



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практической работе №3

по дисциплине «Технологии разработки программных приложений»

Тема практической работы: «Системы сборки»

Выполнил:

Студент группы ИКБО-20-23

Комисарик М.А.

Проверил:

Доцент кафедры МОСИТ,
кандидат технических наук, доцент
Чернов Е.А.

Москва 2025

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ	3
2 ВЫПОЛНЕНИЕ ЗАДАНИЯ.....	4
3 КОНТРОЛЬНЫЕ ВОПРОСЫ	10
ЗАКЛЮЧЕНИЕ	11

1 ПОСТАНОВКА ЗАДАЧИ

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 ЗАДАНИЯ

1. Отсутствующей зависимостью оказался модуль "com.opencsv". На рисунке 1 представлено необходимое исправление в файле build.gradle.

```
23 dependencies {
24     annotationProcessor 'org.projectlombok:lombok:1.18.18'
25     compileOnly 'org.projectlombok:lombok:1.18.18'
26     implementation("com.opencsv:opencsv:4.0")
27     implementation("io.micronaut:micronaut-runtime")
28     implementation("io.micronaut:micronaut-validation")
29     implementation("io.micronaut:micronaut-http-client")
30     implementation("javax.annotation:javax.annotation-api")
31     implementation("org.apache.logging.log4j:log4j-core:2.12.1")
32     runtimeOnly("org.apache.logging.log4j:log4j-api:2.12.1")
33     runtimeOnly("org.apache.logging.log4j:log4j-slf4j-impl:2.12.1")
34 }
```

Рисунок 1 – Зависимости в файле build.gradle

2. В классе CarController была добавлена внутренняя зависимость на класс Car (Рисунок 2).

```
HealthResponse... ×  gradle.properties ×  CarController.java ×
1 package ru.mirea.trpp_second_13.controllers;
2
3 import ru.mirea.trpp_second_13.entity.Car;
4
5 import com.opencsv.bean.CsvToBeanBuilder;
6 import io.micronaut.http.HttpResponse;
7 import io.micronaut.http.annotation.Controller;
8 import io.micronaut.http.annotation.Get;
9
10 import java.io.InputStreamReader;
11 import java.util.List;
12 import java.util.Optional;
13
14 /** Контроллер для работы с автомобилями. */
15 @Controller("/car")
16 public class CarController {
17
```

Рисунок 2 – Исправленные зависимости класса CarController

Кроме того, были добавлены следующие изменения в файлы сборки:

- в файле build.gradle было изменено неправильное имя входного класса (Рисунок 3);

- в файле `gradle.properties` был добавлен путь к совместимой версии JDK (Рисунок 4).

```
38 application {
39     mainClass.set("ru.mirea.trpp_second_13.Application")
40 }
```

Рисунок 3 – Установка имени входного класса в файле `build.gradle`

```
1 micronautVersion=2.3.3
2 org.gradle.java.home=C:\Program Files\Java\jdk-15
```

Рисунок 4 – Файл `gradle.properties`

3. Запустим генератор документации `javadoc` (Рисунок 5).

```
PS C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13> .\gradlew javadoc

> Task :javadoc
C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13\src\main\java\ru\mirea\trpp_second_13\entity\Car.java:19: w
arning: no comment
    public Long getId()
           ^
1 warning

BUILD SUCCESSFUL in 3s
3 actionable tasks: 1 executed, 2 up-to-date
PS C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13> |
```

Рисунок 5 – Запуск генерации `javadoc` файла

После выполнения команды была создана папка `build`, внутри которой по пути `build/docs/javadoc/` располагается `html` документация проекта. На рисунке 6 представлено отображение главной страницы в браузере.



Рисунок 6 – Файл `javadoc/index.html`

4. Чтобы скомпилировать файл зависимостей запустим задачу `shadowJar` (Рисунок 7)

```
PS C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13> ./gradlew shadowJar

BUILD SUCCESSFUL in 5s
3 actionable tasks: 1 executed, 2 up-to-date
```

Рисунок 7 – Создание `UberJar` файла

Файл был создан согласно конфигурации в папку и файл с названием проекта (Рисунок 8).



