|  |
| --- |
|  |
| 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 № 2** |
| **по дисциплине** |
| **«Технология разработки программных приложений»**  **Тема: «Системы сборки»** |

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-02-21 | Лосев А.С. |
| Принял | Петренко А.А. |

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

Москва 2023

**Оглавление**

[Цель работы 3](#_Toc130012513)

[Задание для выполнения 3](#_Toc130012514)

[Персональный вариант 3](#_Toc130012515)

[Ход выполнения работы 4](#_Toc130012516)

[Контрольные вопросы 9](#_Toc130012517)

[Выводы 10](#_Toc130012518)

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

Задание 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***.**

1. В задаче shadowJar добавить к jar-файлу вашу фамилию
2. Выполнить задачу checkstyleMain. Посмотреть сгенерированный отчет. Устранить ошибки оформления кода.

Персональный вариант: №9 репозиторий: https://github.com/rtu-mirea/trpp-second-9, сущность ru.mirea.entity.Nomenclature

## Ход выполнения работы

1. Найти отсутствующую зависимость (рисунок 1) и указать ее в соответствующем блоке в build.gradle (рисунок 2), чтобы проект снова начал собираться, в некоторых классах поправить имя пакета (рисунки 3 и 4):

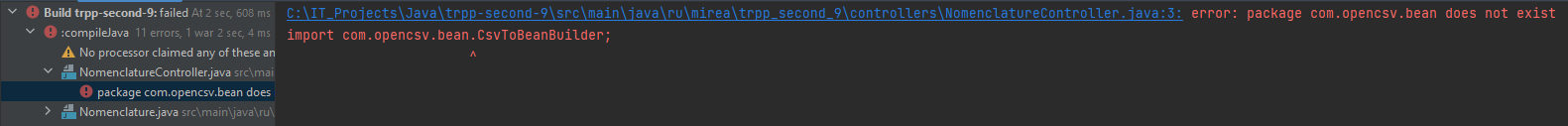


Рисунок 1 – Ошибки при попытке сборки проекта

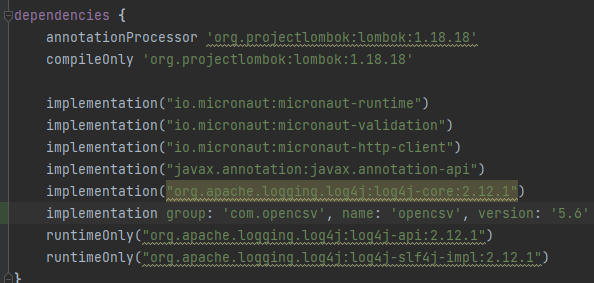


Рисунок 2 – Добавление соответствующей зависимости

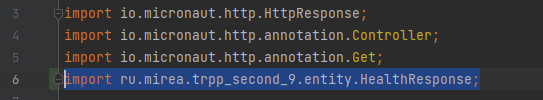


Рисунок 3 – Исправление ошибок

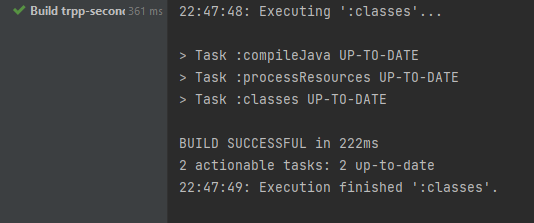


Рисунок 4 – Успешная сборка проекта

1. Собрать документацию проекта (рисунок 5), найти в ней запросы состояния и сущности по идентификатору (рисунок 6):

