к теме 1. Основы языка Python

1. С помощью какого оператора можно изменить порядок вычисления в Python?
    1. Скобки
    2. Умножение
    3. Возведение в степень
    4. Остаток от деления

2. Чтобы результат вычисления не сильно исказился, какой функцией лучше всего преобразовывать вещественное число в целое?
    1. floor()
    2. ceil()
    3. int()
    4. round()

3. Сколько значений содержит булев тип?
    1. 1
    2. 2
    3. 3
    4. Бесконечное количество

4. Сколько вложенных блоков вы можете создать внутри другого вложенного блока?
    1. Сколько угодно
    2. 1
    3. 2
    4. 3

5. 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

6. Найдите неверное выражение
    1. Счетчик цикла не является обязательным для циклов
    2. Цикл while может заменить цикл for во всех случаях
    3. Оператор continue заставляет python игнорировать команды стоящие после него
    4. Цикл for может заменить цикл while во всех случаях

7. Можно ли размещать длинный текст на несколько строк в программе?
    1. True
    2. False

8. Как можно изменить содержимое строки?
    1. Создать новую строку на основе 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
9. Какие объекты можно хранить в списках?
    1. Только объекты, принадлежащие одному классу
    2. Только строки
    3. Любые
    4. Только числа
10. 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
11. 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
12. 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
13. 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’
14. С какого элемента списка мы начинаем чтение параметров, переданных в sys.args?
    1. С первого
    2. Со второго
    3. С третьего
    4. С четвертого

15. В чем заключается основной смысл декораторов?
    1. В возможности изменять поведение функции без изменения ее кода
    2. В возможности вызвать несколько функций одновременно
    3. В возможности изменить имя функции
    4. В возможности измерять время работы функций
16. Что будет, если класс вызвать как функцию?
    1. Вернется экземпляр класса
    2. Исключение - классы не являются вызываемыми объектами
    3. Вернется экземпляр класса, или None если в классе будет отсутствовать return
    4. Ничего не произойдет
17. В чем польза Name Mangling?
    1. Позволяет создавать несколько имен одному свойству
    2. Делает класс read-only
    3. Делает свойства объектов доступными только для чтения
    4. Защищает приватные свойства от случайной перезаписи другим значением
18. В каких случаях нужно применять миксины?
    1. Когда у всех дочерних классов есть одинаковые методы
    2. Когда есть общий функционал разных классов, не требующий сложной реализации
    3. Когда создаем много дочерних классов
    4. Когда применяем множественное наследование
19. Какое основное назначение метода __set_name__?
    1. Для обеспечения обращения к свойствам объекта всегда через методы дескриптора
    2. Одинаковая обработка входных данных при создании разных свойств разных классов
    3. Для сохранения данных всегда только в локальном словаре __dict__
    4. Для создания слабых ссылок на свойства объектов в дескрипторе
20. Как можно создать свойство для метакласса, доступное для всех дочерних классов?
    1. self.property = 50
    2. cls.property = 50
    3. property = 50
    4. dict[‘property’] = 50

Тема 2

Тест к уроку 2.1

1. Что такое автоматизация тестирования?
    1. Методика тестирования, при которой тестирование проводится специальными инструментами автоматизации
    2. Методика тестирования для уменьшения рутины
    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. Составления сценариев
    2. Поддержки
    3. Маркетинга
    4. Планирования
4. Выберите верное утверждение.
    1. Хорошие тесты могут быть документацией
    2. Нестабильные тесты — это плохо
    3. Автоматизированное тестирование — это автоматизация тестирования
    4. Все утверждения верны
5. Какого тестового фреймворка не существует?
    1. Behave
    2. Robot Framework
    3. Lettuce
    4. Putest

Тест к уроку 2.2

1. Что такое «уровень автоматизации тестирования»?
    1. Слой пирамиды тестирования
    2. Слой автоматизированного тестирования
    3. Вид тестирования, который подлежит автоматизации
    4. Часть ручного тестирования

2. 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
3. Какой формы пирамиды тестирования не существует?
    1. Формы «Песочные часы»
    2. Формы «Сота»
    3. Формы «Рыбий глаз»
    4. Формы «Пирамида»
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
5. Какой методологии не существует?
    1. KDT
    2. CDT
    3. DDT
    4. TTT

Тест к уроку 2.3

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

Тест к уроку 2.4

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. Какая команда используется для проверки версии pytest?
    1. pytest -version
    2. pytest version
    3. pytest --version
    4. pytest
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

