МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 3
по дисциплине
«Технология разработки программных приложений»

Тема: «Gradle»

Выполнил студент группы: ИКБО-50-23 Павлов Н.С.

Принял Степанов П.В.

Практическая работа выполнена «__»_____2025г. *(подпись студента)*

«Зачтено» «__»_____2025 г. *(подпись руководителя)*

Москва 2025

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

Цель работы: Знакомство с системой сборки Gradle. Возможности gradle. Управление зависимостями.

Персональный вариант: репозиторий: https://github.com/rtu-mirea/trpp-second-9, сущность ru.mirea.entity.Nomenclature

Задания для выполнения:

Для выполнения необходимо клонировать (или форкнуть) git-репозиторий согласно варианту, и выполнить следующие задания:

1. Найти отсутствующую зависимость и указать ее в соответствующем блоке в build.gradle, чтобы проект снова начал собираться
2. В некоторых классах поправить имя пакета
3. Собрать документацию проекта, найти в ней запросы состояния и сущности по идентификатору
4. Собрать jar со всеми зависимостями (так называемый UberJar), после чего запустить приложение. По умолчанию, сервер стартует на порту 8080.
5. Запросить состояние запущенного сервера (GET запрос по адресу http://localhost:8080)
6. Запросить сущность по идентификатору (GET запрос по адресу: http://localhost:8080/сущность/идентификатор) Идентификатором будут 3 последних цифры в серийном номере вашего студенческого билета.
7. В задаче shadowJar добавить к jar-файлу вашу фамилию
8. Выполнить задачу checkstyleMain. Посмотреть сгенерированный отчет. Устранить ошибки оформления кода.

