



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

«Механизмы разработки программных приложений»

Тема: « Системы сборки »

Выполнил студент группы ИТБО-12-23

Принял

Александр И. В.

Лемченко А. А.

Практическая работа выполнена

«16» 04 2025 г.

Александр И. В.
(подпись студента)

«Зачтено»

«16» 04 2025 г.

Лемченко А. А.
(подпись руководителя)

Москва 2025

СОДЕРЖАНИЕ

1. ПОСТАНОВКА ЗАДАЧИ	3
2. ОСНОВНАЯ ЧАСТЬ	4
3. КОНТРОЛЬНЫЕ ВОПРОСЫ	8
4. ВЫВОДЫ	9

1. ПОСТАНОВКА ЗАДАЧИ

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>)
7. В задаче shadowJar добавить к jar-файлу вашу фамилию
8. Выполнить задачу checkstyleMain. Посмотреть сгенерированный отчет. Устранить ошибки оформления кода.

Номер студента Албахтин И.В. по списку в группе равен $N=3$. Максимальный номер задания в таблице (число вариантов) $M=15$. Тогда номер задания V студента Албахтин И.В. будет равен остатку от деления $V = \text{Ост}((N-1)/M)+1 = \text{Ост}((3-1)/15)+1 = \text{Ост}(2/15)+1 = 2+1=3$

Вариант	Задание
3	репозиторий: https://github.com/rtu-mirea/trpp-second-3 , сущность ru.mirea.entity.Client

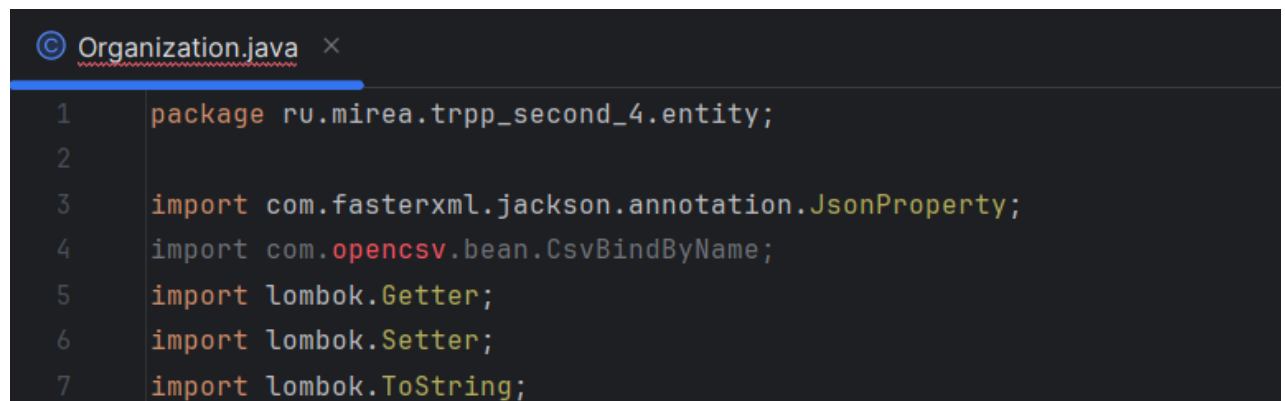
2. ОСНОВНАЯ ЧАСТЬ

Для начала работы склонируем файлы по ссылке в локальный репозиторий

```
ila1b@vex MINGW64 ~/TRPP
$ git clone https://github.com/rtu-mirea/trpp-second-3
Cloning into 'trpp-second-3'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 54 (delta 6), reused 49 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (54/54), 96.32 KiB | 835.00 KiB/s, done.
Resolving deltas: 100% (6/6), done.
```

Рисунок 2.1 — Клонирование репозитория на локальную машину

Как видно на рисунке 2.2, одна из библиотек не загружена



```
1 package ru.mirea.trpp_second_4.entity;
2
3 import com.fasterxml.jackson.annotation.JsonProperty;
4 import com.opencsv.bean.CsvBindByName;
5 import lombok.Getter;
6 import lombok.Setter;
7 import lombok.ToString;
```

