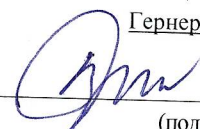


Автономная некоммерческая образовательная организация дополнительного профессионального образования «Учебный Центр Информационные Бизнес Системы»
(АНО ДПО «УЦ ИБС»)

Утверждаю
Директор АНО ДПО «УЦ ИБС»
Гернер В.А.



(подпись)

«09» января 2025 г.



ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

«Fullstack-тестировщик»

ДЛЯ ОБУЧЕНИЯ В РАМКАХ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
(ПРОФЕССИОНАЛЬНАЯ ПЕРЕПОДГОТОВКА)

Москва 2025 г.

1. ОБЩИЕ ПОЛОЖЕНИЯ

Дополнительная профессиональная программа по профессиональной переподготовке «Fullstack-тестировщик» разработана на основе следующих нормативных документов:

- Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказ Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- Профессиональный стандарт «Программист» №4, утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 20.07.2022 № 424н.

2. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ, ЕЕ МЕСТО В УЧЕБНОМ ПРОЦЕССЕ

2.1 Целью преподавания дисциплины является

Повышение квалификации по образовательной программе «Fullstack-тестировщик», в том числе сформировать у слушателей набор знаний и навыков для организации и проведения тестирования программного обеспечения.

2.2. Задачи изучения дисциплины:

- Изучить принципы и этапы тестирования программного обеспечения;
- Освоить методики работы с документацией и требованиями;
- Овладеть техниками тест-дизайна и методологиями тестирования;
- Изучить классификацию дефектов и принципы работы с ними;
- Освоить навыки документирования тестов и ведения тестовой отчетности;
- Овладеть практиками тестирования как вручную, так и с применением автоматизированных инструментов;
- Изучить жизненный цикл ПО и его связь с задачами тестирования на различных этапах разработки.

2.3 Связь дисциплины с другими учебными дисциплинами

Изучение дисциплины «Fullstack-тестировщик» предполагает наличия у слушателей специальных знаний по теории программирования (базовые знания в области программирования на языке Java), архитектуре ПО, веб-технологиям и автоматизации тестирования. Также курс опирается на знания, полученные в рамках дисциплин по DevOps-практикам и системам контроля версий, что позволяет эффективно интегрировать тестирование в процессы непрерывной интеграции и доставки (CI/CD).

3. СОДЕРЖАНИЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Образовательная программа «Fullstack-тестировщик» представляет собой комплексный курс подготовки универсального специалиста по качеству программного обеспечения. Программа охватывает ключевые аспекты ручного и автоматизированного тестирования, начиная с анализа требований и построения тестовой модели, заканчивая реализацией автотестов, интеграцией с CI/CD и нагрузочным тестированием.

Категория слушателей программы:

- начинающие тестировщики ПО;
- ручные тестировщики, желающие перейти в автоматизацию;
- студенты технических специальностей, интересующиеся QA;
- специалисты смежных ИТ-направлений, переходящие в тестирование;
- стажёры и junior-специалисты в ИТ-командах.

К освоению учебной программы допускаются:

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

Базовые компетенции, которыми должен владеть слушатель программы:

- Владение базовыми ИТ-навыками: работа с компьютером, файловой системой, браузерами, почтой;
- Понимание жизненного цикла ПО и роли тестирования в разработке;
- Навыки логического мышления и анализа;
- Базовое представление о веб-технологиях (HTML, браузеры, HTTP);
- Начальные навыки работы с системами трекинга (например, Jira) и таблицами (Excel, Google Sheets);
- Желательно: базовое знание SQL и/или Java на уровне новичка.

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Программа «Fullstack-тестировщик» направлена на развитие у слушателей компетенций по ручному и автоматизированному тестированию, включая работу с современными инструментами, методологиями разработки и CI/CD-практиками. Курс готовит специалистов, способных профессионально участвовать в процессах обеспечения качества ПО на всех этапах жизненного цикла продукта.

