

Департамент информатизации Тюменской области
Государственное автономное учреждение дополнительного образования
Тюменской области
«Региональный информационно-образовательный центр»

СОГЛАСОВАНО

Директор
Департамента информатизации
Тюменской области

С.И. Логинов

«06» марта 2025 г.



УТВЕРЖДАЮ

Заместитель директора
ГАУ ДО ТО «РИО-Центр»

О.А. Кононенко

«06» марта 2025 г.



УЧЕБНАЯ ПРОГРАММА

Тестировщик веб-приложений (технический уровень)

Трудоемкость программы – 80 академических часов

Форма обучения – очная, очно-заочная с применением дистанционных образовательных технологий, заочная

Режим занятий – 3 академических часа в день

Начальные навыки: базовые навыки работы на персональном компьютере

Программа предназначена для обучения широкой аудитории слушателей навыкам тестирования программного обеспечения.

Цель обучения: овладение слушателями системой практических навыков и углубленных знаний, в сфере тестирования программного обеспечения.

В результате освоения данной программы слушатели приобретут следующие

Знания:

- Основных понятий о теории тестирования.
- Различных методологий разработки ПО и роли тестировщика.
- Техники разработки тестовых сценариев, чек-листов и user кейсов.
- Техники тестирования документации.

Умения:

- Проводить описание тестовых случаев.
- Проводить usability тестирование.
- Проводить тестирование запросов и миграций в БД.
- Проводить проверку соответствия фактических и ожидаемых результатов тестирования ПО.
- Проводить тестирование бэкенд части приложения.
- Выполнять тестирование безопасности.
- Проводить релизы и тестирование релизных сборок приложения с использованием git flow и CI/CD.
- Выбирать и использовать инструменты тестирования для решения конкретной задачи.

Навыки:

- Тестирование документации.
- Тестирование API.
- Тестирование безопасности.
- Тестирование адаптивности.
- Функциональное и нефункциональное тестирование графического интерфейса.

Программой обучения предусмотрены аудиторские занятия и самостоятельная работа слушателей. Под самостоятельной работой подразумевается самостоятельное выполнение слушателями практических заданий под контролем или под руководством преподавателя. Все аудиторские занятия и самостоятельная работа могут вестись как в очной форме, так и с применением дистанционных образовательных технологий.

Итоговая оценка качества освоения программы осуществляется в виде тестирования. По окончании обучения и успешной сдачи тестирования слушателям выдается сертификат о прохождении курса.

УЧЕБНЫЙ ПЛАН

№ п/п	Наименование тем	Всего часов (ак. ч.)	в том числе:		Формы контроля
			Аудиторные занятия (ак. ч.)	Самостоятел ьная работа (ак. ч.)	
1.	Жизненный цикл и методологии разработки ПО	9	6	3	Выполнение практических заданий
2.	Техники тест-дизайна и тест-анализа	6	4	2	
3.	Usability тестирование	9	6	3	
4.	Базы данных	9	6	3	
5.	Тестирование безопасности	9	6	3	
6.	Git и GitHub	9	6	3	
7.	Тестирование API	10	7	3	
8.	Производительность и нагрузочное тестирование	6	4	2	
9.	Работа с CI/CD	10	7	3	
10	Итоговое тестирование	3	3		Выполнение теста
ВСЕГО (ак. ч.)		80	55	25	

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Наименование тем	Всего часов (ак. ч.)	в том числе:		Формы контроля
			Аудиторные занятия (ак. ч.)	Самостоятел ьная работа (ак. ч.)	
1	Жизненный цикл и методологии разработки ПО	9	6	3	Выполнение практических

1.1.	Теория тестирования	3	2	1	ких заданий
1.2.	Виды тестирования	3	2	1	
1.3.	Методологии разработки ПО и роль тестирования	3	2	1	
2	Техники тест-дизайна и тест-анализа	6	4	2	Выполнение практических заданий
2.1.	Тест дизайн	3	2	1	
2.2.	Тест анализ	3	2	1	
3	Usability тестирование	9	6	3	Выполнение практических заданий
3.1.	Критерии юзабилити тестирования	3	2	1	
3.2.	Портреты пользователей	3	2	1	
3.3.	Методы юзабилити тестирования	3	2	1	
4	Базы данных	9	6	3	Выполнение практических заданий
4.1.	Основы баз данных SQL, NoSQL	3	2	1	
4.2.	Основные команды, написание SQL запросов	3	2	1	
4.3.	Тестирование баз данных	3	2	1	
5	Тестирование безопасности	9	6	3	Выполнение практических заданий
5.1.	Основные принципы тестирования безопасности, список OWASP	3	2	1	
5.2.	XSS инъекции,	3	2	1	