2 ВЫПОЛНЕНИЕ РАБОТЫ

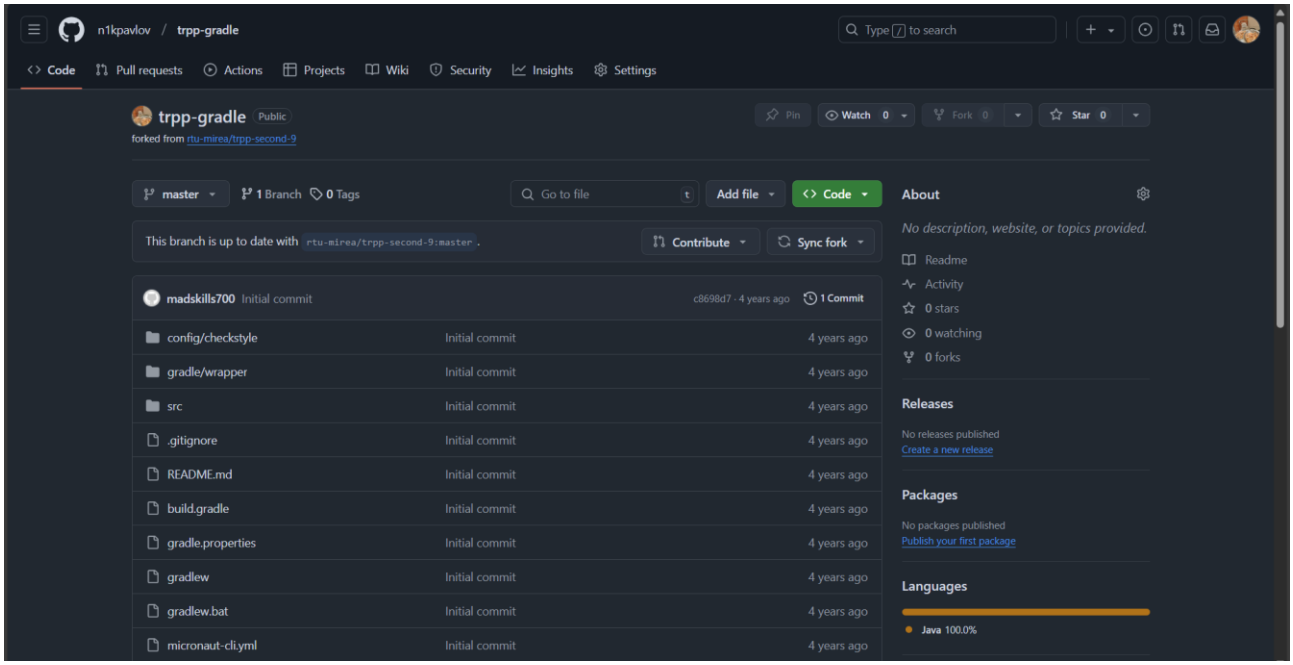

Рисунок 1 – Fork репозитория варианта

Рисунок 2 – Обновленный список зависимостей в build.gradle

Nomenclature.java

```java
package ru.mirea.trpp_second_9.entity;

import com.fasterxml.jackson.annotation.JsonProperty;
import com.opencsv.bean.CsvBindByName;
import lombok.Getter;
import lombok.Setter;
import lombok.ToString;
import java.math.BigDecimal;//new
```

Рисунок 3 – Импорт BigDecimal в Nomenclature.java

HealthController.java

```java
package ru.mirea.trpp_second_9.controllers;

import io.micronaut.http.HttpResponse;
import io.micronaut.http.annotation.Controller;
import io.micronaut.http.annotation.Get;
import ru.mirea.trpp_second_9.entity.HealthResponse;//new
```