После завершения обучения слушатели смогут:

1. Понимать жизненный цикл разработки ПО, типы и уровни требований, отличать функциональные и нефункциональные характеристики качества;
2. Проводить анализ требований, выявлять дефекты на ранних этапах, составлять тестовую документацию: чек-листы, тест-кейсы, сценарии, планы тестирования;
3. Составлять матрицы покрытия и трассируемости, системно оценивать полноту тестирования;
4. Применять техники тест-дизайна: эквивалентные классы, граничные значения, диаграммы переходов состояний, комбинаторные методы, доменные модели и др.;
5. Выполнять ручное функциональное, конфигурационное, производительное и тестирование безопасности веб-приложений;
6. Работать с логами, протоколами HTTP, архитектурой клиент-серверных приложений;
7. Использовать Jira, Zephyr и Test IT для построения тестовой модели, управления тестами и багами;
8. Писать базовые SQL-запросы для проверки корректности данных и связей между таблицами;
9. Работать с форматами XML, JSON и схемами XSD при тестировании интеграций;
10. Использовать Postman для ручного тестирования API, анализировать запросы и ответы в формате REST/SOAP/WebSocket;
11. Создавать и поддерживать автотесты backend (на базе Java и RestAssured) и frontend (Selenium WebDriver, Selenide, Playwright);
12. Применять фреймворки автоматизации и отчетности: JUnit, Cucumber (BDD), Allure;
13. Разворачивать и запускать автотесты в контейнеризированной среде (Docker, Kubernetes, Selenoid, Moon);
14. Понимать и применять практики CI/CD, настраивать запуск тестов в Jenkins и других CI-системах;
15. Писать тесты с использованием Java, применять ООП, работать со строками, коллекциями, потоками ввода-вывода, лямбдами и Stream API;
16. Использовать Git для командной работы: работа с ветками, коммитами, разрешение конфликтов;
17. Выполнять нагрузочное тестирование с помощью Apache JMeter, строить модели нагрузки и анализировать результаты;
18. Эффективно взаимодействовать с командой разработки, презентовать результаты тестирования, участвовать в Agile-процессах (Scrum, Kanban).

5. МОДУЛЬНЫЙ ПЛАН ПРОГРАММЫ

Программа состоит из 8 курсов, содержащих теорию, видео-разборы, задачи и кодовую базу. По итогам обучения предусмотрено выполнение дипломной работы.

№	Наименование модулей	Объем в академ. часах		
		Всего часов	Теория	Практика
1	2	3	4	5
1.	SQA-I-001 Практика проведения тестирования	14	6	8
2.	SQA-I-002 Вспомогательные технологии и инструменты тестировщика	22	6	16
3.	SQA-I-003 Тестирование веб-приложений	16	4	12
4.	SQA-I-004 Java и Git для автоматизации тестирования	56	20	36
5.	SQA-I-005 Автоматизация тестирования back-end	50	20	30
6.	SQA-I-006 Автоматизация тестирования front-end	52	20	32
7.	SQA-I-007 Сопровождение автотестов и написание отчетов	54	20	34
8.	SQA-I-008 Взаимодействие в команде	4	2	2
9.	Итоговая дипломная работа	8		8
ИТОГО		276	98	178

6. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН МОДУЛЕЙ ПРОГРАММЫ

1. SQA-I-001 Практика проведения тестирования

№	Тема	Количество часов	Форма контроля
1	Тестирование документации и требований	2	- тестовые вопросы - задачи для самостоятельной практики - живая практика (лабораторные) с экспертом-преподавателем
2	Подготовка документации для тестирования: чек-листы, тест-кейсы, тестовые сценарии, планы тестирования	2	
3	Составление матрицы покрытия требований тестовыми сценариями	2	
4	Типы и уровни тестирования	2	
5	Методологии разработки сценариев и тест-идей на основе моделей	2	
6	Классификация дефектов. Определение возможных багов и несоответствий качеству на стадии ТЗ	2	
7	Обработка ошибок и работа с дефектами	2	
	Итого	14	

2. SQA-I-002 Вспомогательные технологии и инструменты тестировщика

№	Тема	Количество часов	Форма контроля
1	Инструменты управления процессами: Jira	6	- тестовые вопросы - задачи для самостоятельной практики - живая практика (лабораторные) с экспертом-преподавателем
2	Инструменты управления процессами: Test IT	4	
3	Основы SQL	8	
4	Форматы данных для обмена: XML, XSD, Json	4	
	Итого:	22	