Рисунок 8 – Расположение файла UberJar

После этого запустим приложение (Рисунки 9-10).

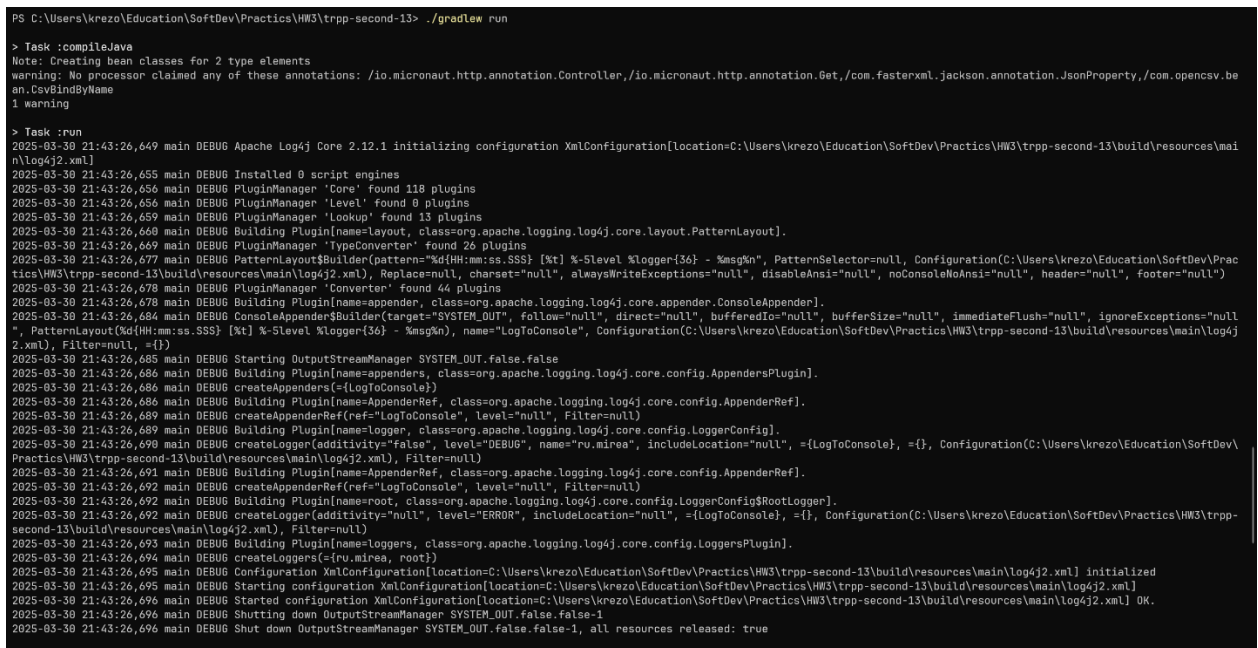


Рисунок 9 – Запуск сервера, часть 1

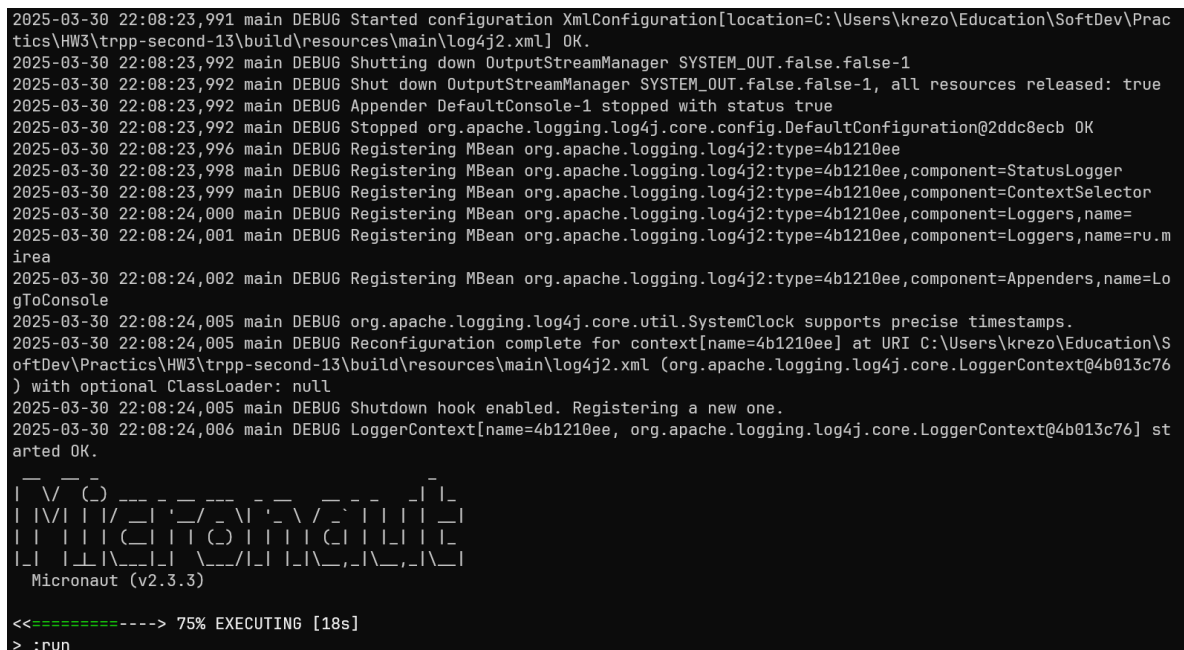


Рисунок 10 – Запуск сервера, часть 2

5. Выполним GET запрос запущенного сервера (Рисунок 11).

```
PS C:\Users\krezo> Invoke-WebRequest http://localhost:8080/

StatusCode      : 200
StatusDescription : OK
Content         : {"status":"ok"}
RawContent      : HTTP/1.1 200 OK
                  Connection: keep-alive
                  Date: Sun, 30 Mar 2025 20:51:50 GMT
                  Content-Type: application/json
                  Content-Length: 15

                  {"status":"ok"}
Headers         : {[Connection, System.String[]], [Date, System.String[]], [Content-Type, System.String[]], [Content-
                  Length, System.String[]]}
Images          : {}
InputFields     : {}
Links           : {}
RawContentLength : 15
RelationLink    : {}
```

Рисунок 11 – Запрос состояния сервера

6. Запросим сущность "car" по идентификатору 310 (Рисунок 12).

```
PS C:\Users\krezo> Invoke-WebRequest http://localhost:8080/car/310

StatusCode      : 200
StatusDescription : OK
Content         : {"id":310,"vendor":"Dodge","model":"Colt","modelYear":"1989","vin":"SCBFU7ZA3FC517211"}
RawContent      : HTTP/1.1 200 OK
                  Connection: keep-alive
                  Date: Sun, 30 Mar 2025 21:42:56 GMT
                  Content-Type: application/json
                  Content-Length: 87

                  {"id":310,"vendor":"Dodge","model":"Colt","modelYear":"1989","vin":"..."
Headers         : {[Connection, System.String[]], [Date, System.String[]], [Content-Type, System.String[]], [Content-
                  Length, System.String[]]}
Images          : {}
InputFields     : {}
Links           : {}
RawContentLength : 87
RelationLink    : {}
```

Рисунок 12 – Запрос сущности по идентификатору

7. В build.gradle изменим конфигурацию shadowJar, добавив к названию фамилию «Комисарик» (Рисунок 13).

```
shadowJar {
    archivesBaseName = "${project.name}_Комисарик"
    libsDirName = "${project.name}"
    classifier('')
}
```

Рисунок 13 – Изменение конфигурации shadowJar

На рисунке 14 представлен UberJar файл, созданный при повторном запуске задачи.

