



Программа курса "Тестировщик-автоматизатор на Python"

Раздел. О курсе и профессии

Модуль. Как мы будем учиться

Модуль. Введение в профессию

Раздел. Основы: компьютер, браузер, протоколы, программное обеспечение

Модуль 1. Сетевые протоколы и работа браузера

- **Изучается:** основные принципы работы компьютера; какие бывают протоколы, как работает браузер, базовый обзор HTML, DOM.
- **Результат:** студент может отвечать на вопросы про основные этапы жизненного цикла программного обеспечения, про то, как устроен компьютер, операционная система, браузер. Знает несколько базовых протоколов и их отличия (http / https / ftp / ssh).
- **Практика:** 1) ответить на вопросы теста по различиям между протоколами; по работе браузера. 2) Создать базовую HTML страницу, где будет описано резюме студента.
- **ПО и сервисы:** SublimeText; любой браузер (Google Chrome, Яндекс, Firefox), html5-editor.net

Модуль 2. Жизненный цикл программного обеспечения

- **Изучается:** что такое жизненный цикл ПО и как он обычно выглядит; зачем нужно тестирование, когда оно начинается и когда заканчивается, какие задачи стоят перед тестированием программного обеспечения
- **Результат:** студент может отвечать на вопросы про основные этапы жизненного цикла программного обеспечения.

Внимание! В процессе разработки и реализации в Skillfactory некоторые модули курса могут изменять название, содержание и место в структуре курса



- **Практика:** прохождение теста на понимание этапов жизненного цикла ПО / процессов
- **ПО и сервисы:** Канбан-доска trello.com

Раздел. Ручное тестирование

Модуль 3. Введение в тестирование

- **Изучается:** зачем нужно тестирование и какие виды тестирования бывают, как выглядит процесс тестирования на различных проектах.
- **Результат:** студент может отвечать на вопросы о том, как устроен процесс тестирования и знает основные виды тестирования и их отличия.
- **Практика:** ответить на вопросы теста про различия видом тестирования и их назначения / этапов жизни ПО, где применяется каждый вид тестирования.
- **ПО и сервисы:** иллюстративно — сервисы для создания ментальных карт (miro.com)

Модуль 4.1. Тест-дизайн

- **Изучается:** ключевые аспекты тест-дизайна, позволяющие тестировщикам эффективно и качественно составлять тест-кейсы и чек-листы
- **Результат:** студент может сам придумать и описать тестовые сценарии для простых систем (например, для калькулятора).
- **Практика:** написать 10 тестовых сценариев для функции отправки личных сообщений в социальной сети (либо для любого учебного проекта)
- **ПО и сервисы:** Word, Excel (или аналоги); Google docs; qase.io; sitechco.ru; pairwise online tool

Модуль 4.2. Техники тест-дизайна

- **Изучается:** составление тестовых сценариев на основании базовых техник тест-дизайна (эквивалентное разделение и анализ граничных значений; причина и следствие; предугадывание ошибки; диаграммы



перехода состояний и таблицы решений; pairwise), создание чек-листов и тест-кейсов.

- **Результат:** студент может сам придумать и описать тестовые сценарии для простых систем (например, для калькулятора).
- **Практика:** написать 10 тестовых сценариев для функции отправки личных сообщений в социальной сети (либо для любого учебного проекта)
- **ПО и сервисы:** Word, Excel (или аналоги); Google docs; qase.io; sitechco.ru; pairwise online tool

Модуль 5. Баг-репорты

- **Изучается:** жизненный цикл задач (багов), приоритеты и статусы, которыми отмечают переходы в другие состояния в баг-трекинговой системе; создание бага в баг-трекинговой системе JIRA, а также отслеживание/изменение статуса багов и задач, прикрепление к багам файлов/скриншотов/видео.
- **Результат:** студент умеет описывать баг репорты в JIRA и знает основные этапы жизненного цикла багов.
- **Практика:** протестировать тестовый проект, найти и описать баги (как минимум 5) в баг трекинговой системе, дождаться ""исправления"" и проверить что проблема исправлена в новой версии (и перевести свои баги в Done).
- **ПО и сервисы:** Jira