6. 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
7. Какой маркер нужно использовать для параметризации тестов?
    1. Params
    2. P
    3. Parameter
    4. Parametrize
8. Что такое маркировка?
    1. Нанесение кода на товар
    2. Навешивание уникального кода на функцию
    3. Процесс навешивания маркеров на тесты
    4. Процесс написания теста
9. Для чего нужен маркер skip?
    1. Чтобы разрешить запуск теста
    2. Бесполезный маркер
    3. Чтобы украсить тесты
    4. Чтобы не запускать тест
10. Какой файл нужно создать для хранения собственных маркеров?
    1. Pytest.py
    2. Python.py
    3. Python.ini
    4. Pytest.ini

Тест к уроку 2.5

1. Какая цель у API?
    1. Предоставление функциональности пользователям, либо программам
    2. Описание интерфейса приложения
    3. Реализация функциональности
    4. Тестирование программного интерфейса
2. Какого вида запросов не существует?
    1. POST
    2. GET
    3. PATCH
    4. POSTING

3. Какие сущности в Postman используются для обращения к API?
    1. Элементы
    2. Классы
    3. Запросы
    4. Тесты
4. Для чего нужен токен?
    1. Токен не требуется
    2. Для проверки на уязвимости API
    3. Для обеспечения безопасности тестов на API
    4. Для разграничения доступа пользователей к API
5. Какой маркер используется для параметризации тестов?
    1. Mark
    2. Parameter
    3. Params
    4. Parametrize

Тест к уроку 2.6

1. При помощи какого инструмента искались элементы в уроке?
    1. Appium Inspector
    2. Appium Server GUI
    3. Appium
    4. Inspector
2. Для чего нужен эмулятор?
    1. Для эмуляции действий Desktop-приложений
    2. Чтобы эмулировать поведение в браузере
    3. Для запуска тестов
    4. Чтобы запускать мобильные приложения
3. Что делает этот код?
    driver.find_element(AppiumBy.ID, 'com.google.android.calculator:id/op_add').click()
    1. Кликает по элементу
    2. Находит элемент по классу
    3. Находит элемент по ID и кликает по нему
    4. Код не 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

Тест к уроку 2.7

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 уроку 2.8

1. Что такое паттерн проектирования?
    1. Способ организации кода тестов
    2. Построение тестовой инфраструктуры
    3. Способ написания теста
    4. Правило наименования теста
2. Какой паттерн позволяет задать несколько входных данных для одного теста?
    1. Сборка окружения
    2. ObjectMother
    3. Page Object
    4. Параметризация
3. Согласно какому паттерну тест должен быть из трех блоков?
    1. Page Object
    2. Параметризация
    3. AAA
    4. AAB
4. Какой паттерн позволяет отделить одни тесты от других?
    1. Разборка окружения
    2. Параметризация
    3. Логирование
    4. Маркировка
5. Какого паттерна не существует?
    1. Параметризация
    2. Наименование тестов
    3. Маркировка
    4. Виртуализация

Тест к теме 2. Автотесты на Python

1. Что такое автоматизация тестирования?
    1. Методика тестирования, при которой тестирование проводится специальными инструментами автоматизации
    2. Методика тестирования для уменьшения рутины
    3. Вид тестирования, который нигде не используется
    4. Вид тестирования, который занимает большую часть в тестировании

2. Выберите верное утверждение.
    1. Хорошие тесты могут быть документацией
    2. Нестабильные тесты — это плохо
    3. Автоматизированное тестирование — это автоматизация тестирования
    4. Все утверждения верны
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
5. Что такое unittest?
    1. Библиотека для модульного тестирования на Python
    2. Внешний фреймворк автоматизации тестирования
    3. Вид тестирования, который подлежит автоматизации
    4. Модульное тестирование
6. 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
7. Что такое Assert?
    1. Проверка в тесте
    2. Проверяющая система
    3. Вид тестирования на Python
    4. Специальный атрибут в Pytest
8. Для чего нужен маркер skip?
    1. Чтобы разрешить запуск теста
    2. Бесполезный маркер
    3. Чтобы украсить тесты
    4. Чтобы не запускать тест

9. Какого вида запросов не существует?
    1. POST
    2. GET
    3. PATCH
    4. POSTING
10. 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
11. 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
12. Какие параметры нужно указать Appium Server?
    1. Url сервера
    2. Имя устройства
    3. Версию Android
    4. Url сервера и настройки приложения
13. Из каких частей состоит веб-приложение?
    1. Клиент и сервер
    2. Клиент и база данных
    3. База данных и сервер
    4. Только серверной
14. Для чего нужны сценарии автоматизации?
    1. Для описания функциональности
    2. Для тестирования тестов
    3. Для поддержки тестов
    4. Для описания шагов теста
15. Что такое паттерн проектирования?
    1. Способ организации кода тестов
    2. Построение тестовой инфраструктуры
    3. Способ написания теста
    4. Правило наименования теста

16. Какой паттерн позволяет отделить одни тесты от других?

