Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №2

по дисциплине "Распределенные программные системы"

ТЕМА РАБОТЫ:

Типовая структура Java приложения. Использование системы управления ЖЦ проекта Apache Maven.

Выполнил:

студент гр. ПРИ-120

Парахин К.В.

Приняла:

Проскурина Г.В.

Владимир 2022 г.

Цель работы:

Познакомиться с принципами разработки Java-приложений с использованием автоматизированной системы управления жизненным циклом проекта Apache Maven. Получить практические навыки решения типовых задач при сборке Java проекта.

Выполнение работы:

Создаем директорию окружения для данного проекта: workspace\projects\lab2\myapp

Затем создаем необходимые папки package: main.java.ru.kirill.lab2.myapp

В ней создаем файл Apllication.java, компилируем его и выполняем.

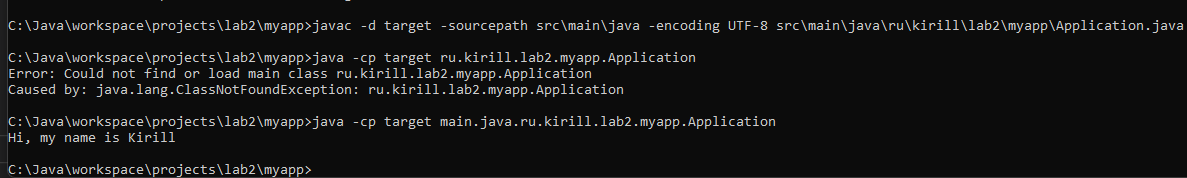


Рисунок 1. Создание каталогов target и компиляция класса

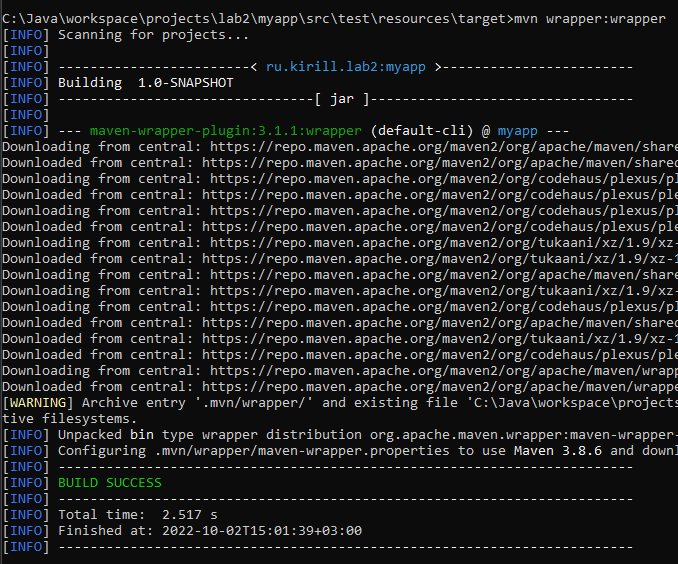


Рисунок 2. Установка Maven Wrapper в проект



Рисунок 3. Очистка проекта с помощью утилиты Maven Wrapper

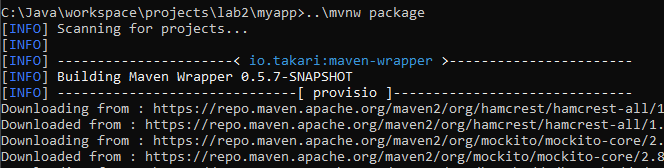


Рисунок 4. Сборка проекта с помощью Maven Wrapper

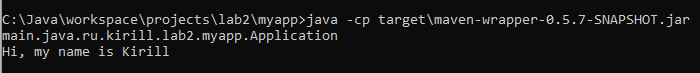


Рисунок 5. Запуск проекта с помощью Maven Wrapper

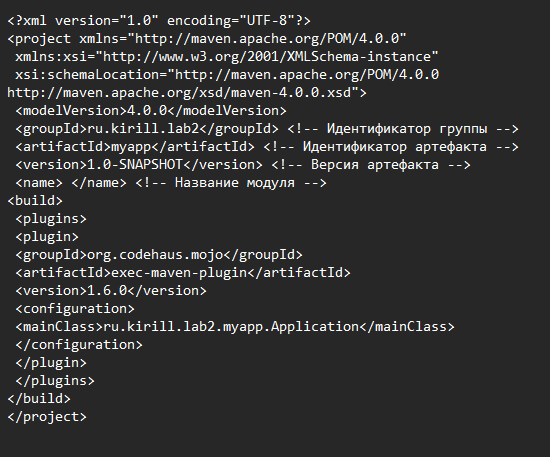


Рисунок 6. Редактирование pom.xml проекта

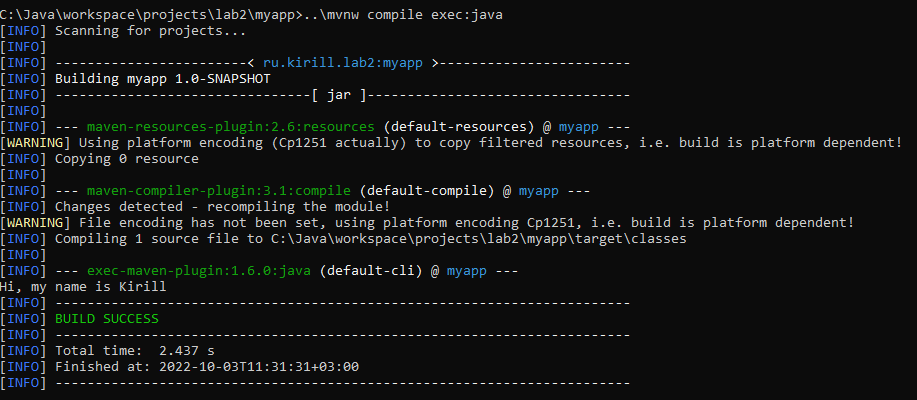


Рисунок 7. Компиляция класса с помощью утилиты exec:java

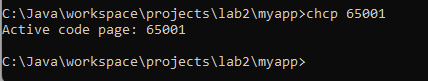


Рисунок 8. Проверка UTF-8 кодировки

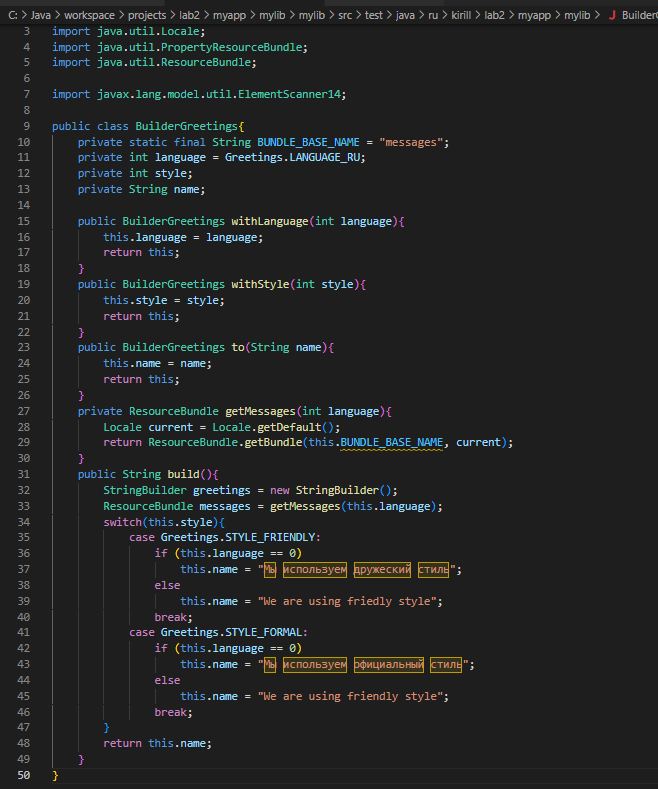


Рисунок 9. Создание дополнительного класса BuilderGreetings