	DoS, DDoS				
5.3.	SQL инъекции, обход авторизации	3	2	1	
6	Git и GitHub	9	6	3	Выполнение практических заданий
6.1.	Знакомство с Git, создание репозитория	3	2	1	
6.2.	GitFlow	3	2	1	
6.3.	Основные команды Git	3	2	1	
7	Тестирование API	10	7	3	Выполнение практических заданий
7.1.	RestAPI, Swagger	3	2	1	
7.2.	Автоматизация тестирования Postman	3	2	1	
7.3	Автоматизация тестирования Python с библиотекой Requests	4	3	1	
8	Производительность и нагрузочное тестирование	6	4	2	Выполнение практических заданий
8.1.	Основы нагрузочного тестирования	3	2	1	
8.2.	Инструменты (JMeter, Gatling, Locust)	3	2	1	
9	Работа с CI/CD	10	7	3	
9.1	Введение в CI/CD (GitLab CI, GitHub Actions)	3	2	1	Выполнение практических заданий
9.2	Unit тесты на Python	4	3	1	
9.3	Настройка автоматических тестов в пайплайне	3	2	1	
10	Итоговое тестирование	3	3		Выполнение теста

10.1	Прохождение тестирования	3	3		
Итого часов:		80	55	25	

СОДЕРЖАНИЕ ПРОГРАММЫ

1. Жизненный цикл и методологии разработки ПО

1.1 Теория тестирования

Изучение активностей, входящих в процесс тестирования ПО, разбор основных терминов, применяющиеся при тестировании ПО.

1.2 Виды тестирования

Обзор различных видов тестирования, таких как системное и интеграционное, их суть и области применения.

1.3 Методологии разработки ПО и роль тестирования

Понимание роли тестировщика в процессе разработки ПО, понимание жизненного цикла разработки программного обеспечения и этапов тестирования, ознакомление с различными моделями разработки (например, XP, Lean).

2. Техники тест-дизайна и тест-анализа

2.1 Тест дизайн

Изучение продвинутых техник тест-дизайна, таких как Cause/Effect, Error Guessing, Exhaustive Testing и т.д.

2.2 Тест анализ

Изучение продвинутых техник тест-анализа, таких как Decision tables, State-Transition Testing, Use Case Testing и т.д.

Рефлексия. Обсуждение выполненных заданий. Консультация по индивидуальному проекту.

Разбор сложного материала, защита промежуточных проектов.

3. Usability тестирование

3.1 Критерии юзабилити тестирования

Изучение различных критериев юзабилити тестирования для веб и мобильных приложений

3.2 Портреты пользователей

Научиться составлять и использовать на практике портреты пользователей

3.3 Методы юзабилити тестирования

Изучение продвинутых методов юзабилити тестирования, таких как Think aloud, Observation, Heuristic evaluation и т.д.

Рефлексия. Обсуждение выполненных заданий. Консультация по индивидуальному проекту.

Разбор сложного материала, защита промежуточных проектов.

4. Базы данных

4.1 Основы баз данных SQL, NoSQL

Изучение основных понятий БД, СУБД, не/реляционная база данных, таблица, столбец, атрибут и т.д.

4.2 Основные команды, написание SQL запросов

Научиться писать запросы на добавление данных в БД, выборку данных из одной таблицы, выборку данных из нескольких таблиц и использованием JOIN.

4.3 Тестирование баз данных

Тестирование целостности, оптимизации, миграций в БД, понятия транзакции, индекс, триггер.

Рефлексия. Обсуждение выполненных заданий. Консультация по индивидуальному проекту.

Разбор сложного материала, защита промежуточных проектов.

5. Тестирование безопасности

5.1 Основные принципы тестирования безопасности, список OWASP

Изучение основных понятий и целей тестирования безопасности, ознакомление со списком уязвимостей OWASP.

5.2 XSS инъекции, DoS, DDoS

Научиться находить и предупреждать уязвимости системы к XSS инъекциям, DoS и DDoS атакам.

5.3 SQL инъекции, обход авторизации

Научиться находить и предупреждать уязвимости системы к SQL инъекциям, обходам авторизации.

Рефлексия. Обсуждение выполненных заданий. Консультация по индивидуальному проекту.

Разбор сложного материала, защита промежуточных проектов.

6. Git и GitHub

6.1 Знакомство с Git, создание репозитория

Изучение области применения git, особенности работы с git для тестировщика. Уметь создать свой репозиторий и загружать в него файлы.