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

Итоговое тестирование

1. С помощью какого оператора можно изменить порядок вычисления в Python?

1. Скобки
2. Умножение
3. Возведение в степень
4. Остаток от деления

2. Чтобы результат вычисления не сильно исказился, какой функцией лучше всего преобразовывать вещественное число в целое?

1. floor()
2. ceil()
3. int()

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. Сколько вложенных блоков вы можете создать внутри другого вложенного блока?

1. Сколько угодно
2. 1
3. 2

5. Найдите неверное выражение

1. Счетчик цикла не является обязательным для циклов
2. Цикл while может заменить цикл for во всех случаях
3. Оператор continue заставляет python игнорировать команды стоящие после него
4. Цикл for может заменить цикл while во всех случаях

6. Какие объекты можно хранить в списках?

1. Только объекты, принадлежащие одному классу
2. Только строки
3. Любые
4. Только числа

7. 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
8. С какого элемента списка мы начинаем чтение параметров, переданных в sys.args?
    1. С первого
    2. Со второго
    3. С третьего
    4. С четвертого
9. Что будет, если класс вызвать как функцию?
    1. Вернется экземпляр класса
    2. Исключение - классы не являются вызываемыми объектами
    3. Вернется экземпляр класса, или None если в классе будет отсутствовать return
    4. Ничего не произойдет
10. В чем польза Name Mangling?
    1. Позволяет создавать несколько имен одному свойству
    2. Делает класс read-only
    3. Делает свойства объектов доступными только для чтения
    4. Защищает приватные свойства от случайной перезаписи другим значением
11. Какое основное назначение метода __set_name__?
    1. Для обеспечения обращения к свойствам объекта всегда через методы дескриптора
    2. Одинаковая обработка входных данных при создании разных свойств разных классов
    3. Для сохранения данных всегда только в локальном словаре __dict__
    4. Для создания слабых ссылок на свойства объектов в дескрипторе
12. Как можно создать свойство для метакласса, доступное для всех дочерних классов?
    1. self.property = 50
    2. cls.property = 50
    3. property = 50
    4. dict[‘property’] = 50
13. Выберите верное утверждение.
    1. Хорошие тесты могут быть документацией
    2. Нестабильные тесты — это плохо
    3. Автоматизированное тестирование — это автоматизация тестирования
    4. Все утверждения верны

14. 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
15. 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
16. Какого вида запросов не существует?
    1. POST
    2. GET
    3. PATCH
    4. POSTING
17. 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
18. Из каких частей состоит веб-приложение?
    1. Клиент и сервер
    2. Клиент и база данных
    3. База данных и сервер
    4. Только серверной
19. Что такое паттерн проектирования?
    1. Способ организации кода тестов
    2. Построение тестовой инфраструктуры
    3. Способ написания теста
    4. Правило наименования теста
20. Какой паттерн позволяет отделить одни тесты от других?
    1. Разборка окружения
    2. Параметризация
    3. Логирование
    4. Маркировка

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 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 видеоинформации на персональные компьютеры участников реализована при скорости интернет-соединения не менее 134 кбит/с.

Основные функции программы Контур.Школа:

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

Требованиям к информационным и учебно-методическим условиям

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. Python для гиков / [АзифМ.], пер. с англ. - СПб.: БХВ-Петербург, 2024. - 432 с.
2. Python. Лучшие практики и инструменты. 4-е изд. / [Михаил Яворски, Тарек Зиаде]; - СПб.: Питер, 2024. - 592 с.:
3. Python: большая книга примеров / [А. Л. Марченко]; — Москва: Издательство Московского университета, 2023. — 361, [1] с.
4. Безопасность веб-приложений на Python / [Бирн Д.], пер. с англ. С. С. Скобелева, А. Н. Киселева. – М.: ДМК Пресс, 2023. – 334 с.
5. Основы тестирования программного обеспечения. Учебное пособие для СПО / [Старолетов Сергей Михайлович]; Лань; 2023 г. - 192 с.
6. Тестирование программного обеспечения. Базовый курс/ [Святослав Куликов]; EPAM Systems, 2023 г. – 301 с.
7. Тестирование и отладка программного обеспечения: учебное пособие / [А. Н. Алпатов, А. А. Лобанов, Ю. С. Лобанова]; Киров: Изд-во МЦИТО, 2021г. – 58 с.
8. Что такое тестирование. Курс молодого бойца / [Назина Ольга]; БХВ; 2022г. – 592 с.
9. 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. 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://vladislaveremeev.gitbook.io/qa_bible/avtomatizaciya-beta/luchshie-praktiki-avtomatizacii
6. Тестирование Android приложений // Хабр, 2023. –

   URL: https://habr.com/ru/articles/352334/