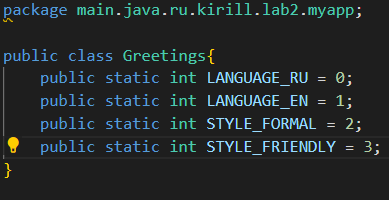


Рисунок 10. Создание дополнительного класса Greetings

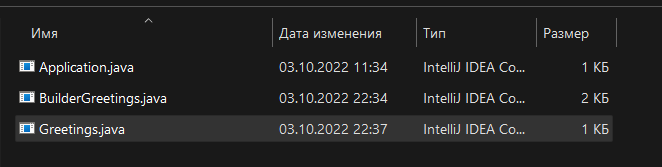


Рисунок 12. Расположение созданных файлов в корневой директории проекта

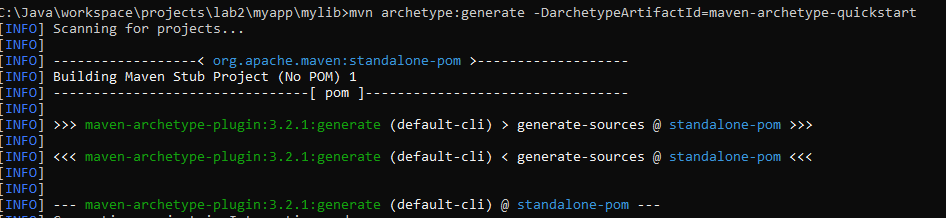


Рисунок 13. Создание проекта типа maven-archetype-quickstart

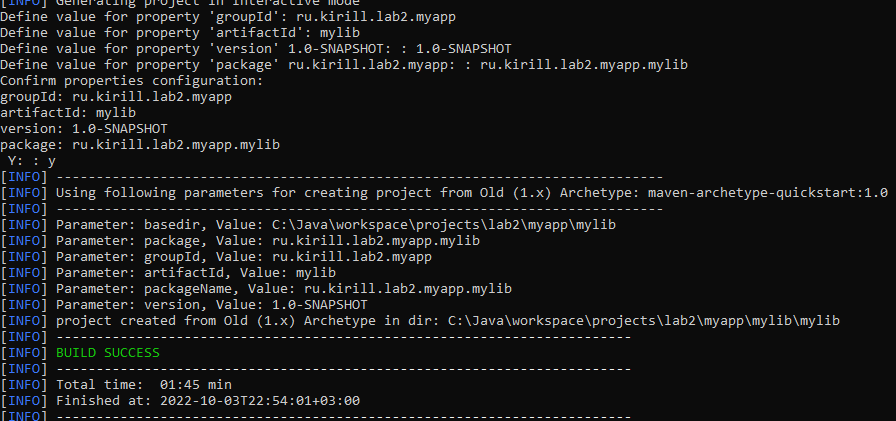


Рисунок 14. Развертывание проекта в новой пустой директории mylib

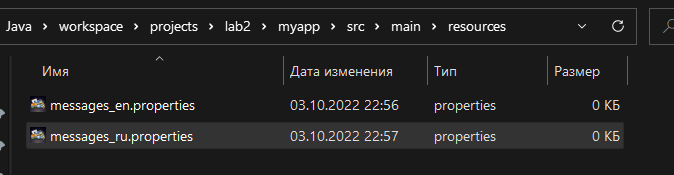


Рисунок 15. Создание словарей свойств в директории resources

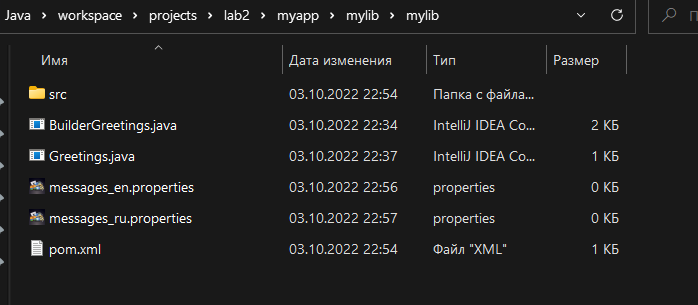


Рисунок 16. Перенос классов и словарей из проекта myapp в проект mylib

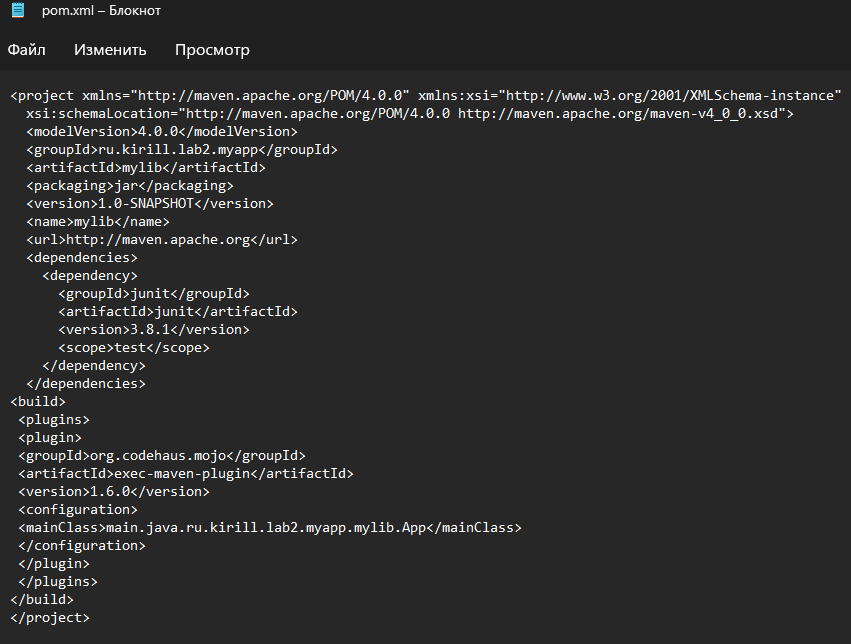


Рисунок 17. Перенос объявление плагина org.codehaus.mojo:native2ascii-maven-plugin