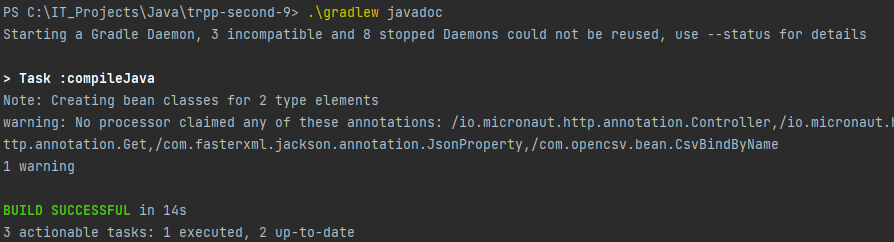


Рисунок 5 – Выполнение сборки документации

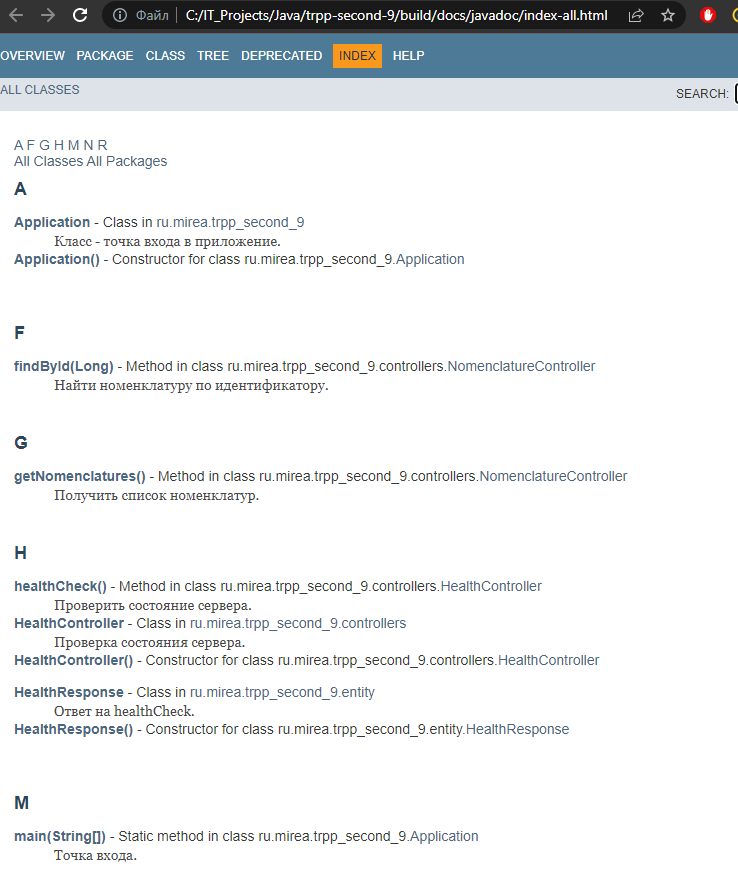


Рисунок 6 – Сгенерированная документация

1. Собрать jar со всеми зависимостями после чего запустить приложение. По умолчанию, сервер стартует на порту 8080:

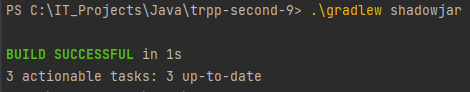


Рисунок 7 – Генерация Jar

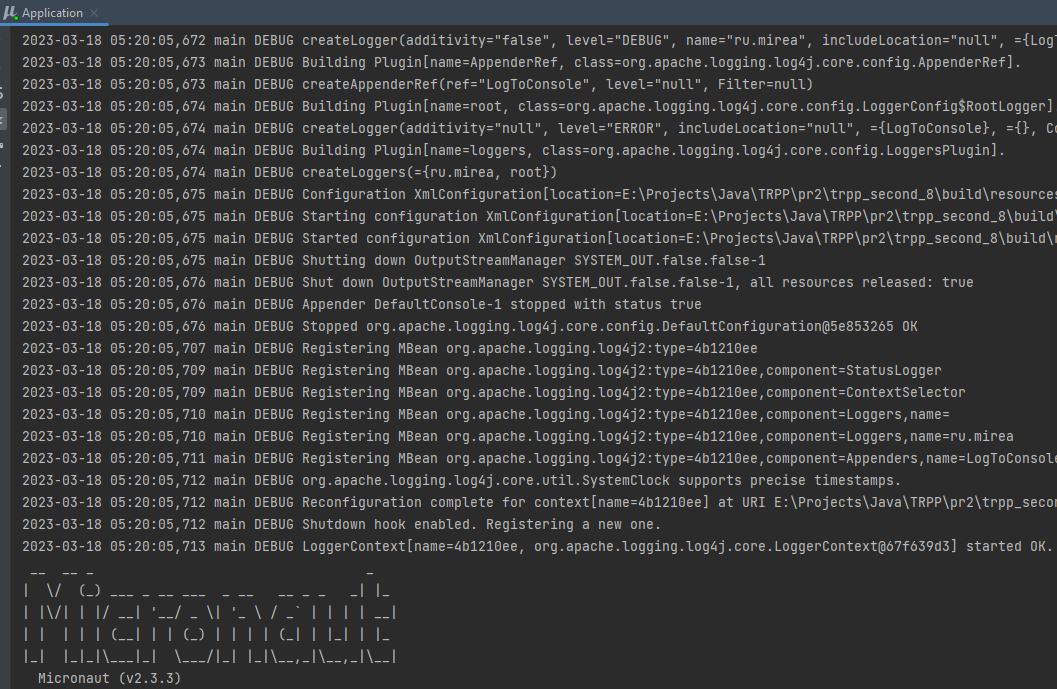


Рисунок 8 – Запуск приложения

1. Запросить состояние запущенного сервера (GET запрос по адресу http://localhost:8080, рисунок 8):

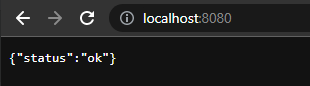


Рисунок 9 – Выполнение запроса к серверу по адресу

1. Запросить сущность по идентификатору (GET запрос по адресу: http://localhost:8080/сущность/идентификатор, рисунок 9):

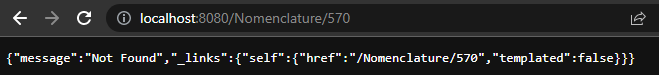


Рисунок 9 – Выполнение запроса к сущности проекта по id

1. В задаче shadowJar добавить к jar-файлу вашу фамилию (рисунок 10):

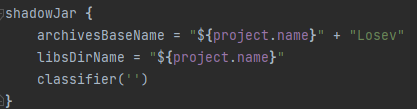


Рисунок 10 – Добавление к имени файла личных данных

1. Выполнить задачу checkstyleMain (рисунок 11). Посмотреть сгенерированный отчет (рисунок 12). Устранить ошибки оформления кода (рисунок 13):



Рисунок 11 – Выполнение задачи проверки код-стайла проекта



Рисунок 12 – Сгенерированный отчет по ошибкам код-стайла



Рисунок 13 – Успешное прохождение проверки после исправление ошибок

Контрольные вопросы

1. **Что такое mavencentrale?**

Maven Central — это Maven-репозиторий.

1. **Что такое maven?**

Фреймворк для автоматизации сборки проектов на основе описания их структуры в файлах.

1. **Что такое gradle?**

Cистема автоматической сборки, построенная на принципах Maven, но предоставляющая DSL на языках Groovy и Kotlin вместо традиционной XML-образной формы представления конфигурации проекта.

1. **Что делает run?**

Запускает основное приложение.

1. **Что такое аннотация в Java?**

Это своего рода метки в коде, описывающие метаданные для функции, класса или пакета.

1. **Чем компиляция отличается от сборки?**

Сборка выполняется при подготовке приложения к выпуску, которое включает компиляцию, упаковку, тестирование и т. д. Компиляция выполняется в любое время, когда компилятор занимается переводом кода языка программирования в машинный код.

## Выводы

В результате выполнения практической работы были получены навыки в работе с системой сборки Gradle.