Модуль 6. Тестирование веб-интерфейсов

- **Изучается:** освоение инструментов и техник тестирования веб-интерфейсов с использованием популярных сценариев и Chrome DevTools для нахождения багов
- **Результат:** студент может тестировать веб-интерфейсы, с использованием самых популярных сценариев, а также умеет использовать Chrome DevTools
- **Практика:** протестировать тестовый проект, и найти минимум 5 багов, которые требуют знание HTML / CSS / Chrome Web Tools - например, найти JS ошибки на странице, найти неправильные ссылки.
- **ПО и сервисы:** Google Chrome



Модуль 7. Тестирование API с помощью Postman

- **Изучается:** базовые навыки работы с Postman на примере выполнения основных видов REST API запросов.
- **Результат:** студент умеет различать разные виды API запросов, умеет смотреть запросы в Network вкладке браузера
- **Практика:** составить список API запросов из вкладки Network, разделить запросы по типу (GET, POST, DELETE)
- **ПО и сервисы:** Postman

Модуль 8. Postman и отправка REST API запросов

- **Изучается:** тестирование с помощью Postman веб-приложений с использованием самостоятельно составленных тестов для REST API запросов и примеров тест-кейсов.
- **Результат:** студент может делать GET/POST запросы с использованием Postman.
- **Практика:** составить тесты для REST API с помощью Postman на основе документации для REST API и примеров тест кейсов.
- **ПО и сервисы:** Postman; petstore.swagger.io

Модуль 9. Основные команды GIT

- **Изучается:** системы контроля версий; GIT; базовые команды GIT; создание проекта на github; что такое ветки; как делать коммиты и README.
- **Результат:** студент умеет создавать github репозитории, работать с ветками и коммитами, умеет пользоваться основными командами git.
- **Практика:** создать собственный профиль и репозиторий на github.com, разместить в репозитории решение базовых задач по программированию и файл README с описанием решения.
- **ПО и сервисы:** git-scm.com/downloads; desktop.github.com; github.com

Модуль 10. Проект № 1. Мой первый заказ на тестирование сайта



- **Изучается:** уровень готовности студента к участию в ручном тестировании реального проекта; уровень сформированности необходимых hard skills по чек-листу (тест-опроснику); способность указывать эти навыки при составлении резюме на headhunter и создании аккаунтов в github, linkedin
- **Результат:**
 - 1) Выполнено комплексное ручное тестирование сайта на основе требований Заказчика.
 - 2) Составлено резюме на headhunter.ru с описанием навыков и опыта ручного тестирования, со ссылками на аккаунты в github, linkedin.

Раздел. Основы Python

Модуль 11. Введение в Python

- **Изучается:** введение в язык Python, установка интерпретатора и зависимостей, основные типы данных.
- **Результат:** студент умеет запускать программы на языке Python и может решать базовые задачи по программированию.
- **Практика:** студент умеет запускать программы на языке Python и может решать базовые задачи по программированию.
- **ПО и сервисы:** SublimeText; Python 3.6 или более; repl.it/languages/python3

Модуль 12. Основные типы данных и конструкции в Python

- **Изучается:** основные типы данных и конструкции в Python (циклы и ветвления)
- **Результат:** студент умеет использовать циклы и ветвления на Python, может решать более сложные задачи на Python с использованием этих конструкций.
- **Практика:** решить более сложные задачи на строки / массивы на языке Python, закоммитить их в свой репозиторий.
- **ПО и сервисы:** SublimeText; Python 3.6 или более; repl.it/languages/python3

Модуль 13. Функции в Python

Внимание! В процессе разработки и реализации в Skillfactory некоторые модули курса могут изменять название, содержание и место в структуре курса



- **Изучается:** функции в Python и стандартные функции языка для обработки данных в Python.
- **Результат:** студент умеет писать и использовать функции в Python, и может использовать стандартные функции, такие как `dir`, `len`, `random`, `zip`, `map`, `sorted`.
- **Практика:** решить задачи (10), которые требуют использование простых функций - подсчет буквы `a` в слове, сложение всех чисел больше 10 и пр.
- **ПО и сервисы:** SublimeText; Python 3.6 или более; repl.it/languages/python3