Рисунок 4 – Импорт класса HealthResponse в HealthController.java

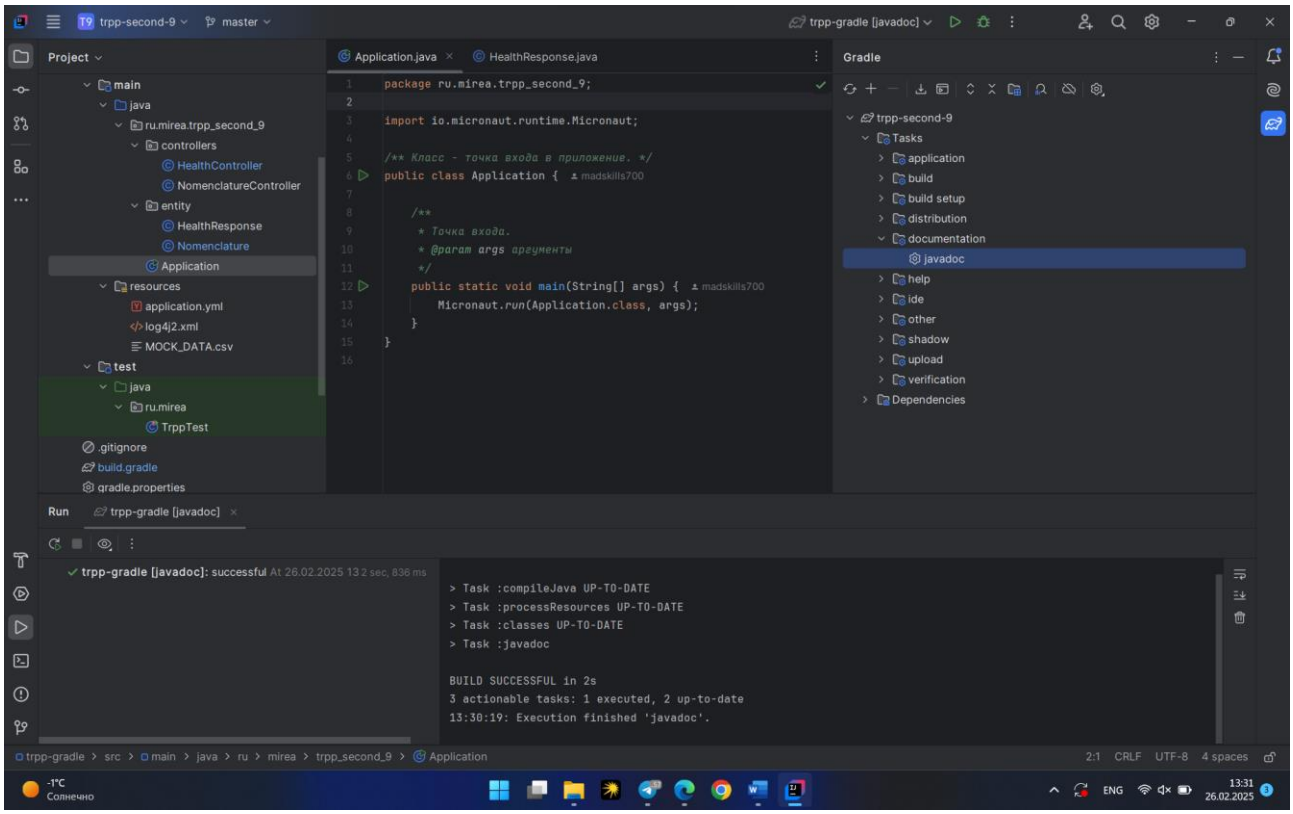

Рисунок 5 – Сборка документации проекта

Method Detail

getNomenclatures

```
@Get public io.micronaut.http.HttpResponse<java.util.List<Nomenclature>> getNomenclatures()
```

Получить список номенклатур.

Returns:

список номенклатур

findById

```
@Get("/{id}") public io.micronaut.http.HttpResponse<Nomenclature> findById(java.lang.Long id)
```

Найти номенклатуру по идентификатору.

Parameters:

`id` - идентификатор номенклатуры

Returns:

Номенклатура, если есть, иначе 404 ошибка

Рисунок 6 – Запросы состояния и сущности по идентификатору

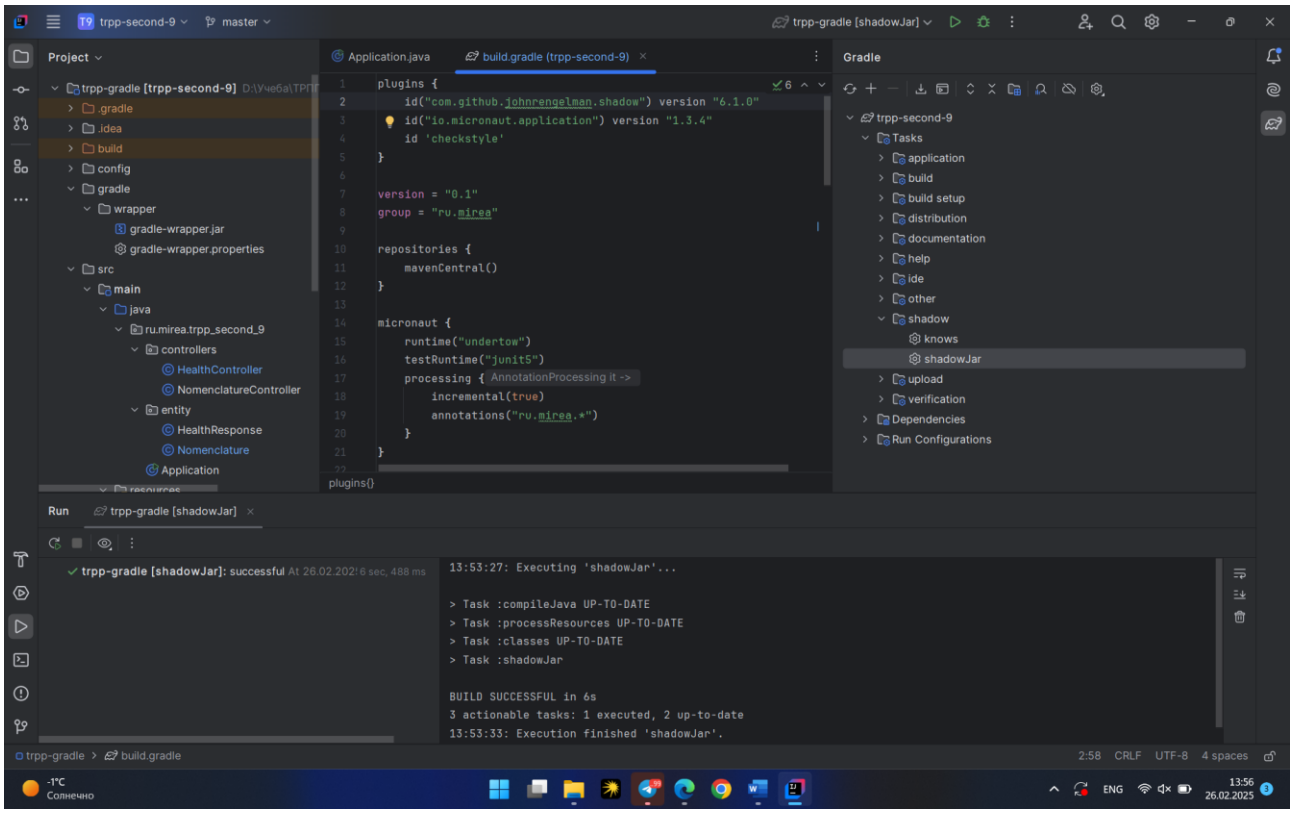

Рисунок 7 – Сборка jar со всеми зависимостями

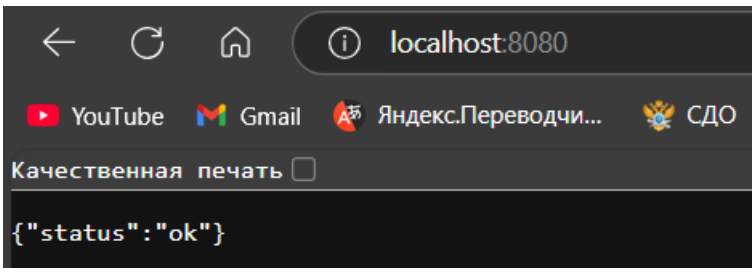

Рисунок 8 – GET запрос по адресу http://localhost:8080

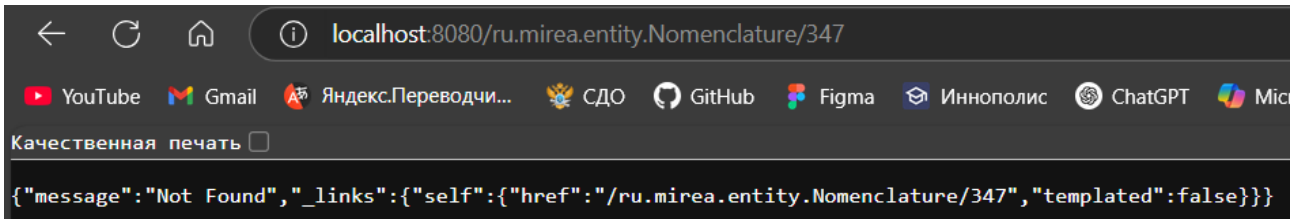

Рисунок 9 – GET запрос по адресу:

localhost:8080/ru.mirea.entity.Nomenclature/347

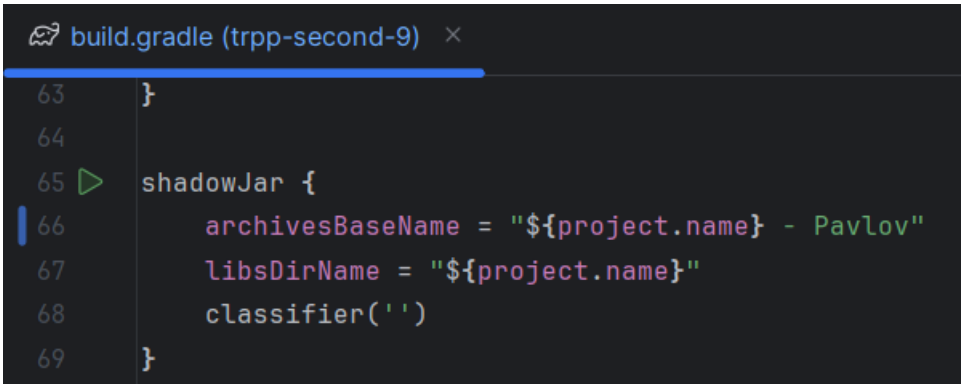

Рисунок 10 – Добавление в задаче shadowJar фамилии

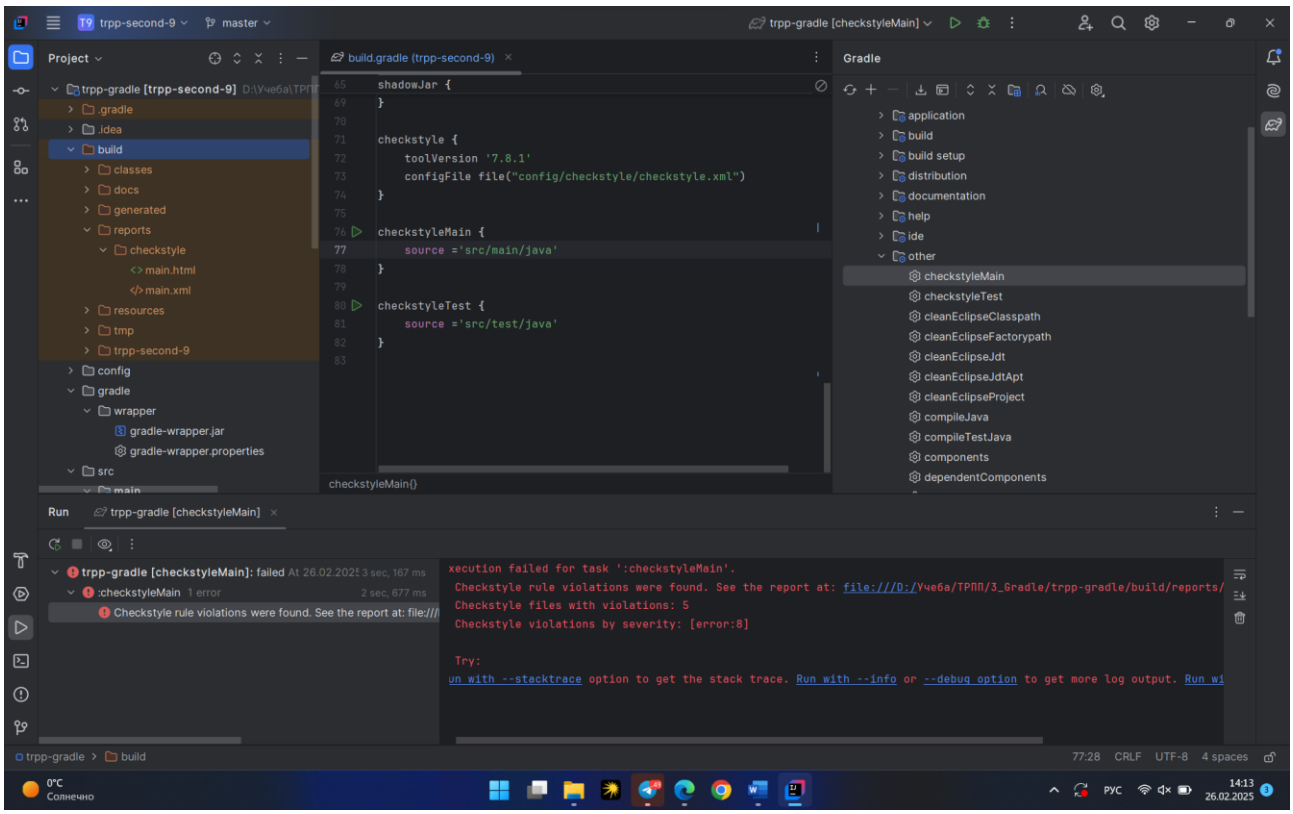

Рисунок 11 – Выполнение задачи checkstyleMain

File D:\Учеба\ТРПП\3_Gradle\trpp-gradle\src\main\java\ru\mirea\trpp_second_9\Application.java	
Error Description	**Line**
Name 'ru.mirea.trpp_second_9' must match pattern '^[a-z]+(\.[a-z][a-z0-9]{1,})*$'.	1

Back to top

File D:\Учеба\ТРПП\3_Gradle\trpp-gradle\src\main\java\ru\mirea\trpp_second_9\controllers\HealthController.java	
Error Description	**Line**
Name 'ru.mirea.trpp_second_9.controllers' must match pattern '^[a-z]+(\.[a-z][a-z0-9]{1,})*$'.	1
';' is not followed by whitespace.	6

