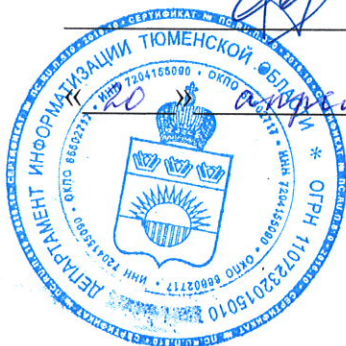


Департамент информатизации Тюменской области
Государственное автономное учреждение дополнительного образования
Тюменской области
«Региональный информационно-образовательный центр»

СОГЛАСОВАНО

Заместитель Губернатора
Тюменской области, директор
Департамента информатизации
Тюменской области


С.И. Логинов



2026 г.

УТВЕРЖДАЮ

Заместитель директора
ГАУ ДО ТО «РИО-Центр»


О.А. Кононенко



2026 г.

УЧЕБНАЯ ПРОГРАММА
Автотестировщик (Automated Quality Assurance, AQA)

Трудоемкость программы – 40 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, базовое знание теории тестирования

Цель обучения: овладение системой практических навыков и углубленных знаний в сфере автоматизации тестирования программного обеспечения с использованием TypeScript.

В курсе рассмотрены особенности автоматизации тестирования веб-приложений:

- работа с системой контроля версий Git Flic для хранения и совместной разработки кода автотестов;

- программирование на TypeScript: типизация, объектно-ориентированный подход, модульная структура;

- модульное тестирование (Unit) с использованием Jest;

- тестирование API: работа с HTTP-клиентами, аутентификация, валидация схем ответов;

- нагрузочное тестирование с использованием инструмента k6;

- сквозное (E2E) тестирование графического интерфейса с использованием Playwright;

- настройка непрерывной интеграции (CI/CD) с Git Flic для автоматического запуска тестов.

Показана практическая работа тестировщика-автоматизатора с распределенными системами контроля версий, TypeScript, современными фреймворками тестирования (Jest, Playwright, k6), инструментами API-тестирования и CI/CD пайплайнами.

В результате освоения данной программы слушатели приобретут следующие

Знания:

- основные принципы работы систем контроля версий и командной работы с Git Flic;

- основы языка TypeScript: типы данных, интерфейсы, классы, модули;

- принципы модульного тестирования и возможности фреймворка Jest;

- архитектуру REST API, методы HTTP, коды ответов, форматы данных;

- подходы к нагрузочному тестированию и работу с инструментом k6;

- возможности и преимущества Playwright для E2E тестирования веб-приложений;

- принципы построения надежных селекторов и паттерн Page Object Model;

- основы CI/CD и настройку пайплайнов в Git Flic.

Умения:

- создавать и настраивать проект автоматизации тестирования на TypeScript с нуля;
- работать с локальными и удаленными Git-репозиториями, создавать ветки, разрешать конфликты, оформлять Pull Request;
- писать модульные тесты для проверки утилитарных функций с использованием Jest;
- разрабатывать API тесты с проверкой статус-кодов, тела ответа и JSON Schema;
- создавать сценарии нагрузочного тестирования в k6 с использованием виртуальных пользователей и пороговых значений;
- разрабатывать E2E тесты с использованием Playwright с применением правильных селекторов (getByRole, getByTestId);
- реализовывать паттерн Page Object Model для структурирования кода тестов;
- настраивать фикстуры, работу с окружениями и генерацию отчетов (Allure, HTML Reporter);
- интегрировать тесты в CI/CD пайплайн для автоматического запуска при изменениях в репозитории.

Программой обучения предусмотрены аудиторные занятия и самостоятельная работа слушателей. Под самостоятельной работой подразумевается самостоятельное выполнение слушателями практических заданий под контролем или под руководством преподавателя. Все аудиторные 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.

УЧЕБНЫЙ ПЛАН

№ п/п	Наименование тем	Всего часов (ак. ч.)	в том числе:		Формы контроля
			Аудиторные занятия (ак. ч.)	Самостоятельная работа (ак. ч.)	
1.	Система контроля версий Git Flic	6	4	2	Выполнение практических заданий
2.	Основы TypeScript	6	4	2	
3.	Unit тестирование	6	4	2	

4.	API и нагрузочное тестирование	9	6	3	
5.	E2E тестирование с Playwright	9	6	3	
6.	Работа с CI/CD. Итоговый проект в Git Flic	3	2	1	
7.	Итоговое тестирование	1	1		Выполнение теста
ВСЕГО (ак. ч.)		40	27	13	

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Наименование тем	Всего часов (ак. ч.)	в том числе:		Формы контроля
			Аудиторные занятия (ак. ч.)	Самостоятельная работа (ак. ч.)	
1.	Система контроля версий Git Flic	6	4	2	Выполнение практических заданий
1.1	Основы Git Flic. Работа с локальным репозиторием	3	2	1	
1.2	Ветвление и удаленные репозитории	3	2	1	
2.	Основы TypeScript	6	4	2	Выполнение практических заданий
2.1	Настройка окружения. Базовые типы и функции	3	2	1	
2.2	Объекты, интерфейсы, классы. Работа с модулями	3	2	1	
3	Unit тестирование	6	4	2	Выполнение практических заданий
3.1	Введение в Jest. Первые тесты. Матчеры	3	2	1	
3.2	Асинхронные тесты. Моки и шпионы. Покрытие кода	3	2	1	
4.	API и нагрузочное	9	6	3	Выполнение

