**Цель работы**

Знакомство с системой сборки Gradle. Возможности gradle. Управление зависимостями.

**Задание.** Вариант №1

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.

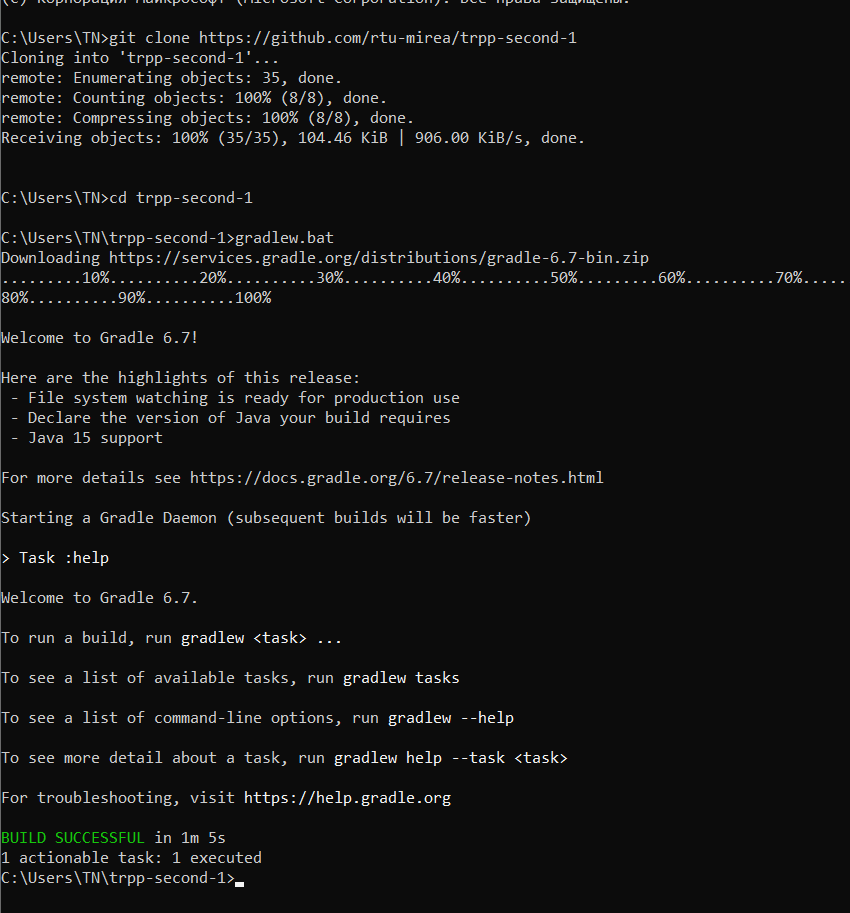
7. В задаче shadowJar добавить к jar-файлу вашу фамилию

8. Выполнить задачу checkstyleMain. Посмотреть сгенерированный

отчет. Устранить ошибки оформления кода

**Ход работы**

Согласно варианту для выполнения необходимо клонировать (или форкнуть) git-репозиторий.

  
Рисунок 1 – Клонирование репозитория

Для того, чтобы собрать проект через Gradle необходимо было устранить все ошибки, которые мешали сборке.

Необходимо было добавить новую зависимость в файл  
build.gradle - библиотеку OpenCSV для работы с .csv файлами. Также нужно  
было сделать импорт внутреннего класса (класс Employee) для корректной  
работы кода. Было исправлено название пакета trpp\_second\_1 на  
trppsecond1, потому что нормы и соглашения наименования пакетов в Java не  
позволяют использовать нижние подчеркивания в названии. Для успешной сборки необходимо было исправить стилевые ошибки в написании кода.