6.2 GitFlow

Знать в чем заключается паттерн git-flow, как проводятся релизы, хот-фиксы, разработка в данном паттерне.

6.3 Основные команды Git

Уметь применять команды для создания коммитов, считывания логов, слияния веток, отмены изменений.

Рефлексия. Обсуждение выполненных заданий. Консультация по индивидуальному проекту.

Разбор сложного материала, защита промежуточных проектов.

7. Тестирование API

7.1 RestAPI, Swagger

Изучение основных понятий API, rest/restful/rest-like API. Уметь использовать Swagger для тестирования API.

7.2 Автоматизация тестирования Postman

Уметь тестировать запросы на получение и изменение данных по инструкции с помощью Postman. Уметь использовать Postman для организации коллекции запросов.

7.3 Автоматизация тестирования Python с библиотекой Requests

Уметь написать автоматизированные тесты для тестирования API запросов с помощью языка Python с библиотекой Requests.

Рефлексия. Обсуждение выполненных заданий. Консультация по индивидуальному проекту.

Разбор сложного материала, защита промежуточных проектов.

8. Производительность и нагрузочное

8.1 Основы нагрузочного тестирования

Изучение основ нагрузочного тестирования, основные понятия, виды нагрузочного тестирования.

8.2 Инструменты (JMeter, Gatling, Locust)

Изучение основных инструментов для проведения нагрузочных тестов, таких как JMeter, Gatling, Locust.

Рефлексия. Обсуждение выполненных заданий. Консультация по индивидуальному проекту.

Разбор сложного материала, защита промежуточных проектов.

9. Работа с CI/CD

9.1 Введение в CI/CD (GitLab CI, GitHub Actions)

Знать понятия continuous integration (CI), continuous delivery (CD).
Знать понятия pipeline, stage, job.

9.2 Unit тесты на Python

Изучение процесса написания и структурирования автоматизированных тестов, выполнение тестов и анализ результатов.

9.3 Настройка автоматических тестов в пайплайне

Уметь создать и настроить yaml файл для автоматического запуска CI/CD в GitHub.

Рефлексия. Обсуждение выполненных заданий. Консультация по индивидуальному проекту.

Разбор сложного материала. Защита итоговых проектов.

10. Итоговое тестирование

10.1 Прохождение тестирования.

СПИСОК ЛИТЕРАТУРЫ

1. Баженов А.Н. Тестирование программного обеспечения. М.: Издательско-торговая корпорация "Дашков и К", 2017. - 320 с.
2. Волкова Е.И., Клочков Ю.Б. Тестирование программного обеспечения: методология, методы, практика. СПб.: БХВ-Петербург, 2016. - 352 с.
3. Гальюкова И.В. Основы тестирования программного обеспечения. М.: ДМК Пресс, 2018. - 272 с.
4. Колошин П.Н. Тестирование и отладка программного обеспечения. М.: Бином, Лаборатория знаний, 2016. - 448 с.
5. Лазарев Н.В. Тестирование программного обеспечения. М.: Высшая школа, 2015. - 256 с.
6. Мартченко Н.Е., Ткачев С.В. Тестирование программного обеспечения. СПб.: Питер, 2017. - 352 с.
7. Мирошник В.В. Тестирование программного обеспечения: подход на основе использования моделей. М.: Горячая линия-Телеком, 2016. - 224 с.
8. Поттони М. Как тестировать программное обеспечение? Пер. с англ. СПб.: Питер, 2017. - 320 с.
9. Роберт С.М., Майерс Г.Л. Тестирование объектно-ориентированного программного обеспечения. СПб.: Символ-Плюс, 2016. - 480 с.
10. Таранец В.В. Тестирование программного обеспечения. М.: ИНФРА-М, 2018. - 352 с.
11. Хундт Б. Как тестировать программное обеспечение. Пер. с англ. М.: Рид Групп, 2015. - 416 с.
12. Чепурной Н.В. Тестирование программного обеспечения: учебное пособие. Ростов н/Д: Феникс, 2019. - 192 с.

13. Шилдт Г. Основы тестирования программного обеспечения. СПб.: Символ-Плюс, 2017. - 304 с.

14. Эккерт Дж. Полное руководство по тестированию программного обеспечения. Пер. с англ. М.: DMK Press, 2018. - 512 с.

15. Ярмош С.С., Кошкин В.В. Методы тестирования программного обеспечения. М.: Издательский дом Российского университета дружбы народов, 2016. - 256 с.