из файла pom.xml проекта myapp в pom.xml нового проекта mylib

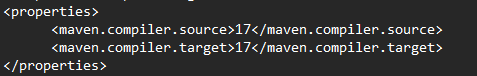


Рисунок 18. Добавление новых свойств в pom.xml проекта mylib

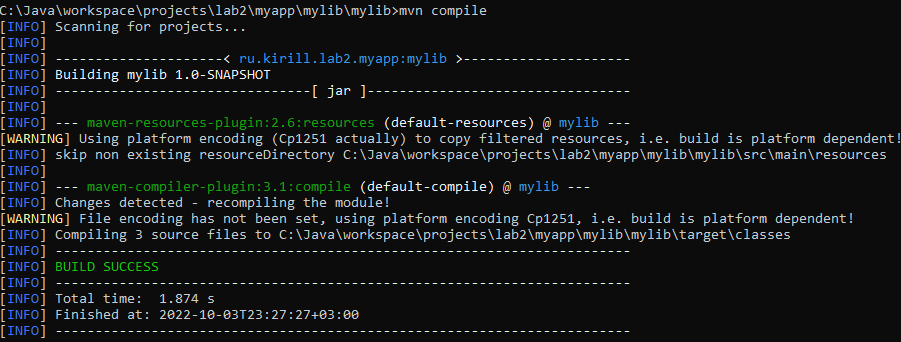


Рисунок 19. Успешная компиляция при использовании maven команды compile

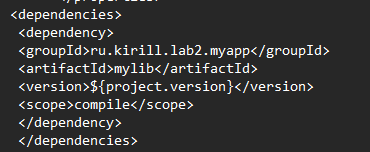


Рисунок 20. Добавление зависимостей в pom.xml проекта myapp

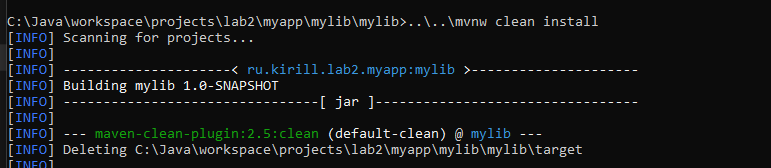


Рисунок 21. Формирование артефактов в каталогах проекта

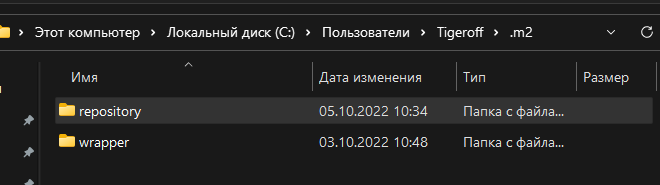


Рисунок 22. Каталоги с созданными артефактами

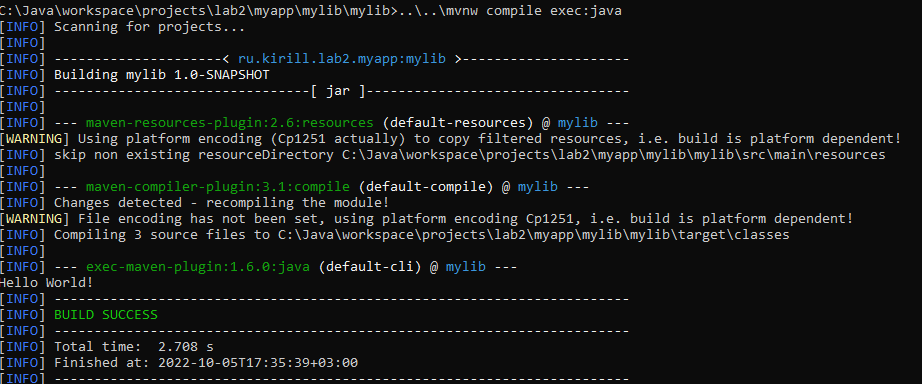


Рисунок 23. Компиляция и запуск проекта

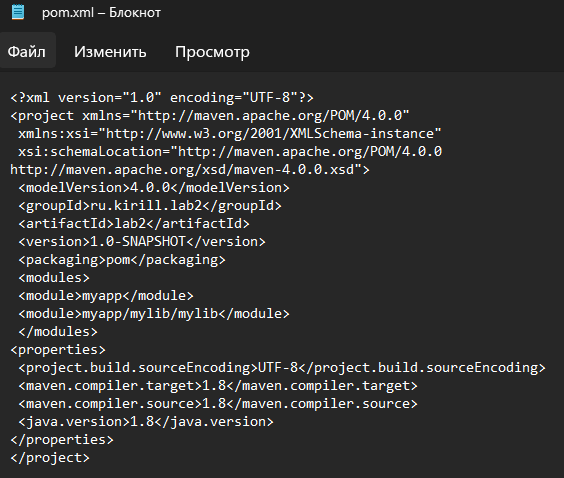
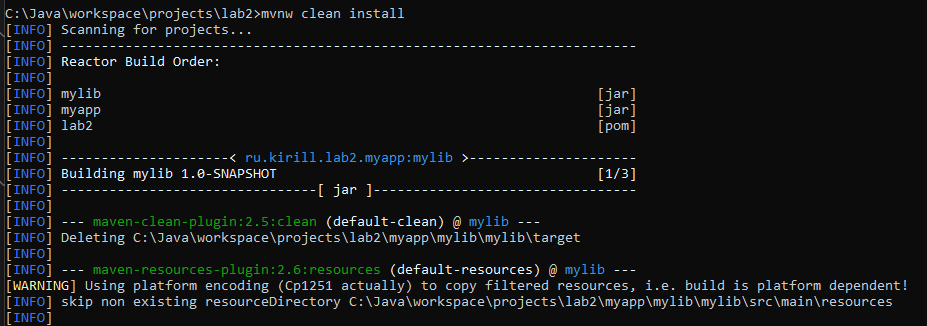


Рисунок 24. Создание нового конфигурационного файла pom.xml в корневой папке проекта