Рисунок 2 – Добавление новой зависимости

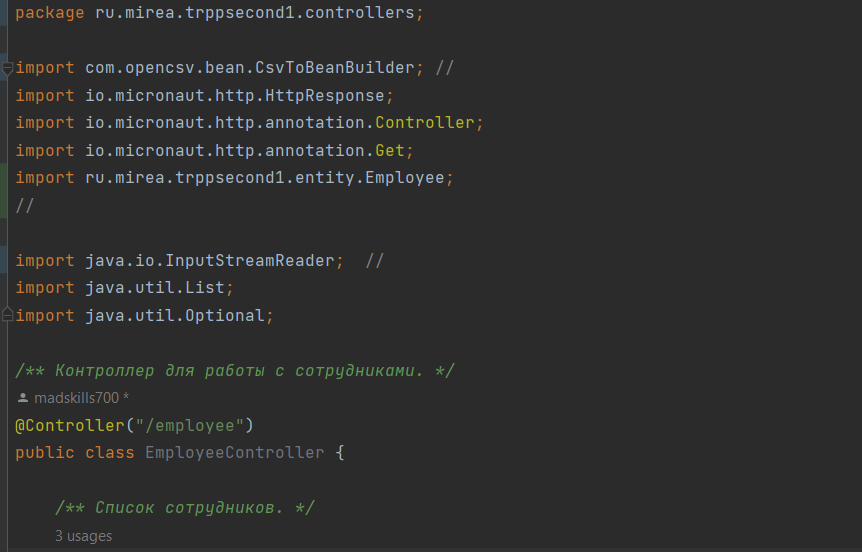


Рисунок 3 – Импорт необходимого класса

Результат успешной сборки проекта представлен на рисунке 7.

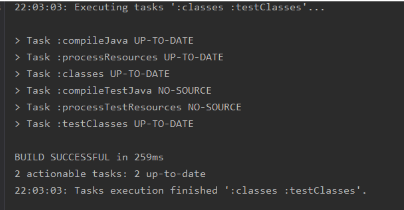


Рисунок 4 – Успешная сборка проекта

Для сборки документации используется задача javadoc, результаты  
сборки документации находятся в папке проекта по пути ./build/docs/javadoc.