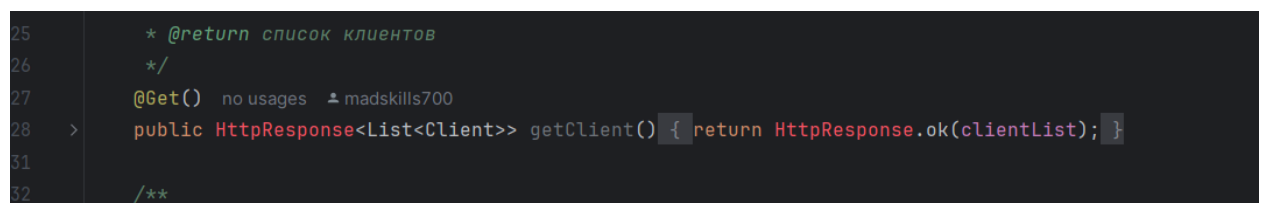
Рисунок 2.2 — Найдена отсутствующая зависимость по импортам

Включим отсутствующую библиотеку в build.gradle

```
implementation 'com.opencsv:opencsv:5.7.1'
```

Рисунок 2.3 — Добавление зависимости в build.gradle

Также заметим отсутствующий импорт — необходимо добавить импорт класса entity.Organization



```
25 * @return список клиентов
26 */
27 @Get() no usages madskills700
28 > public HttpResponse<List<Client>> getClient() { return HttpResponse.ok(clientList); }
31
32 /**
```

Рисунок 2.4 — Найден отсутствующий импорт

Импортируем нужный класс для устранения ошибки

```
2 import io.micronaut.http.HttpResponse;
```

Рисунок 2.5 — Добавление отсутствующего импорта

Убедимся в том, что все ошибки найдены и исправлены, и запустим проект

```
1 package ru.mirea.trpp_second_3.controllers;
2
3 import io.micronaut.http.HttpResponse;
4 import com.opencsv.bean.CsvToBeanBuilder;
5 import io.micronaut.http.annotation.Controller;
6 import io.micronaut.http.annotation.Get;
7 import ru.mirea.trpp_second_3.entity.Client;
8 import java.io.InputStreamReader;
9 import java.util.List;
10 import java.util.Optional;
11
```

Рисунок 2.6 — Отсутствие нарушения в зависимостях и импортах

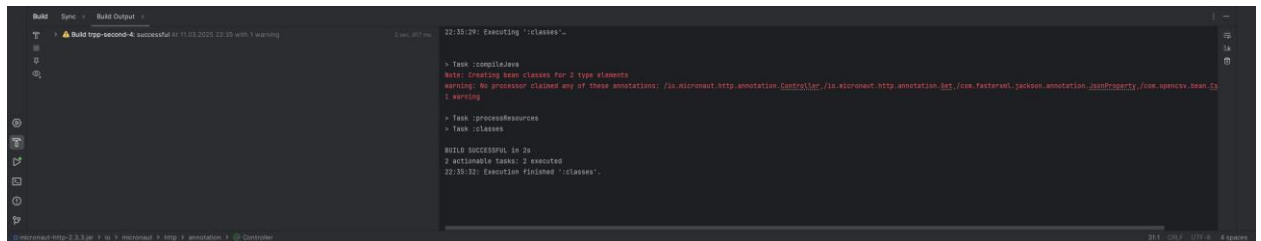


Рисунок 2.7 — Успешный build

После успешного билда проекта сгенерируем его документацию. Документация проекта представлена на рисунке 2.8

