
Мануальное тестирование + интернатура

Программа курса

Ты достоин лучшего образования

Английский
(для желающих)
16 занятий
24 часа

Мануальное тестирование
4 месяца
45-50 занятий в зависимости от уровня группы
**90-100 часов + 90-100 часов на выполнение домашней
работы**

Интернатура
2 месяца part-time
работы

МАНУАЛЬНОЕ ТЕСТИРОВАНИЕ

Занятия: онлайн, 2-3 раза в неделю

Продолжительность: 4 месяца

Знания: IT + QA

Преподавателей: 4 (от 15 лет опыта у каждого)

Модулей: 13+1

Консультаций: неограничено

ЗАНЯТИЯ

Формат: онлайн, 2-3 раза в неделю

Продолжительность: 1.5-2 часа

Язык: English (минимальный уровень - B1)

Вопросы на занятии: неограничено

Вопросы между занятиями: неограничено

Группа: 15 студентов

Домашняя работа (в среднем): 2-3 часа после каждого занятия

ЧТО, ЕСЛИ МОЙ АНГЛИЙСКИЙ НЕ ТАК ХОРОШ?

Курс IT-английского: перед началом первого модуля

Продолжительность: 1 месяц

Периодичность: 4 раза в неделю

Требуемый уровень: A2-B1

Группа: 5 студентов

Интеграция с курсом мануального тестирования:

- Лекторы участвуют в 4 занятиях, проводят игры
- Используются материалы курса мануального тестирования

Модуль 0. АНГЛИЙСКИЙ ДЛЯ ТЕСТИРОВЩИКОВ (отдельный спецкурс)

**IT
терминология**

Чтение и
понимание

**IT
коммуникации**

Аудирование и
говорение

**Интеграция с
нашим
курсом**

Наши менторы
участвуют в
некоторых
занятиях

**Наши
материалы**

На занятиях
используются
требования к
нашим
приложениям

Модуль 1. ВВЕДЕНИЕ В ПРОФЕССИЮ

Миссия выполнима

Кто такие
тестировщики и
зачем они

Как устроено тестирование

От кнопочек на UI к
серверу по сети - и
обратно

Модуль 2. ГОТОВИМСЯ ТЕСТИРОВАТЬ: КАК УСТРОЕН LINUX

Linux в жизни тестировщика

Зачем тестерам Linux?
История Linux
Использование Terminal и SSH

Пользователи и группы

Основная концепция
Привилегии: su, sudo
Права доступа: chown, chmod

Основы Linux

Концепция файловой системы UNIX
cd, pwd, ls
cat, less, man
cp, mv, rm, mkdir, ln

Процессы и сервисы

Список процессов: ps, pgrep, top
Управление процессами: kill, pkill, killall
Управление сервисами: systemctl

Чтение файлов

head, tail
sort, uniq, wc
Использование pipe и regExp

Создание файлов

touch, echo, nano
Перенаправление потоков, stdout / stderr
Последовательность выполнения: &&, “;”, ||

Поиск

Поиск файлов: find
Сбор данных: uname, df, mount, free, cpuinfo
Архивирование и компрессия: tar, gzip, bzip2
Передача файлов: scp

Модуль 3. ГОТОВИМСЯ ТЕСТИРОВАТЬ: КАК УСТРОЕНЫ СЕТИ

Поиск неисправностей в сетевом окружении

Зачем это тестировщику

Основы сетей

Как информация перемещается в сети

Сетевые протоколы

Какие есть способы передать информацию

Транспортный уровень

И еще немного о протоколах

Прикладные протоколы

И теперь уже совсем крутые вещи про сеть: ни один баг не пройдет

Модуль 4. ГОТОВИМСЯ ТЕСТИРОВАТЬ: КАК УСТРОЕН УРОВЕНЬ ДАННЫХ

Основы реляционной теории

Таблицы и отношения
Ключи
Констрейнты
ER-диаграммы

SQL Subqueries

```
SELECT * from  
table_name where  
value in (SELECT ...)
```

Основы SQL

SELECT, UPDATE,
INSERT, DELETE

Database Connection

DBeaver
Параметры
соединения
Метаданные
Пароли

SQL Joins

LEFT, RIGHT, FULL,
SELF

Database Tools

SQL Editor
Лог
Диаграммы

Модуль 5. ГОТОВИМСЯ ТЕСТИРОВАТЬ: КАК УСТРОЕН УРОВЕНЬ СЕРВИСОВ

REST запросы

GET
POST
PUT
DELETE
HEAD
OPTIONS
CONNECT
TRACE

REST сервисы

HTTP
URI
Protocol
Host
Port
Path
Status codes
JSON
XML

Postman

Основы
использования и
коллекции

Development Tools

Console
Network
Memory
Application

Модуль 6. ГОТОВИМСЯ ТЕСТИРОВАТЬ: КАК УСТРОЕН УРОВЕНЬ WEB UI

UI объекты

DOM-дерево

Development Tools

Elements
Cookies
Cache

Основы UI скриптов

JavaScript
Script elements

Параметры UI объектов

XPath
Inspect Element
Properties

Модуль 7. ГОТОВИМСЯ ТЕСТИРОВАТЬ: TROUBLESHOOTING

Архитектура 3- уровневых приложений

Основные цели и
использование
Компоненты: frontend,
backend, DBMS
Тестовое приложение