Модуль 14. Классы в Python

- **Изучается:** основные принципы объектно-ориентированного программирования; понятие "класс", написание класса, хранение/изменение его параметров; методы класса; наследование; полиморфизм.
- **Результат:** студент умеет писать собственные классы на языке Python, знает, как работает наследование и полиморфизм.
- **Практика:** решить две задачи на создание классов/наследование, написать декоратор для функции который изменяет поведение функции.
- **ПО и сервисы:** SublimeText; Python 3.6 или более; repl.it/languages/python3

Модуль 15. Сортировки и операции со строками

- **Изучается:** синтаксис работы со строками и списками в Python, алгоритмы сортировки; функции для сортировки элементов массива, для обработки строк; решение задачи.
- **Результат:** умеет решать задачи на сортировку массивов, умеет пользоваться срезами в Python
- **Практика:** решить 15 задач на сортировку или выборку из строки, в которых нужно читать и записывать данные в файлы на диске.
- **ПО и сервисы:** SublimeText; Python 3.6 или более; repl.it/languages/python3

Модуль 16. Чтение и запись информации в файл

**SKILLFACTORY**

Программа разработана специально для Skillfactory.

Реализуется на онлайн-платформе Skillfactory в рамках курса-симулятора.

- **Изучается:** синтаксис работы с файлами в Python; программы, которые читают и пишут информацию в файлы.
- **Результат:** умеет писать в текстовый файл и JSON и читать из файла данные с помощью Python.
- **Практика:** решить 10 задач на запись / чтение из файлов и использование стандартных алгоритмов (сортировка, подсчет элементов)
- **ПО и сервисы:** SublimeText; Python 3.6 или более; repl.it/languages/python3

Модуль 17. Решение алгоритмических задач

- **Изучается:** основные алгоритмы, которые понадобятся для решения задач по программированию на интервью.
- **Результат:** умеет решать основные типы алгоритмических задач, которые задают на интервью, умеет правильно выбирать алгоритмы для решения, проверять собственный код и доводить решение до конца.
- **Практика:** решение 20 алгоритмических задач, которые используются для интервью.
- **ПО и сервисы:** SublimeText; Python 3.6 или более; repl.it/languages/python3

Модуль 18. Проект № 2. Моя команда получила заказ

- **Изучается:** энергосберегающие и эффективные коммуникации в команде; работа в команде; работа с заказчиком; поиск и выполнение заказа (на фрилансе).
- **Результат:** умеет общаться с разработчиками и другими QA специалистами, а также умеет общаться с заказчиком на фриланс проектах.
- **Практика:** формирование команды из участников курса, совместный выбор проекта и его реализация.

Раздел. Автоматизация тестирования

Модуль 19. Тестовый фреймворк Pytest и автотесты для REST API

Внимание! В процессе разработки и реализации в Skillfactory некоторые модули курса могут изменять название, содержание и место в структуре курса



- **Изучается:** тестовый фреймворк Pytest. Запуск первых простых тестов. Создание автотестов для REST API. Библиотека requests
- **Результат:** умеет использовать библиотеку requests и писать автоматизированные тесты на Python & PyTest.
- **Практика:** написать 10 базовых автоматизированных тестов для REST API с использованием фреймворка PyTest
- **ПО и сервисы:**

Модуль 20. Тестовый дизайн для REST API

- **Изучается:** основные сценарии, которые необходимо проверять при автоматизированном тестировании REST API.
- **Результат:** умеет составлять список сценариев для REST API сервисов и умеет писать сложные автотесты для REST API на Python.
- **Практика:** покрыть тестовый REST API сервис автоматизированными тестами на Python (GET, POST, DELETE, UPDATE + базовая авторизация и авторизация с помощью cookies).
- **ПО и сервисы:**

Модуль 21. Декораторы в Python

- **Изучается:** что такое декоратор в Python и для чего он применяется; как его создать.
- **Результат:** умеет писать собственные декораторы (простые) на языке Python.
- **Практика:** написать декоратор для функции который измеряет время работы функции + декоратор который печатает с какими параметрами была вызвана функция
- **ПО и сервисы:** SublimeText; Python 3.6 или более; repl.it/languages/python3

Модуль 22. Параметризация тестов в PyTest

- **Изучается:** синтаксис для передачи параметров для автоматизированных тестов на Python