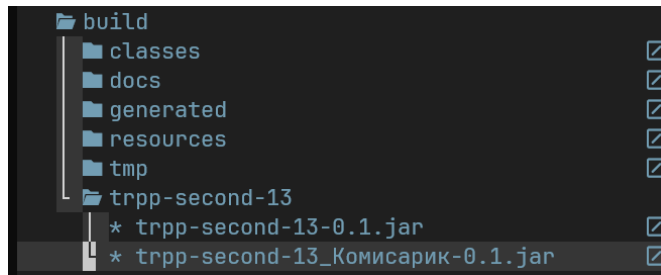


Рисунок 14 – Расположение нового файла UberJar

8. Выполним задачу checkstyleMain (Рисунок 15).

```
PS C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13> ./gradlew checkstyleMain

> Task :checkstyleMain
[ant:checkstyle] [ERROR] C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13\src\main\java\ru\mirea\trpp_second_13\Application.java:1:9: Name 'ru.mirea.trpp_second_13' must match pattern '[a-z](\.[a-z][a-z-9]{1,})?$', [PackageName]
[ant:checkstyle] [ERROR] C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13\src\main\java\ru\mirea\trpp_second_13\controllers\CarController.java:1:9: Name 'ru.mirea.trpp_second_13.controllers' must match pattern '[a-z](\.[a-z][a-z-9]{1,})?$', [PackageName]
[ant:checkstyle] [ERROR] C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13\src\main\java\ru\mirea\trpp_second_13\controllers\CarController.java:5: Wrong order for 'com.opensrv.bean.CsvToBeanBuilder' import. [ImportOrder]
[ant:checkstyle] [ERROR] C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13\src\main\java\ru\mirea\trpp_second_13\controllers\CarController.java:23: Line is longer than 120 characters (found 158). [LineLength]
[ant:checkstyle] [ERROR] C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13\src\main\java\ru\mirea\trpp_second_13\controllers\HealthController.java:1:9: Name 'ru.mirea.trpp_second_13.controllers' must match pattern '[a-z](\.[a-z][a-z-9]{1,})?$', [PackageName]
[ant:checkstyle] [ERROR] C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13\src\main\java\ru\mirea\trpp_second_13\entity\Car.java:1:9: Name 'ru.mirea.trpp_second_13.entity' must match pattern '[a-z](\.[a-z][a-z-9]{1,})?$', [PackageName]
[ant:checkstyle] [ERROR] C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13\src\main\java\ru\mirea\trpp_second_13\entity\Car.java:20:5: '{' at column 5 should be on the previous line. [LeftCurly]
[ant:checkstyle] [ERROR] C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13\src\main\java\ru\mirea\trpp_second_13\entity\HealthResponse.java:1:9: Name 'ru.mirea.trpp_second_13.entity' must match pattern '[a-z](\.[a-z][a-z-9]{1,})?$', [PackageName]

> Task :checkstyleMain FAILED

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':checkstyleMain'.
> Checkstyle rule violations were found. See the report at: file:///C:/Users/krezo/Education/SoftDev/Practics/HW3/trpp-second-13/build/reports/checkstyle/main.html
Checkstyle files with violations: 5
Checkstyle violations by severity: [error:8]

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debug option to get more log output. Run with --scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 3s
3 actionable tasks: 1 executed, 2 up-to-date
```

Рисунок 15 – Запуск задачи checkstyleMain

Задача не была выполнена успешно, так как не соблюдены все требования по стилизации кода. Откроем файл репорта по пути "build/reports/checkstyle/main.html" (Рисунок 16).