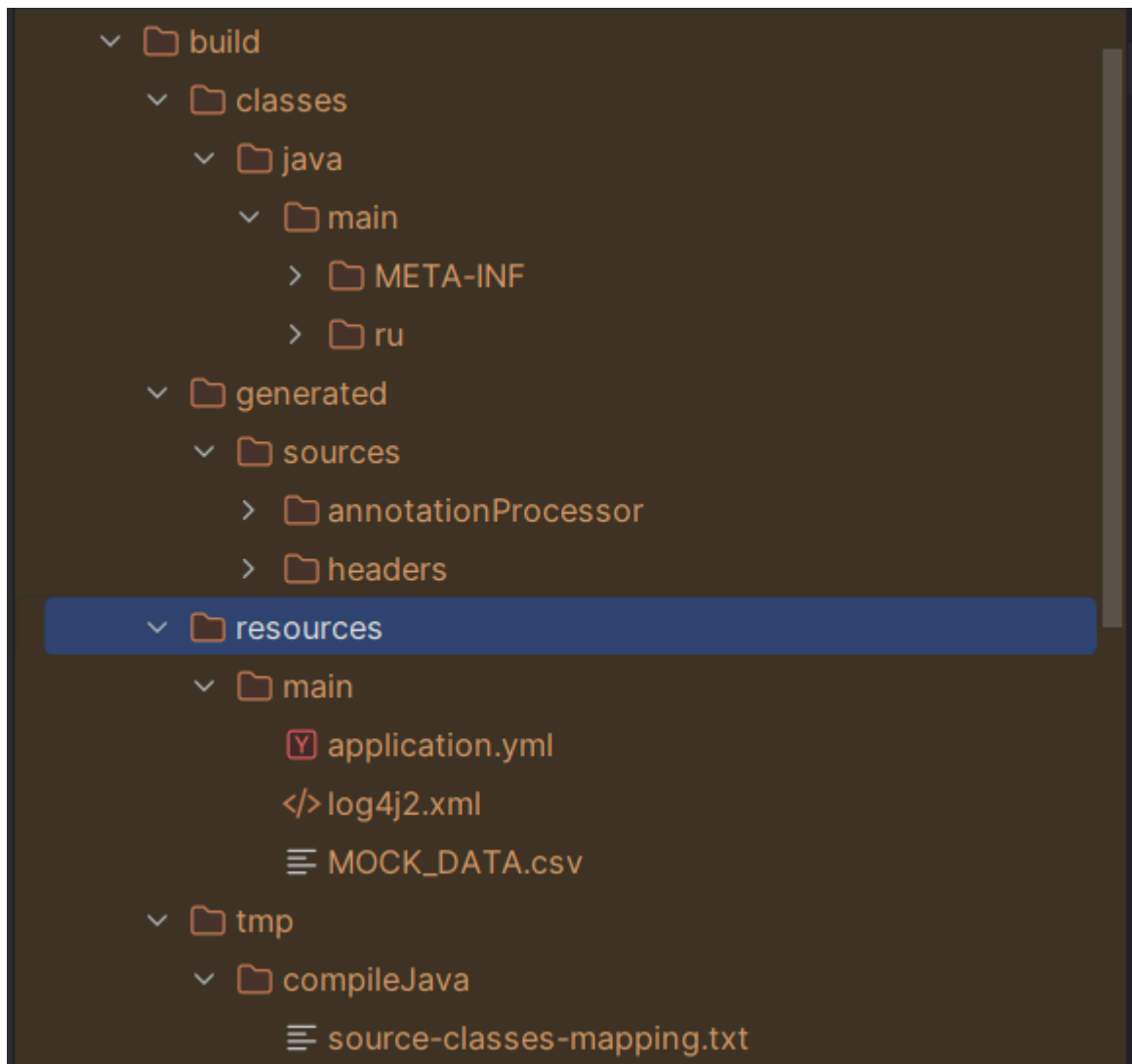


Рисунок 2.8 — Сгенерированная документация

Применим команду `gradlew` для запуска проекта. `./gradlew run` — это команда, которая запускает приложение, используя Gradle. Она предполагает, что в проекте есть задача `run`, которая определена в `build.gradle` файле.

Кроме того, `./gradlew` позволяет запускать Gradle-команды в проекте, даже если нет установленного Gradle на компьютере. Он автоматически загрузит нужную версию Gradle для проекта.