Back to top

File D:\Учеба\ТРПП\3_Gradle\trpp-gradle\src\main\java\ru\mirea\trpp_second_9\controllers\NomenclatureController.jav	
Error Description	**Line**
Name 'ru.mirea.trpp_second_9.controllers' must match pattern '^[a-z]+(\.[a-z][a-z0-9]{1,})*$'.	1
Line is longer than 120 characters (found 185).	22

Back to top

File D:\Учеба\ТРПП\3_Gradle\trpp-gradle\src\main\java\ru\mirea\trpp_second_9\entity\HealthResponse.java	
Error Description	**Line**
Name 'ru.mirea.trpp_second_9.entity' must match pattern '^[a-z]+(\.[a-z][a-z0-9]{1,})*$'.	1

Back to top

File D:\Учеба\ТРПП\3_Gradle\trpp-gradle\src\main\java\ru\mirea\trpp_second_9\entity\Nomenclature.java	
Error Description	**Line**
Name 'ru.mirea.trpp_second_9.entity' must match pattern '^[a-z]+(\.[a-z][a-z0-9]{1,})*$'.	1
';' is not followed by whitespace.	8

Back to top

Рисунок 12 – Отчет об ошибках оформления кода

Files	
Name	**Errors**
D:\Учеба\ТРПП\3_Gradle\trpp-gradle\src\main\java\ru\mirea\trppsecond9\Application.java	0
D:\Учеба\ТРПП\3_Gradle\trpp-gradle\src\main\java\ru\mirea\trppsecond9\controllers\HealthController.java	0
D:\Учеба\ТРПП\3_Gradle\trpp-gradle\src\main\java\ru\mirea\trppsecond9\controllers\NomenclatureController.java	0
D:\Учеба\ТРПП\3_Gradle\trpp-gradle\src\main\java\ru\mirea\trppsecond9\entity\HealthResponse.java	0
D:\Учеба\ТРПП\3_Gradle\trpp-gradle\src\main\java\ru\mirea\trppsecond9\entity\Nomenclature.java	0

Рисунок 13 – Отчет после исправления ошибок

3 ВЫВОДЫ

В ходе выполнения практической работы были получены навыки работы с системой сборки Gradle. Изучены функционал и возможности Gradle, исправлены ошибки в исходной сборке, сгенерирована и обработана документация по проекту.

4 ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как указать зависимости проекта?

Зависимости проекта указываются в файле build.gradle (или build.gradle.kts для Kotlin DSL) в секции dependencies. В этом файле вы описываете библиотеки и другие модули, от которых зависит ваш проект.

2. Что такое Gradle?

Gradle — это гибкий и высокопроизводительный инструмент автоматизации сборки, используемый для автоматизации процессов сборки программного обеспечения. Он написан на Groovy (или Kotlin) и использует концепцию "build scripts" (скрипты сборки), которые описывают весь процесс сборки проекта.

3. Что такое Maven?

Maven — это инструмент управления проектами и сборки программного обеспечения, основанный на концепции Project Object Model (POM). Он предоставляет стандартизированный способ управления зависимостями, сборки, тестирования и развертывания проектов Java (и не только). Maven использует XML для описания проекта и его зависимостей.

4. Что такое Javadoc?

Javadoc — это инструмент для автоматической генерации документации по коду Java из комментариев в исходном коде. Комментарии, написанные в специальном формате (документационные комментарии), извлекаются и преобразуются в HTML-документацию. Javadoc облегчает создание понятной и хорошо структурированной документации для вашего кода.

5. Что такое Checkstyle?

Checkstyle — это инструмент статического анализа кода, который проверяет соответствие кода Java заданному стилю кодирования. Он помогает поддерживать единообразие стиля в проекте, что улучшает читаемость и поддерживаемость кода.

6. Что такое UberJar? При помощи какой задачи его собрать?

UberJar (или "fat JAR") — это JAR-файл, который содержит не только код вашего приложения, но и все необходимые зависимости. Это удобно для развертывания, так как вам не нужно вручную управлять зависимостями на целевой машине. Для сборки UberJar в Gradle обычно используется плагин shadow.