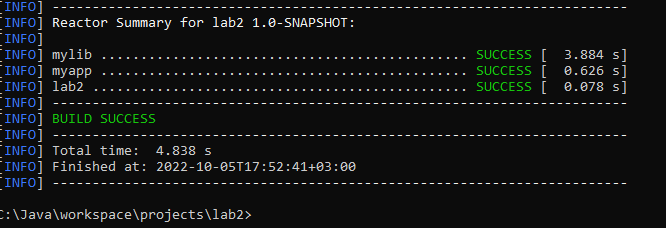


Рисунок 25. Очистка и сборка корневого проекта

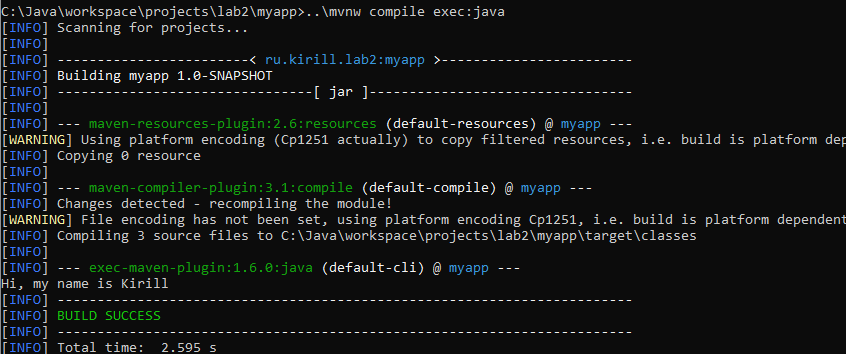


Рисунок 26. Проверка успешной компиляции дочернего проекта myapp

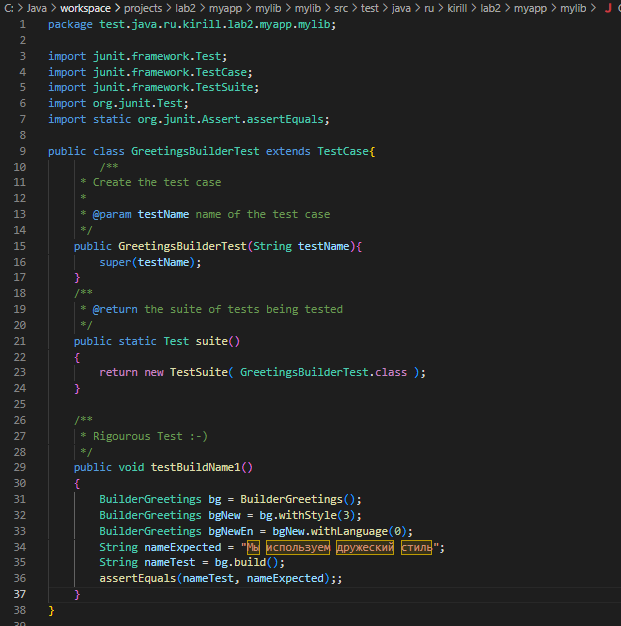


Рисунок 27. Самостоятельное написание второго теста для проекта

(для класса GreetingsBuilder.java)

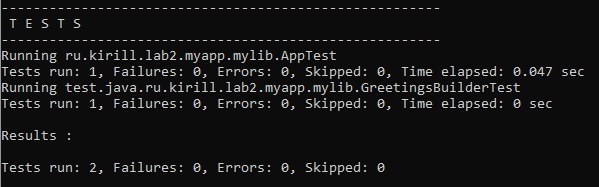


Рисунок 28. Проверка прохождения тестов с помощью утилиты сборки проекта

(mvnw clean install)