3. SQA-I-003 Тестирование веб-приложений

№	Тема	Количество часов	Форма контроля
1.	Архитектура и структура веб-приложений	6	- тестовые вопросы - задачи для самостоятельной практики - живая практика (лабораторные) с экспертом-преподавателем
2.	Тестирование веб-приложений	6	
3.	Логи в тестировании	4	
	Итого	16	

4. SQA-I-004 Java и Git для автоматизации тестирования

№	Тема	Количество часов	Форма контроля
1	Программирование на Java. Структура и операторы.	4	- тестовые вопросы - задачи для самостоятельной практики - живая практика (лабораторные) с экспертом-преподавателем
2	Типы данных и объектно-ориентированное программирование (ООП)	6	
3	Работа со строками и массивами	2	
4	Углубленный дизайн классов	6	
5	Дженерики и коллекции	4	
6	Задачи ввода-вывода Java	4	
7	Лямбда-выражения и встроенные функциональные	8	

	интерфейсы. Stream API		
8	Избранные шаблоны проектирования	6	
9	Продвинутая обработка исключений	4	
10	Аннотации и рефлексия	4	
11	Ведение проекта в Git	8	
	Итого	56	

5. SQA-I-005 Автоматизация тестирования back-end

№	Тема	Количество часов	Форма контроля
1	Автоматизация тестирования back-end	20	- тестовые вопросы - задачи для самостоятельной практики - живая практика (лабораторные) с экспертом-преподавателем
2	Применение Postman для выполнения запросов	20	
3	Технологии асинхронного обмена сообщениями: RabbitMQ и Kafka в сравнении	10	
	Итого	50	

6. SQA-I-006 Автоматизация тестирования front-end

№	Тема	Количество часов	Форма контроля
1	Основы тестирования и методологии. Процесс работы JUnit	10	- тестовые вопросы - задачи для самостоятельной практики - живая практика (лабораторные) с экспертом-преподавателем
2	Проектирование и структурирование тестов Selenium WebDriver и взаимодействие с браузером	14	
3	Фреймворки для тестирования	16	
4	Модели и методологии разработки и тестирования. Фреймворк для отчетов - Cucumber и Allure	12	
	Итого	52	

7. SQA-I-007 Сопровождение автотестов и написание отчетов

№	Тема	Количество часов	Форма контроля
1	Linux, основы работы с командной строкой и правами	6	- тестовые вопросы - задачи для самостоятельной практики - живая практика (лабораторные) с экспертом-преподавателем
2	Инструменты и методологии CI/CD	6	
3	Введение в Jenkins: определение, цели и задачи	8	
4	Контейнеризация и автоматизация - Docker	8	
5	Развёртывание приложений в Kubernetes и доступ через сервисы	6	
6	Selenoid	8	
7	Moop	6	
8	Apache JMeter - использование в нагрузочном тестировании	6	
	Итого	54	

8. SQA-I-008 Взаимодействие в команде

№	Тема	Количество часов	Форма контроля
1	Роль и взаимодействие тестировщика в проектной команде	1	- тестовые вопросы - задачи для самостоятельной практики
2	Методологии разработки ПО	1	
3	Карьерное развитие и собеседования	1	
4	Ключевые soft skills	1	

	Итого	4	
--	--------------	---	--

9. Итоговая дипломная работа

№	Тема	Количество часов	Форма контроля
1	Итоговая дипломная работа	8	• Онлайн защита
	Итого	8	

7. ФОРМА ОБУЧЕНИЯ

Обучение проходит в системе дистанционного обучения в смешанном формате:

- Материалы курсов изучаются в системе дистанционного обучения (СДО). По окончании теоретической части каждого курса программы предусмотрен тест для проверки уровня знаний.
- Практические групповые сессии с преподавателем проходят в режиме вебинара.

8. ФОРМА КОНТРОЛЯ

Контроль усвоения учебной программы проводится в различных формах:

1. Решение практических задач;
2. Промежуточный контроль (тестовый контроль);
3. Тестовый итоговый контроль;
4. Защита дипломной работы.

Курс считается успешно пройденным при достижении следующих результатов:

1. В системе дистанционного обучения прогресс по изучению материалов не менее 80%;
2. Тестовый итоговый контроль сдан не менее чем на 80%;
3. Защита итоговой дипломной работы на оценку не ниже удовлетворительно.