	тестирование				
4.1	Основы HTTP. Первые API тесты	3	2	1	практических заданий
4.2	Структурирование API тестов	3	2	1	
4.3	Введение в нагрузочное тестирование (к6)	3	2	1	
5	E2E тестирование с Playwright	9	6	3	Выполнение практических заданий
5.1	Введение в Playwright. Настройка и первые тесты	3	2	1	
5.2	Продвинутые возможности Playwright. Паттерн Page Object	3	2	1	
5.3	Фикстуры, окружения, отчеты	3	2	1	
6.	Работа с CI/CD. Итоговый проект в Git Flic	3	2	1	Выполнение практических заданий
7.	Итоговое тестирова- ние	1	1		Выполнение теста
ВСЕГО (ак.ч.)		40	27	13	

СОДЕРЖАНИЕ ПРОГРАММЫ

1. Система контроля версий Git Flic

1.1 Основы Git Flic. Работа с локальным репозиторием

Изучение принципов работы систем контроля версий. Освоение базовых команд Git Flic: инициализация репозитория, отслеживание изменений, создание коммитов, просмотр истории. Настройка игнорирования файлов с помощью `.gitignore`. Получение практических навыков работы с локальным репозиторием для хранения кода автотестов.

1.2 Ветвление и удаленные репозитории

Изучение моделей ветвления (Git Flow, GitHub Flow). Освоение команд создания и слияния веток, разрешения конфликтов. Работа с удаленными репозиториями на платформе Git Flic: клонирование, отправка изменений, получение обновлений. Практика создания Pull Request.

2. Основы TypeScript

2.1 Настройка окружения. Базовые типы и функции

Изучение экосистемы Node.js и менеджеров пакетов (npm/yarn/pnpm). Настройка проекта с TypeScript, компиляция, конфигурирование `tsconfig.json`. Освоение базовых типов данных: примитивы, массивы, кортежи, `enum`. Изучение типизации функций: параметры, возвращаемые значения, опциональные параметры.

2.2 Объекты, интерфейсы, классы. Работа с модулями

Изучение объектных типов, интерфейсов и `type aliases`. Освоение работы с классами: свойства, методы, конструкторы, модификаторы доступа. Изучение модульной системы: импорт и экспорт модулей. Практика создания моделей данных и классов-утилит для тестирования.

3. Unit тестирование

3.1 Введение в Jest. Первые тесты. Матчеры

Изучение принципов модульного тестирования. Настройка Jest с поддержкой TypeScript. Освоение структуры тестов: `describe`, `it/test`, `expect`. Изучение основных матчеров для проверки значений, объектов, массивов и исключений.

3.2 Асинхронные тесты. Моки и шпионы. Покрытие кода

Изучение подходов к тестированию асинхронного кода (Promises, `async/await`). Освоение механизмов мокирования: `jest.fn()`, `jest.mock()`, `jest.spyOn()`. Изучение хуков жизненного цикла тестов. Параметризация тестов. Настройка анализа покрытия кода и интерпретация отчетов.

4. API и нагрузочное тестирование

4.1 Основы HTTP. Первые API тесты

Изучение архитектуры REST API: методы запросов, коды ответов, заголовки, форматы данных. Настройка HTTP-клиента Axios с типизацией ответов. Освоение обработки ошибок и конфигурирования базовых параметров запросов. Написание первых тестов для публичных API.

4.2 Структурирование API тестов

Изучение паттернов организации API тестов: Client, Helper, Fixtures. Освоение работы с аутентификацией: Bearer Token, Basic Auth, сохранение cookies. Изучение валидации JSON Schema с использованием библиотек (zod/joi). Практика создания переиспользуемых API клиентов.

4.3 Введение в нагрузочное тестирование (k6)

Изучение целей и видов нагрузочного тестирования: нагрузочное, стрессовое, тестирование стабильности. Освоение инструмента k6: установка, настройка, написание сценариев на TypeScript. Изучение концепций виртуальных пользователей (VUs), длительности теста, пороговых значений (thresholds). Практика запуска нагрузочных тестов и анализа результатов.

5. E2E тестирование с Playwright

5.1 Введение в Playwright. Настройка и первые тесты

Изучение преимуществ Playwright перед другими инструментами автоматизации. Установка и настройка Playwright с поддержкой TypeScript. Освоение конфигурационного файла playwright.config.ts. Написание первых тестов: навигация, взаимодействие с элементами, базовые проверки. Изучение режимов запуска: headless/headed, debug mode.

5.2 Продвинутые возможности Playwright. Паттерн Page Object

Изучение работы со сложными элементами: формы, выпадающие списки, iframe, вкладки, всплывающие окна. Освоение явных и неявных ожиданий. Изучение паттерна Page Object Model (POM) на TypeScript: создание классов страниц, композиция page objects, вынесение локаторов и методов взаимодействия.

5.3 Фикстуры, окружения, отчеты

Изучение механизма фикстур для создания переиспользуемых контекстов тестирования. Освоение работы с переменными окружения для разных сред (dev/stage/prod). Изучение возможностей Playwright для выполнения API-запросов в рамках E2E тестов. Настройка отчетов: HTML

Reporter, JSON, Allure. Освоение Trace Viewer для анализа упавших тестов.

6. Работа с CI/CD. Итоговый проект в Git Flic

Изучение принципов непрерывной интеграции и непрерывной доставки. Освоение настройки пайплайнов в Git Flic: структура workflow, события запуска, jobs, steps. Практика настройки автоматического запуска Unit, API и E2E тестов при изменениях в репозитории. Настройка параллельного запуска, матрицы браузеров и публикации отчетов как артефактов. Подготовка итогового проекта с требованиями к реализации всех изученных видов тестирования.

7. Итоговое тестирование

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.: ДМК 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.