- **Результат:** умеет писать сложные параметризованные автоматизированные тесты для REST API, умеет применять декораторы PyTest.
- **Практика:** написать параметризованные тесты для REST API сервиса, полностью покрыть сервис базовыми регрессионными проверками (должно быть 100+ автотестов с перебором значений параметров).
- **ПО и сервисы:**

Модуль 23. Локаторы элементов на странице и полезные плагины

- **Изучается:** локатор элементов; какие бывают; как найти; в каких ситуациях какой тип локаторов лучше использовать.
- **Результат:** может определять локаторы элементов на странице, умеет пользоваться плагинами, например XPath Helper, а также плагины для скриншота всей страницы и др.
- **Практика:** даем студенту 20 элементов, для которых нужно определить различные виды локаторов и написать их в таблицу + задания на использование плагинов
- **ПО и сервисы:** Google Chrome

Модуль 24. Selenium и автоматизация веб интерфейсов

- **Изучается:** проект Selenium и его архитектура; синтаксис библиотеки Selenium; использование PyTest + Selenium.
- **Результат:** умеет писать базовые автотесты с использованием библиотеки Selenium и фреймворка PyTest.
- **Практика:** написать базовые автотесты для веб интерфейса, автоматизировать ввод текста на странице, нажатие кнопки.
- **ПО и сервисы:**

Модуль 25. Поиск элементов с помощью Selenium, ожидание элемента на странице

- **Изучается:** функции библиотеки Selenium и ее основные возможности; методы поиска элементов на странице с помощью Selenium; ожидание в



Selenium; создание автотестов, в которых есть взаимодействие с различными элементами и ожидание элемента на странице.

- **Результат:** умеет работать с разными типами локаторов для элементов и умеет взаимодействовать со сложными элементами на странице, умеет использовать умные ожидания в selenium.
- **Практика:** написать более сложные сценарные тесты с использованием Selenium, где требуется ожидание элементов, поиск элемента на XPath, id, class_name.
- **ПО и сервисы:**

Модуль 26. Паттерн Page Object и работа Smart Page Object

- **Изучается:** паттерны программирования; паттерн PageObject, его назначение; паттерн Page Factory, его назначение.
- **Результат:** умеет писать автоматизированные тесты на Selenium с применением паттерна Page Object (используем готовый класс).
- **Практика:** написать Page Objects для 2-3 страниц, написать 15 сценарных автоматизированных тестов для этих страниц.
- **ПО и сервисы:**

Модуль 27. Дополнительные возможности Selenium

- **Изучается:** популярные методы для управления браузером, которые пригодятся при написании сложных автоматизированных сценариев.
- **Результат:** умеет писать сложные тесты с использованием Selenium, умеет выполнять произвольный JS код на странице.
- **Практика:** написать сложные автоматизированные сценарии для веб интерфейса, где надо удалять элементы на странице, дожидаться загрузки изображений на странице, скролить страницу до нужного элемента.
- **ПО и сервисы:** Selenium

Модуль 28. Инфраструктура для запуска тестов

- **Изучается:** инфраструктура для запуска тестов (Jenkins CI, Allure, Selenoid).

Внимание! В процессе разработки и реализации в Skillfactory некоторые модули курса могут изменять название, содержание и место в структуре курса

**SKILLFACTORY**

Программа разработана специально для Skillfactory.

Реализуется на онлайн-платформе Skillfactory в рамках курса-симулятора.

- **Результат:** умеет создавать сборки в Jenkins, умеет запускать автоматизированные тесты внутри Selenium кластера, умеет делать Allure отчеты после запуска тестов.
- **Практика:** написать автотесты в связке PyTest & Selenium & Allure, и сделать сборку в Jenkins которая запускает тесты внутри Selenium кластера и автоматически делает Allure отчет.
- **ПО и сервисы:** Jenkins CI, Allure, Selenium

Модуль 29. Проект № 3. Как пройти тестовое техническое интервью и получить работу

- **Изучается:** самопрезентация; техническое интервью; выполнение тестовых заданий
- **Результат:** умеет делать самопрезентацию, умеет проходить техническое интервью на позицию Junior QA Automation Engineer, умеет отвечать на типичные вопросы.
- **Практика:** пройти тестовое техническое интервью и решить тестовые задания.

Внимание! В процессе разработки и реализации в Skillfactory некоторые модули курса могут изменять название, содержание и место в структуре курса