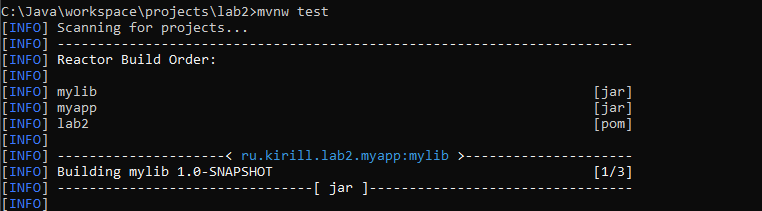


Рисунок 29. Повторение операции с помощью специальной утилиты mvnw test

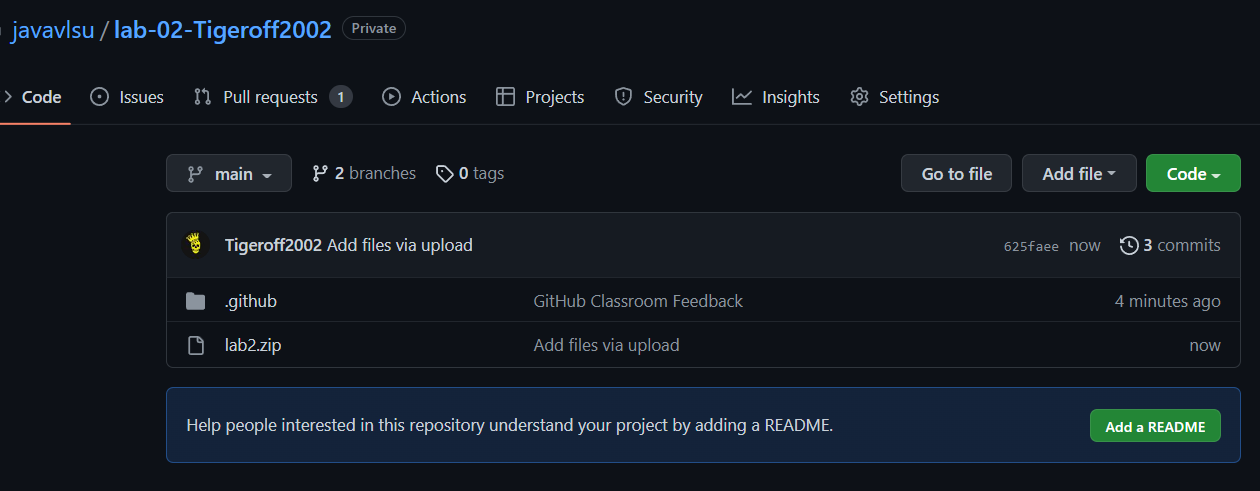


Рисунок 30. Загрузка файлов лабораторной работы на репозиторий github

Вывод

В результате выполнения работы я успешно познакомился с принципами разработки Java-приложений с использованием автоматизированной системы управления жизненным циклом проекта Apache Maven, а также получил практические навыки решения типовых задач при сборке Java проекта.