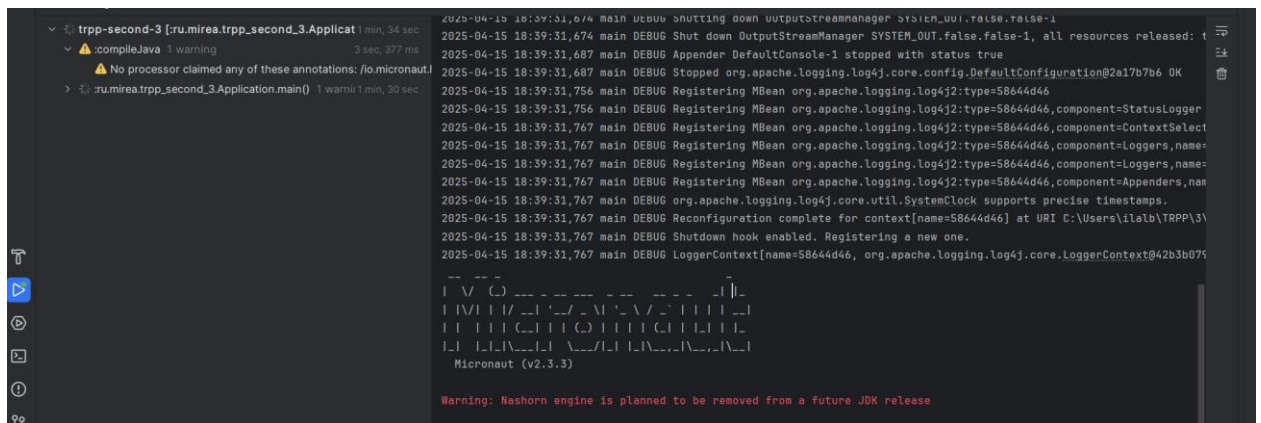


Рисунок 2.9 — Успешный запуск

После успешного запуска проекта выполним прочие задания практической работы:

1)Запросить состояние запущенного сервера (GET запрос по адресу <http://localhost:8080/>)

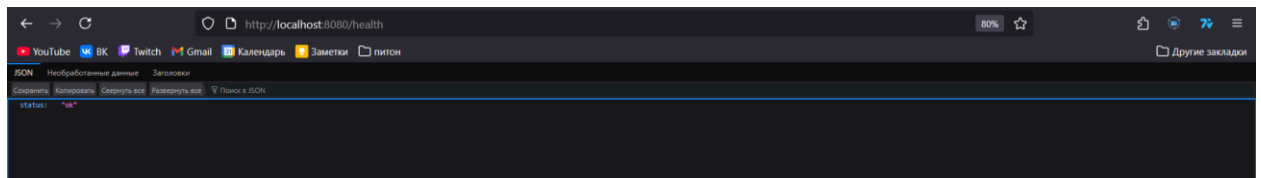


Рисунок 2.10 — Запрос состояния сервера по адресу

2)Запросить сущность по идентификатору (GET запрос по адресу: <http://localhost:8080/сущность/идентификатор>)

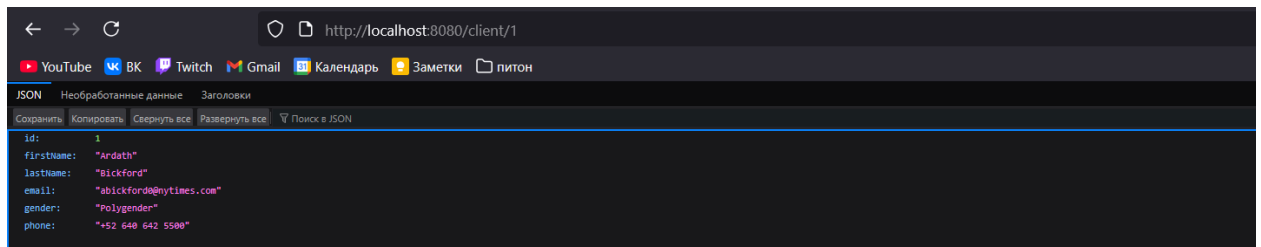


Рисунок 2.11 — Запрос сущности по идентификатору

3)В задаче shadowJar добавить к jar-файлу вашу фамилию:

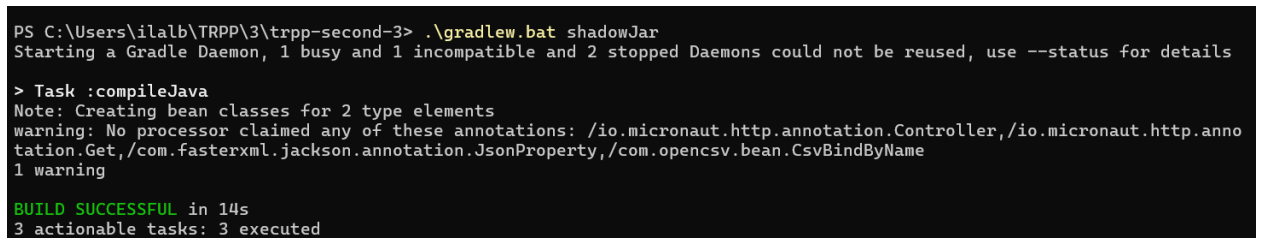


Рисунок 2.12 — Запуск shadowJar

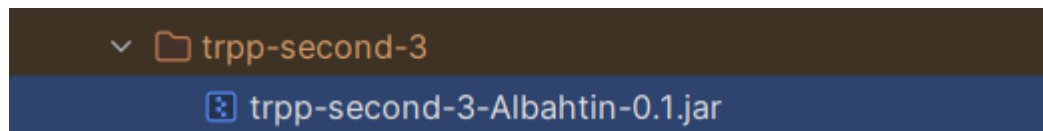


Рисунок 2.13 — Добавление моей фамилии к jar-файлу

Выполнить задачу checkstyleMain. Посмотреть сгенерированный отчет. Устранить ошибки оформления кода.

```
PS C:\Users\ilalb\TRPP\3\trpp-second-3> .\gradlew.bat checkstyleMain

> Task :compileJava
Note: Creating bean classes for 2 type elements
warning: No processor claimed any of these annotations: /io.micronaut.http.annotation.Controller,/io.micronaut.http.annotation.Get,/com.fasterxml.jackson.annotation.JsonProperty,/com.opencsv.bean.CsvBindByName
1 warning

BUILD SUCCESSFUL in 6s
3 actionable tasks: 3 executed
```

CheckStyle Audit	
Designed for use with CheckStyle and Ant .	
Summary	
Files	Errors
5	0
Files	
Name	Errors
C:\Users\ilalb\TRPP\3\trpp-second-3\src\main\java\ru\mirea\trppsecond3\Application.java	0
C:\Users\ilalb\TRPP\3\trpp-second-3\src\main\java\ru\mirea\trppsecond3\controllers\ClientController.java	0
C:\Users\ilalb\TRPP\3\trpp-second-3\src\main\java\ru\mirea\trppsecond3\controllers\HealthController.java	0
C:\Users\ilalb\TRPP\3\trpp-second-3\src\main\java\ru\mirea\trppsecond3\entity\Client.java	0
C:\Users\ilalb\TRPP\3\trpp-second-3\src\main\java\ru\mirea\trppsecond3\entity\HealthResponse.java	0

Рисунок 2.14 — Генерация отчета

Рисунок 2.14 — Просмотр сгенерированного отчета(ошибок нет)

Выполнение практической работы закончено, перейдем к ответам на контрольные вопросы.

3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Чем компиляция отличается от сборки?

Компиляция отличается от сборки тем, что компиляция — это процесс преобразования исходного кода программы в машинный код или интерпретируемый код, тогда как сборка включает в себя этапы компиляции, линковки и другие операции для создания исполняемого файла или другого конечного продукта.

2. Что такое система сборки?

Система сборки — это инструментальный или программное обеспечение, которое автоматизирует процесс компиляции, тестирования и создания программного продукта из его исходных файлов. Она облегчает управление зависимостями, конфигурацией проекта и другими аспектами разработки.

3. Что такое репозиторий?

Репозиторий — это хранилище данных, в котором хранятся файлы исходного кода, библиотеки, зависимости и другие компоненты программного проекта. Репозитории используются для управления версиями и распространения программного обеспечения.

4. Как указать зависимости проекта?

Для указания зависимостей проекта в файле проекта или конфигурационном файле (например, `build.gradle` в Gradle или `pom.xml` в Maven) обычно используются специальные секции или блоки, где перечисляются необходимые библиотеки или модули, а также их версии.

5. Что такое gradle?

Gradle — это инструмент автоматизации сборки проектов, который использует декларативный DSL (язык описания предметной области) на базе Groovy или Kotlin.

6. Что такое maven?

Maven — это инструмент автоматизации сборки и управления проектами, который использует XML-файлы для описания структуры проекта, его зависимостей и сценариев сборки.

4. ВЫВОДЫ

Было произведено знакомство с системой сборки Gradle и возможностями gradle, а также управление зависимостями.