|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | 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** | |
| **«**Технология разработки программных приложений**»**  **Тема: «Системы сборки»** | |
|  | |
| Выполнил студент группы ИКБО-33-22 | Шило Ю. С. |
| Принял преподаватель | Петрова А.А. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2024

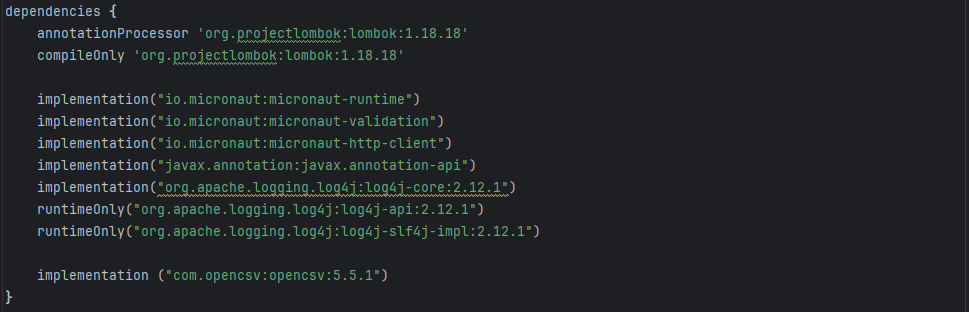
**ЗАДАНИЕ**

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

**Часть 1. Базовые Bash-скриптов.**

*Задание 1.* Найти отсутствующую зависимость и указать ее в соответствующем блоке в build.gradle, чтобы проект снова начал собираться.

Для выполнения задания в файле build.gradle была добавлена зависимость с помощью команды: implementation ("com.opencsv:opencsv:5.5.1"). Результат изменений в файле build.gradle представлен на рисунке 1.1.

Рисунок 1.1 – Результат изменений в файле build.gradle

*Задание 2.* В некоторых классах поправить имя пакета.

Для выполнения данного задания в проекте был изменено название пакета ru.mirea.trpp\_second\_3 было изменено на trpp\_second\_3. Результат изменений представлен на рисунке 1.2.

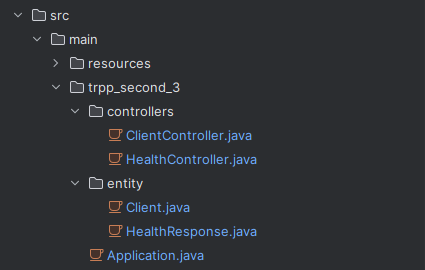


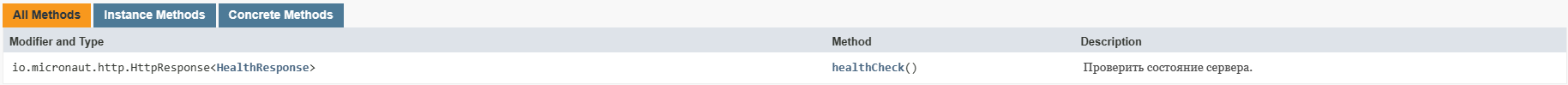
Рисунок 1.2 – Изменение имени пакетов в проекте

*Задание 3.* Собрать документацию проекта, найти в ней запросы состояния и сущности по идентификатору.

Для выполнения данного задания была использована команда „./gradlew javadoc“. После того как была создана документация - запускаем её и в разделе контроллеров находятся методы, в которых используются запросы сущности по идентификатору. На рисунке 1.3 представлены методы контроллера машин.

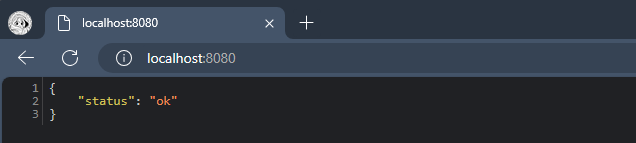
 Рисунок 1.3 – методы контроллера машин в созданной документации

На рисунке 1.4 представлены методы контроллера состояния сервера для запроса состояния.

 Рисунок 1.4 — методы контроллера состояния сервера в созданной документации

*Задание 4.* Собрать jar со всеми зависимостями (так называемый UberJar), после чего запустить приложение. По умолчанию, сервер стартует на порту 8080.

Для выполнения данного задания были использованы команды: ./gradlew shadowJar и ./gradlew run. Первая команда необходима для сбора jar файла со всеми зависимостями (так называемый UberJar). Вторая команда необходима для запуска приложения из собранного билда. На рисунке 1.5 представлен ответ сервера при запросе на «[http://localhost:8080](http://localhost:8080/)», что говорит об успешном запуске.

 Рисунок 1.5 – Ответ сервера при запуске приложения

*Задание 5.* Запросить состояние запущенного сервера (GET запрос по адресу http://localhost:8080)

Для выполнения данного задания был запущен сервер и с помощью браузера FireFox был совершен запрос и получен ответ с состоянием «ok», что говорит об успешном запуске сервера. На рисунке 1.6 представлен ответ сервера.

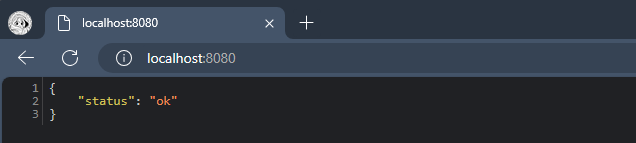


Рисунок 1.6 – Ответ сервера при GET запросе

*Задание 6.* Запросить сущность по идентификатору (GET запрос по адресу: http://localhost:8080/сущность/идентификатор).

