Бауманский учебный центр «Специалист»

Программа курса Модульное тестирование и разработка TDD на C++

Код курса: **СИРАЗ**

Тема	Ак. часов
Модуль 1. Современный TDD и инструментарий C++ 1. **Ведение в TDD** Разберёте основные принципы разработки через тестирование (TDD) и последовательность действий при разработке по этому подходу. 2. **Экосистема тестирования в C++** Познакомитесь с инструментами и средствами, которые используются для модульного тестирования C++-кода. 3. **Первые шаги в TDD** Освоите базовый подход к разработке через тестирование и попробуете применять его при создании C++-кода. 4. **Лабораторная работа** Настроите проект с CMake и Google Test и реализуете небольшую программу строго по TDD.	**4**
Модуль 2. Продвинутое модульное тестирование 1. **Тестовые двойники (Test Doubles)** Разберёте назначение тестовых двойников и их применение при модульном тестировании кода с зависимостями. 2. **Тестирование труднодоступного кода** Рассмотрите подходы к тестированию частей кода, которые сложно проверять изолированно. 3. **Специальные проверки** Познакомитесь со специальными видами проверок, применяемыми при модульном тестировании. 4. **Лабораторная работа** Создадите тестируемый класс-планировщик задач, зависящий от интерфейса времени, напишете модульные тесты и примените моки для проверки логики работы планировщика.	**4**
Модуль 3. TDD и работа с существующим кодом 1. **Стратегия тестирования легаси-кода** Разберёте особенности тестирования существующего кода и подход к выбору стратегии его тестирования. 2. **Рефакторинг для тестируемости** Рассмотрите изменения в структуре существующего кода, которые позволяют сделать его более удобным для тестирования. 3. **TDD как инструмент архитектурного дизайна** Разберёте применение TDD при проектировании и изменении структуры программного кода. 4. **Лабораторная работа** Выполните рефакторинг кода и добавите новые возможности с использованием TDD.	**4**

Модуль 4. Качество тестов и Continuous Testing 1. **ASan и UBSan** Познакомитесь с AddressSanitizer (ASan) и UndefinedBehaviorSanitizer (UBSan) и их применением при проверке C++-кода. 2. **Анти-паттерны и запахи тестов** Разберёте распространённые проблемы в организации тестов и признаки, по которым их можно выявлять. 3. **Покрытие кода и CI** Рассмотрите покрытие кода и его использование в процессе тестирования, а также автоматизацию запуска тестов в CI. 4. **Лабораторная работа** Доработаете код из предыдущих лабораторных работ, примените санитайзеры и создадите конфигурацию GitHub Actions, при которой тесты и отчёт о покрытии формируются при push.	**4**
Аудиторная нагрузка в классе с преподавателем	**16 ак.ч. + 16 ак.ч. самостоятельно**

По окончании обучения на курсе проводится итоговая аттестация. Аттестация проводится в виде теста на последнем занятии или на основании оценок практических работ, выполняемых во время обучения на курсе.

Контакты центра

Для частных лиц
+7 (495) 232-32-16

Для организаций
+7 (495) 780-48-44

E-mail
info@specialist.ru