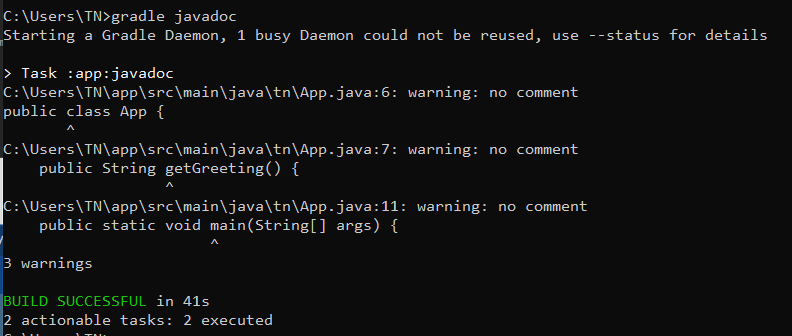


Рисунок 5 – javadoc

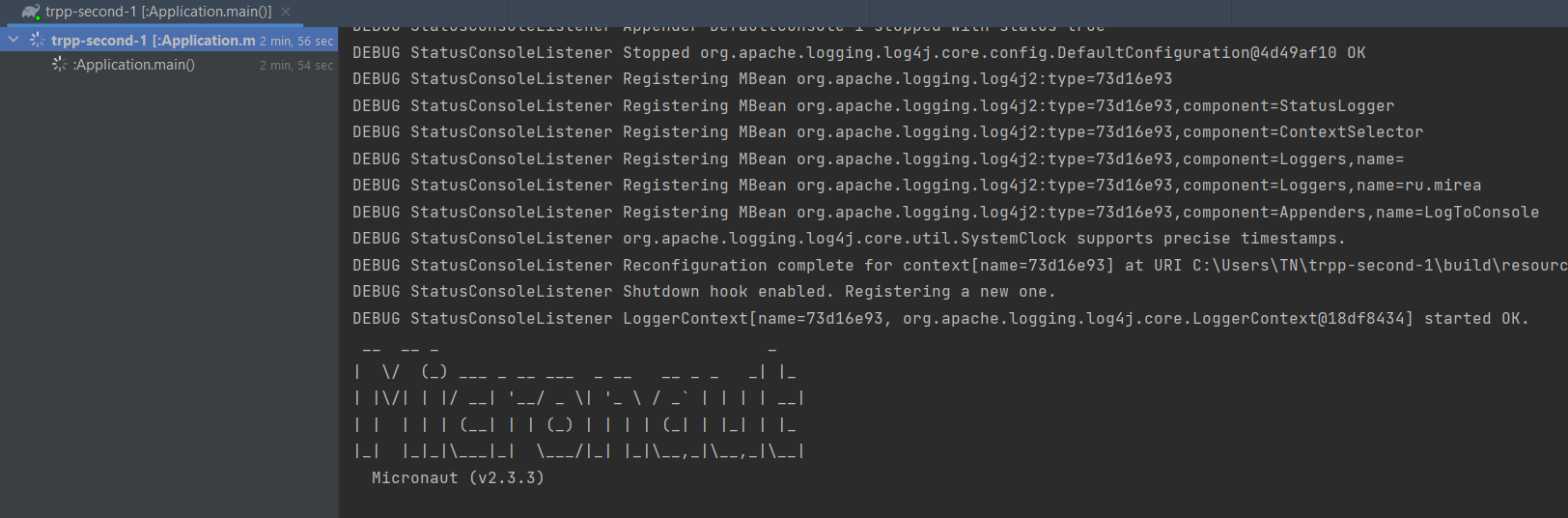


Рисунок 6 – Запуск сервера

Необходимо запросить состояние запущенного сервера, а именно  
выполнить GET запрос по адресу <http://localhost:8080>

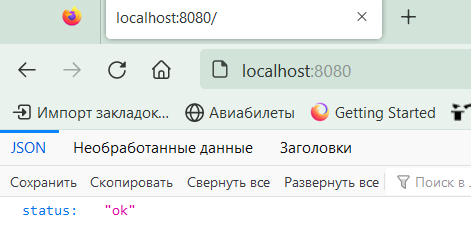


Рисунок 7 – GET запрос

Необходимо запросить сущность по идентификатору, а именно выполнить GET запрос по адресу:

<http://localhost:8080/сущность/идентификатор>.

Идентификатором будут 3 последних цифры в серийном номере вашего студенческого билета.

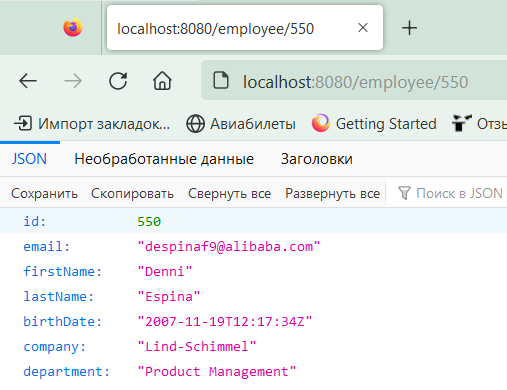


Рисунок 8 – GET запрос по идентификатору

Добавление фамилии к названию jar файла изображено на рисунке 9.

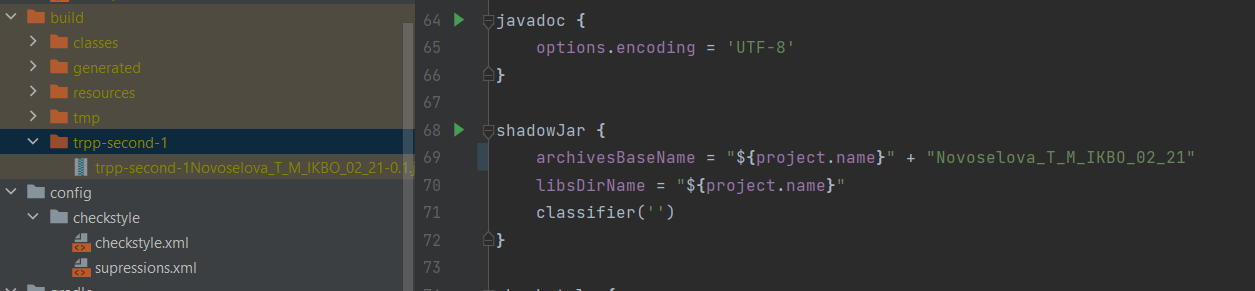


Рисунок 9 – Добавление фамилии

Задача checkstyleMain не выдала ошибок, так как все ошибки стилей  
были исправлены ещё в самом начале, при сборке проекта, рис. 10.