Для выполнения данного задания был запущен сервер и с помощью браузера FireFox был совершен запрос «[localhost:8080/client/860](http://localhost:8080/client/860)» для получения сущности по идентификатору. Ответ сервера представлен на рисунке 1.7.

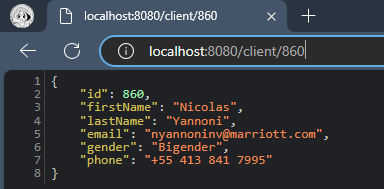
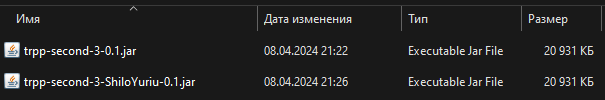


Рисунок 1.7 – Ответ сервера при использовании GET запрос и получение сущности по идентификатору

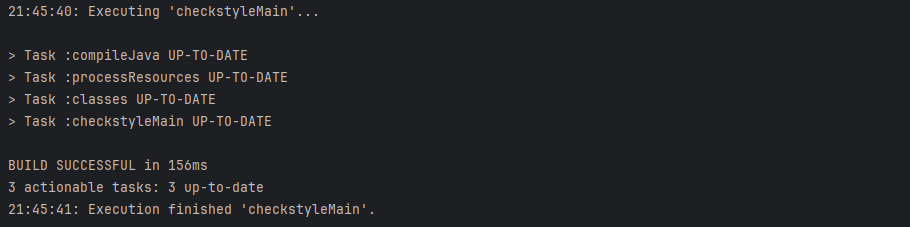
*Задание 7.* В задаче shadowJar добавить к jar-файлу вашу фамилию.

Для выполнения данного задания был изменён файл build.gradle. В строке «archivesBaseName = "${project.name}"» была добавлена моя фамилия. В итоге строке выглядит так: «archivesBaseName = "${project.name} — Tolstopyat"». Результат созданного Jar файла представлен на рисунке 1.8.

Рисунок 1.8 - созданный Jar файл по заданию shadowJar с моей фамилией

*Задание 8.* Выполнить задачу checkstyleMain. Посмотреть сгенерированный отчет. Устранить ошибки оформления кода.

Для выполнения данного задания была использована команда ./gradlew checkstyleMain, которая проверяет код на стилистические ошибки и нарушение правил оформления кода. Если ошибки присутствуют — они выводятся в консоль. Результат выполнения задачи, а также отсутствие стилистических ошибок представлено на рисунке 1.9.

 Рисунок 1.9 - Результат вывода задачи checkstyleMain

**Вопросы**

1. Чем компиляция отличается от сборки?

Ответ: Компиляция — процесс преобразования кода в машинный код или иной, необходимый для выполнения. Сборка — это не только преобразование кода в машинный, но и объединение компонентов, необходимых для работы приложения, по определенным правилам.

2. Что такое система сборки?

### Ответ: Система сборки — программное обеспечение, которое автоматизирует процесс сборки программного проекта.

3. Что такое репозиторий?

Ответ: Репозиторий — централизованное хранилище для хранения, управления файлами, данными или кодом.

4. Как указать зависимости проекта?

Ответ: Для указаний зависимостей проекта в Gradle используется раздел dependencies в buid.gradle.

5. Что такое gradle?

### Ответ: Gradle - это система автоматической сборки, которая позволяет описывать процессы сборки с использованием гибкого языка описания сценариев.

6. Что такое maven?

Ответ: Maven — это инструмент для управления проектами и сборки Java проектов. Он использует файлы pom.xml для описания проекта или зависимостпй.

7. Что такое mavencentral?

Ответ: Централизованный репозиторий, который содержит множество библиотек и зависимостей для проекта.

8. Что делает задача clean?

Ответ: Удаляет сгенерированные файлы и папки (например папку build).

9. Что делает задача build?

Ответ: Компилирует и собирает проект, создавая исполняемый файл для запуска.

10. Что делает задача compileJava?

Ответ: Компилирует файлы с Java кодом в байт-код.

11. Что делает задача run?

Ответ: Запускает собранный проект.

12. Что делает задача test?

Ответ: Запускает тесты Junit проекта.

13. Что такое javadoc?

Ответ: Задача. Генерирует документацию исходного кода в формате HTML.

14. Что такое checkstyle?

Ответ: Задача. Проверяет соответствие Java кода в проекте нормам стиля написания кода.

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

Ответ: исполняемый JAR-файл зависимостей проекта.

16. Что такое micronaut?

Ответ: Фреймворк для создания микросервисов и приложений на JVM.

17. Что такое lombok?

Ответ: Библиотека Java. Нужна для генерации стандартного кода (геттеры, сеттеры и т.д.)

18. Что такое postman?

Ответ: Инструменты тестирования API. Позволяет создавать, отправлять и анализировать HTTP запросы.

19. Что такое аннотация в Java?

Ответ: Вид метаданных, которые добавляется к классам, методам, переменным и другим элементам кода для предоставления доп. информации (может быть инструкция к компилятору или среде, где исполняется код).

**Вывод**

В ходе данной практической работы были получены навыки работы с системой сборки Gradle. Были изучены возможности gradle, управление зависимостями.