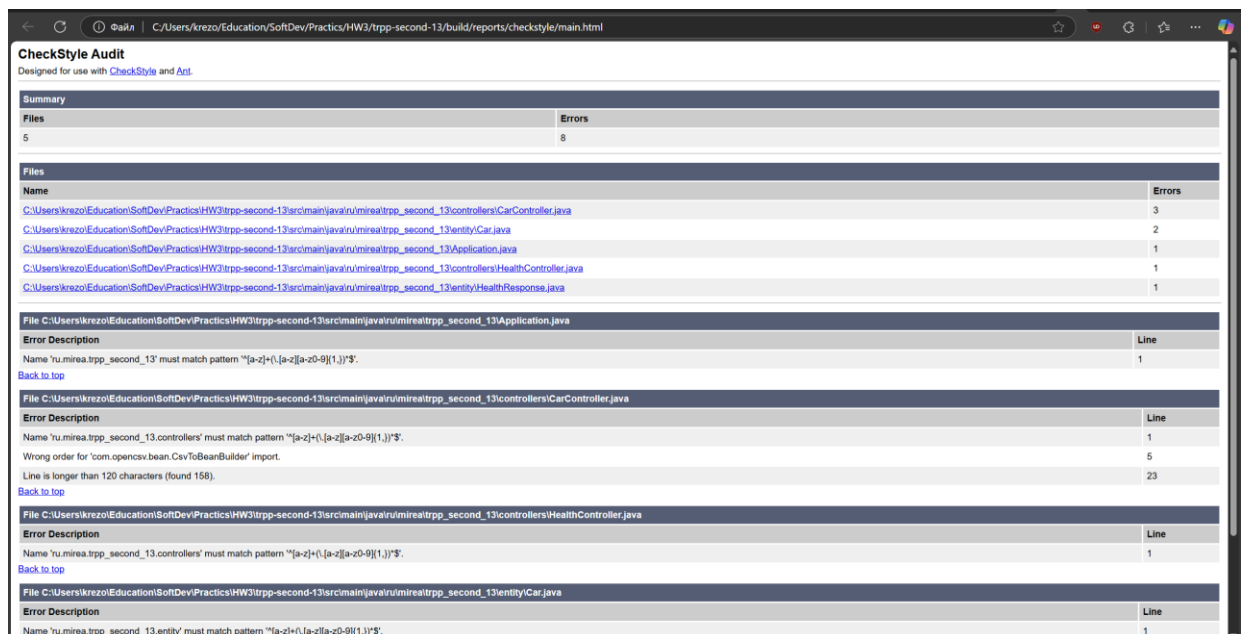


Рисунок 16 – Файл отчета по оформлению кода

Поправим указанные ошибки и запустим задачу снова (Рисунок 17).

```
PS C:\Users\krezo\Education\SoftDev\Practics\HW3\trpp-second-13> ./gradlew shadowJar

> Task :compileJava
Note: Creating bean classes for 2 type elements
warning: A file for type 'ru.mirea.trppsecond13.controllers.$HealthControllerDefinition' already exists on the sourcepath or classpath
warning: A file for type 'ru.mirea.trppsecond13.controllers.$HealthControllerDefinition$$exec1' already exists on the sourcepath or classpath
warning: A file for type 'ru.mirea.trppsecond13.controllers.$HealthControllerDefinitionClass' already exists on the sourcepath or classpath
warning: No processor claimed any of these annotations: /io.micronaut.http.annotation.Controller,/io.micronaut.http.annotation.Get
4 warnings

BUILD SUCCESSFUL in 5s
3 actionable tasks: 2 executed, 1 up-to-date
```

Рисунок 17 – Повторный запуск задачи checkstyleMain

В этот раз ошибок не обнаружено.

3 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Чем компиляция отличается от сборки?

Компиляция — преобразование кода в машинный/байт-код; сборка — создание готового продукта (включая компиляцию, упаковку и т. д.).

2. Что такое система сборки?

Система сборки — инструмент для автоматизации компиляции, тестирования, упаковки проекта (например, Maven, Gradle).

3. Как указать зависимости проекта?

Через конфигурационный файл (например, `pom.xml` в Maven, `build.gradle` в Gradle).

4. Что делает задача `clean`?

`clean` — удаляет сгенерированные файлы (например, `target/`, `build/`).

5. Что делает задача `build`?

`build` — компилирует, тестирует и упаковывает проект.

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

`UberJar` — исполняемый JAR со всеми зависимостями. В gradle собирается задачей `shadowJar`.

ЗАКЛЮЧЕНИЕ

В ходе проделанной работы были изучены основы работы с системой сборки Gradle на примере Java проекта с запуском сервера. Были получены навыки сборки, устранения ошибок при сборке, конфигурации различных параметров сборки.