Общие методы локализации ошибок

Типичные случаи ошибок в
трехуровневых приложениях
Методика локализации
Получение информации с клиента,
фронта, бэка, базы

Применение Linux и сетей

ps, ss/netstat, systemctl,
journalctl, egrep, nc, tcpdump

Практика troubleshooting

Неправильный логин и пароль
Несоответствие записей в БД
Упавший сервер
Бэк не может залогиниться в базу
Проблемы с Firewall

Модуль 8. ТЕСТИРОВАНИЕ ТРЕБОВАНИЙ

Требования на разных уровнях

UI уровень
API уровень
DB уровень

Анализ качества требований

IBM классификация
INVEST
Другие
классификации

UML

Структурная
диаграмма
Use Case диаграмма
Activity диаграмма
State-Machine
диаграмма
Sequence диаграмма

Модуль 9. ТИПЫ И УРОВНИ ТЕСТИРОВАНИЯ

Классификация тестирования: пирамида тестирования

Unit тестирование
Integration
тестирование
Mocks
Stubs
System тестирование
Acceptance
тестирование

Классификация тестирования: виды тестирования, регрессионное тестирование

Blackbox
Gray box
Whitebox
Функциональное тестирование
Нефункциональное тестирование
Smoke тестирование
Sanity тестирование
Полная регрессия
Связь между типами тестирования

Модуль 10. ТЕСТ-ДИЗАЙН

Введение в тест-дизайн

Типы и стадии разработки

Классы эквивалентности и граничные значения

Простые и сложные объекты на разных уровнях приложения

Таблицы принятия решений

Простые и сложные объекты на разных уровнях приложения

Попарное тестирование

Простые и сложные объекты на разных уровнях приложения

Тестирование состояний и переходов

Простые и сложные объекты на разных уровнях приложения

Тестирование на основе сценариев

Уровни черного ящика

Модуль 11. ТЕСТОВАЯ ДОКУМЕНТАЦИЯ

Чек-листы

Правила
использования
Комбинация с тест-
кейзами

Тест-кейсы

Поля
TMS
Лучшие практики

Тест-сьюты и тестовые прогоны

Правила
использования

Тест-планы

Master тест-план
Child тест-план
Правила
использования
Секции

Defect Tracking System (DTS)

Workflow дефектов
Поля дефекта
Severity и priority

Описание дефекта

Лучшие практики

Модуль 12. НАЧАЛА АВТОМАТИЗАЦИИ ДЛЯ МАНУАЛЬНЫХ ТЕСТИРОВЩИКОВ

Git

git clone
git init
git add
git checkout
git commit
git push
Branching strategy

Автоматизация и CI/CD

Кто такой автоматизатор
Коммуникация с командой
автоматизации
Принципы автоматизации
Стратегии CI/CD

Основы нужных программ

Docker
Kubernetes
RabbitMQ

Модуль 13. МЕТОДОЛОГИЯ РАЗРАБОТКИ ПО И КОМАНДА

Методологии разработки

Вотерфол
Agile: SCRUM
Agile: Kanban
Спиральная модель
V-модель

Agile: Деловая игра

Jira
Таски
Стендап
Грумминг
Пленинг
Ретроспектива
Business demo

БОНУС ДЛЯ БЫСТРЫХ ГРУПП

Исследовательское тестирование и тестовые туры

Monkey

Gorilla

Ad hoc

Угадывание ошибок

Тестовые туры

Сессии и отчеты по исследовательскому
тестированию

С ЧЕМ ТЫ НАУЧИШЬСЯ РАБОТАТЬ



ИНТЕРНАТУРА

Продолжительность: 2 месяца

Реальный проект: Да

Международная компания: Да

Практика в вашем CV: Да

Применение знаний: Да

Возможность стать лидером: Да

Консультации лекторов: неограниченно

ЭТАПЫ ИНТЕРНАТУРЫ

Этап 1 (1-2 недели)

Ментор контролирует команду каждый день

Этап 2 (8 недель)

Один-два лида докладывают ментору об успехах команды

Этап 3 (2 недели)

Разбор полетов, подготовка резюме

ТЕСТОВОЕ ИНТЕРВЬЮ

Продолжительность: 1 час для каждого успешного студента

Менторов: 1-2

Языки: Русский + English

Обратная связь: Да

Помощь с резюме: Да

Помощь с трудоустройством: Возможна

КАК С НАМИ СВЯЗАТЬСЯ

Подробности: <https://Mentorpiece.ru/курсытестировщика>

Telegram-канал курса: [@to_be_qa](https://t.me/to_be_qa)

Telegram-чат курса: [@to_be_qa_chat](https://t.me/to_be_qa_chat)

Бесплатные индивидуальные консультации:

[@MentorPiece_consult_ru_bot](https://t.me/MentorPiece_consult_ru_bot)