

Лабораторная работа №11

Тестирование и отладка приложений

1 Цель работы

- 1.1 Освоить процесс разработки с использованием методологии TDD.

2 Литература

2.1 Фленов М. Е. Библия C#. — 5-е изд., перераб. и доп. / М. Е. Фленов — СПб.: БХВ-Петербург, 2022. — 464 с.

2.2 Куликов, С. С. Тестирование программного обеспечения. Базовый курс : практ. пособие. / С. С. Куликов. — Минск: Четыре четверти, 2020. — 294 с

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 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

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

6 Порядок выполнения работы

- 6.1 Повторить теоретический материал п. 3.1;
- 6.2 Выполнить задания п.5.1-5.6
- 6.3 Ответить на контрольные вопросы п.8;
- 6.4 Заполнить отчет п. 7.

7 Содержание отчета

- 7.1** Титульный лист;
- 7.2** Цель работы;
- 7.3** Ответы на контрольные вопросы п. 6.3;
- 7.4** Вывод по проделанной работе.

8 Контрольные вопросы

- 8.1** Как расшифровывается аббревиатура TTD?
- 8.2** Какие этапы обычно выделяют при разработке через тестирование?

9 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().NotNull();
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);
```