# Лабораторная работа №11

**Тестирование и отладка приложений**

1. **Цель работы**
   1. Освоить процесс разработки с использованием методологии TDD.
2. **Литература**
   1. Фленов М. Е. Библия C#. — 5-е изд., перераб. и доп. / М. Е. Фленов — СПб.: БХВ-Петербург, 2022. — 464 с.
   2. Куликов, С. C. Тестирование программного обеспечения. Базовый курс : практ. пособие. / С. С. Куликов. — Минск: Четыре четверти, 2020. — 294 с
3. **Подготовка к работе**
   1. Повторить теоретический материал (см. п.2).
   2. Изучить описание лабораторной работы.
4. **Основное оборудование**
   1. Персональный компьютер.
5. **Задание**
   1. Создать форк, а затем клонировать репозиторий по ссылке <https://github.com/ReyRom/MassCultureLibrary>

Проект находится на «Red» этапе разработки: для неразработанного функционала написаны автоматические тесты, (некоторые из них дают ложноположительные результаты)

При написании кода учитывайте, что интерфейс классов Service менять можно, но интерфейс классов Repository - нельзя

* 1. Выбрать **два** из представленных тестов (распределите тесты так, чтобы они не повторялись) изучить связанную с ним логику.
  2. Реализовать минимально необходимый код для прохождения теста (этап «Green»).
  3. Выполнить рефакторинг и оптимизацию написанного кода (этап «Refactor»)
  4. Написать **три** новых «Red» теста для приложения, подробно описав в комментарии к ним новую функциональность или исправления, которые они должны проверять
  5. Отправить запрос на слияние в исходный репозиторий.
  6. Составить отчет по проделанной работе

1. **Порядок выполнения работы**
   1. Повторить теоретический материал п. 3.1;
   2. Выполнить задания п.5.1-5.6
   3. Ответить на контрольные вопросы п.8;
   4. Заполнить отчет п. 7.
2. **Содержание отчета**
   1. Титульный лист;
   2. Цель работы;
   3. Ответы на контрольные вопросы п. 6.3;
   4. Вывод по проделанной работе.
3. **Контрольные вопросы**
   1. Как расшифровывается аббревиатура TTD?
   2. Какие этапы обычно выделяют при разработке через тестирование?
4. **Приложение**

Для реализации подтверждений в тестах рекомендуется использовать библиотеку FluentAssertions (<https://fluentassertions.com/introduction>)

Она предоставляет широкие возможности для проверки результатов выполнения теста.

Примеры подтверждений с использованием FluentAssertions

movie.Should().NotBeNull();

username.Should().Be("admin");

string actual = "ABCDEFGHI";

actual.Should().StartWith("AB").And.EndWith("HI").And.Contain("EF")

.And.HaveLength(9);

dictionary.Should().ContainValue(myClass).Which.SomeProperty

.Should().BeGreaterThan(0);