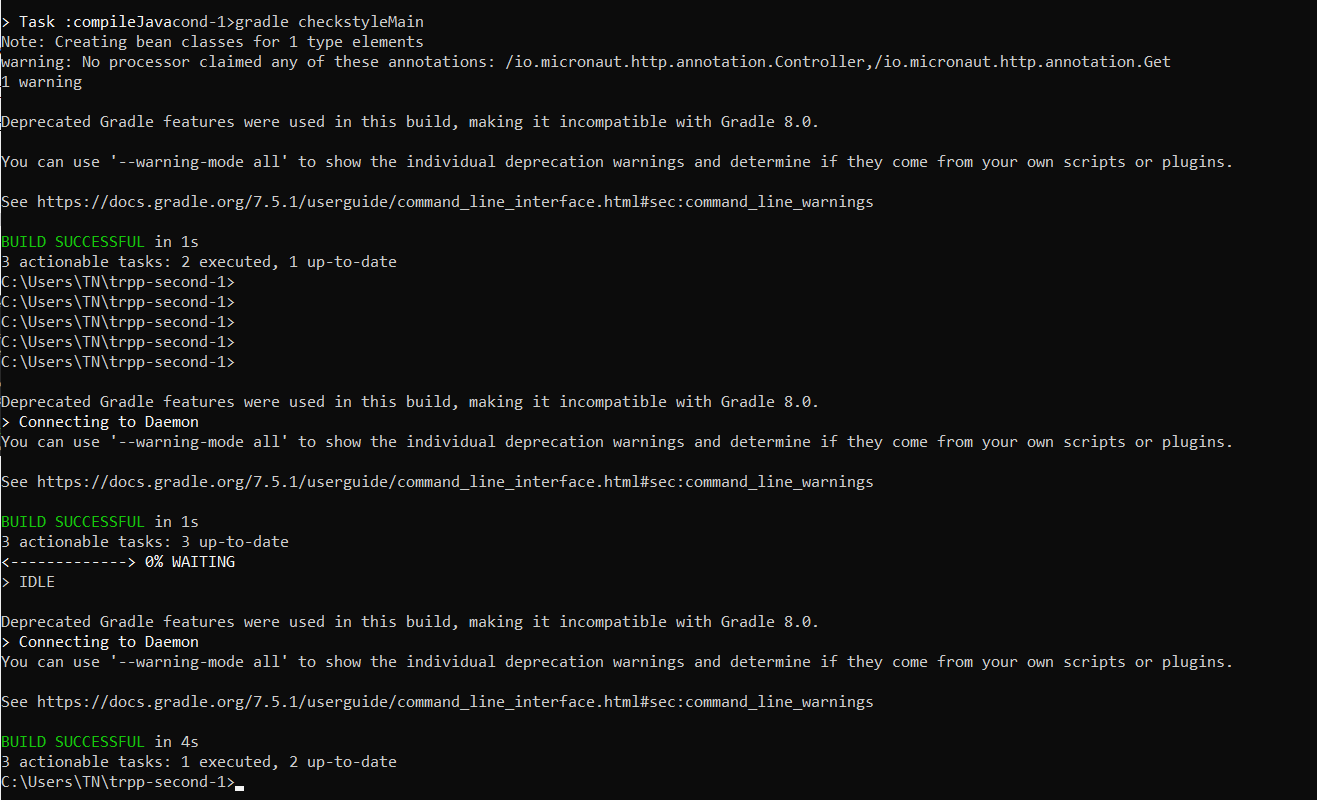


Рисунок 10 – Выполнение задачи checkstyleMain

Получить доступ к результатам сборки документации можно открыв файл index.html, который находится в этой же папке. Просмотр документации представлен на рисунке 11.