При успешном прохождении итоговой аттестации или получении на итоговой аттестации удовлетворительных результатов, выдается Диплом о профессиональной переподготовке установленного образца.

Лицам, не прошедшим итоговой аттестации или получившим на итоговой аттестации неудовлетворительные результаты выдаётся Справка об обучении по программам дополнительного профессионального образования в соответствии с Положением о порядке выдаче и форме данной справки.

Примеры тестовых вопросов:

1. Какой командой в Git создается новая ветка и сразу выполняется переход на нее?
 - a) `git branch <название_ветки>`
 - b) `git checkout <название_ветки>`
 - c) `git switch <название_ветки>`
 - d) `git checkout -b <название_ветки>`
2. Укажите имя столбца, который является первичным ключом, если его добавили в таблицу следующим запросом:
`ALTER TABLE persons_code`
`ADD CONSTRAINT fk1 PRIMARY KEY (code);`
 - a) `persons_code`
 - b) `fk1`
 - c) `persons`
 - d) `code`
3. В чем отличие методов `isDisplayed()` и `exists()` в `SelenideElement`?
 - a) `exists()` определяет, виден ли элемент и есть ли он в DOM. А `isDisplayed()` только наличие элемента в DOM.
 - b) Эти методы идентичны.
 - c) `isDisplayed()` определяет, виден ли элемент и есть ли он в DOM. А `exists()` только наличие элемента в DOM.
 - d) `exists()` применим только к коллекциям элементов.
4. Как с помощью Selenide искать внутренние веб-элементы?
 - a) С помощью методов `find()` / `findAll()`, вызываемых на объекте начального веб-элемент.

- b) С помощью аннотации @FindBy.
 - c) С помощью статических методов Selenium.find() / Selenium.findAll().
 - d) Selenium не имеет подобной функциональности.
5. Какие ассерты можно применять в автотестах, если на проекте используется Playwright?
- a) Встроенные PlaywrightAssertions, а также любые сторонние библиотеки ассертов.
 - b) Только встроенные PlaywrightAssertions.
 - c) Только ассерты JUnit.
 - d) Только ассерты AssertJ, которые требуется устанавливать отдельно.
6. Можно ли выполнять HTTP-запросы с помощью Playwright?
- a) Да, но требуется самостоятельная установка сторонних библиотек, таких как RestAssured, которые будут работать под капотом.
 - b) Да, Playwright предоставляет такой функционал из коробки.
 - c) Нет, Playwright не может использоваться для выполнения HTTP-запросов.

9. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ПРОГРАММЕ

1. Сэм Канер и др. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений./Пер с англ./-К.: Издательство «ДиаСофт», 2001.-544с.
2. Савин Р. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах. – 2017.
3. Пероцкая В. Н., Градусов Д. А. Основы тестирования программного обеспечения. Учебное пособие. – 2019.
4. Куликов С. Тестирование программного обеспечения. Базовый курс. – 2021.
5. Тамре Л. Введение в тестирование программного обеспечения. – 2018.
6. Русов А. Автоматизация тестирования веб-приложений. – 2020.
7. Desikan S., Ramesh G. Software Testing: Principles and Practices. – Pearson Education, 2006.
8. Дастин Э., Рэшка Д., Пол Дж. Автоматизированное тестирование программного обеспечения. – Лори, 2003.
9. Humble J., Farley D. Continuous Delivery. – Addison-Wesley, 2010.
10. Graham D., Fewster M. Experiences of Test Automation: Case Studies of Software Test Automation. – Addison-Wesley, 2012.
11. Sambamurthy M. Test Automation Engineering Handbook. – Packt Publishing, 2023.
12. Horstmann C. Big Java Early Objects. – 7th Ed. – Wiley, 2018.
13. Sierra K., Bates B. OCA Java SE 8 Programmer I Exam Guide (Exam 1Z0-808). – McGraw-Hill, 2015.
14. Varabasi B., Belida S. Introducing Maven. – Apress, 2014.
15. Chacon S., Straub B. Pro Git. – 2nd Ed. – Apress, 2014.
16. Appel F. Testing with JUnit. – Packt Publishing, 2015.

Учебная программа разработана: Отделом разработки и методологии АНО ДПО «УЦ ИБС»