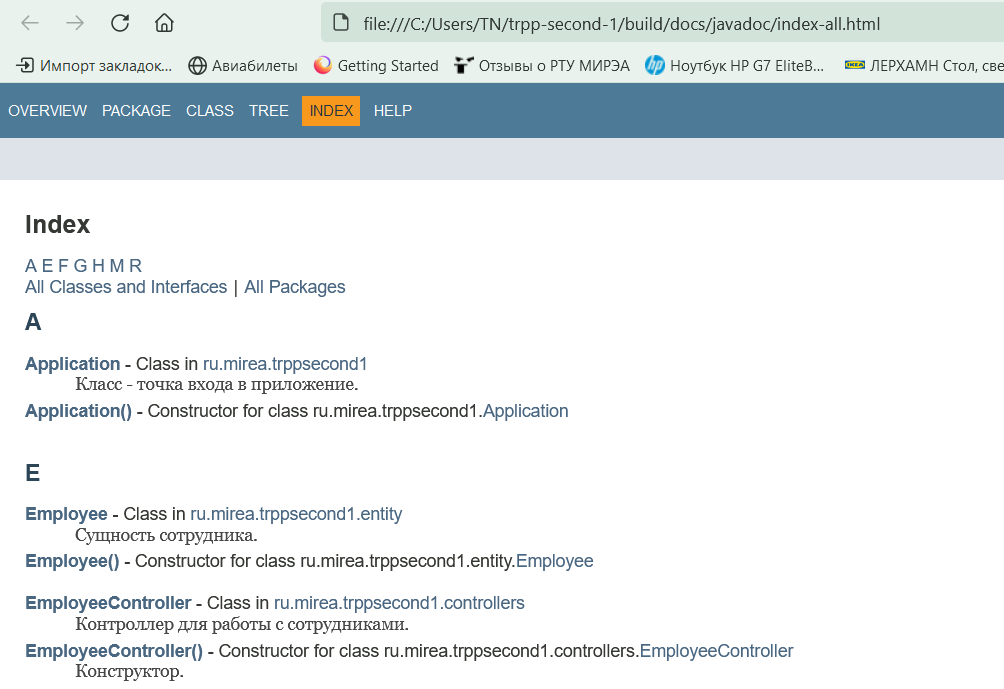


Рисунок 11 – Просмотр документации

**Контрольные вопросы**

1. Чем компиляция отличается от сборки?

Компиляция — это процесс преобразования исходного кода программы на языке высокого уровня в машинный код, который может быть понят и исполнен процессором компьютера.

Сборка — это процесс объединения скомпилированных файлов (или других ресурсов, таких как изображения и шрифты) в единый исполняемый файл или библиотеку. Сборка выполняется после компиляции всех файлов с исходным кодом и может включать этапы оптимизации кода, линковки и упаковки исполняемого файла. Сборка может выполняться несколько раз в течение процесса разработки, чтобы включить изменения в исходный код и обновить исполняемый файл.

5. Что такое gradle?

Gradle — это система автоматической сборки проектов, которая используется в различных областях разработки программного обеспечения, включая Android-приложения, серверные приложения и веб-приложения. Она является современной системой автоматической сборки, которая позволяет создавать сложные проекты с множеством зависимостей и модулей.

6. Что такое Maven?

Maven — это инструмент автоматической сборки проектов, который используется в различных областях разработки программного обеспечения, включая Java-приложения и серверные приложения. Он является популярным инструментом для управления зависимостями, сборки проектов и автоматизации процесса разработки.

7. Что такое mavencentral? Maven Central — это крупнейший репозиторий для Java-библиотек и плагинов, который используется в системе автоматической сборки Maven. Этот репозиторий хранит множество библиотек и плагинов, которые могут быть использованы в проектах Java.

9. Что делает задача build?

Задача " build" в системе автоматической сборки, такой как Maven или Gradle, выполняет сборку приложения целиком.

18. Что такое Postman?

Postman — это инструмент для тестирования API и создания запросов к веб-сервисам.