

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**»** |

|  |  |
| --- | --- |
| **Выполнил:**  Студент группы ИКБО-11-22 | Гришин А. В. |
| **Проверил:**  Преподаватель | Копанева А. А. |

Москва 2024 г.

**Оглавление**

1. [Изменения которые были проведены в исходном коде проекте](#_bookmark0) [для всех пунктов задания 3](#_bookmark0)
2. [Скриншоты результатов, полученного в п.5-8 задания 7](#_bookmark1)
3. [Ответы на вопросы 11](#_bookmark2)
4. [Вывод 13](#_bookmark3)

# Изменения которые были проведены в исходном коде проекте для всех пунктов задания

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

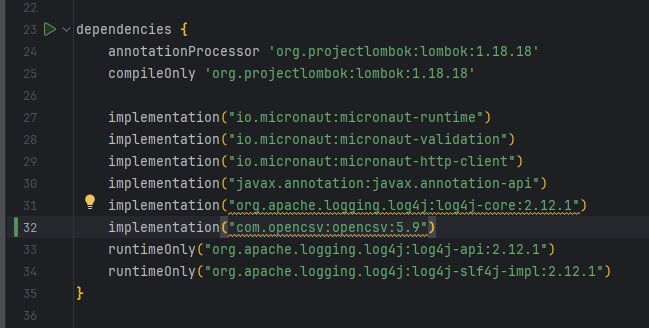


Рисунок 1 – Изменения в файле build.gradle

* 1. В некоторых классах поправить имя пакета

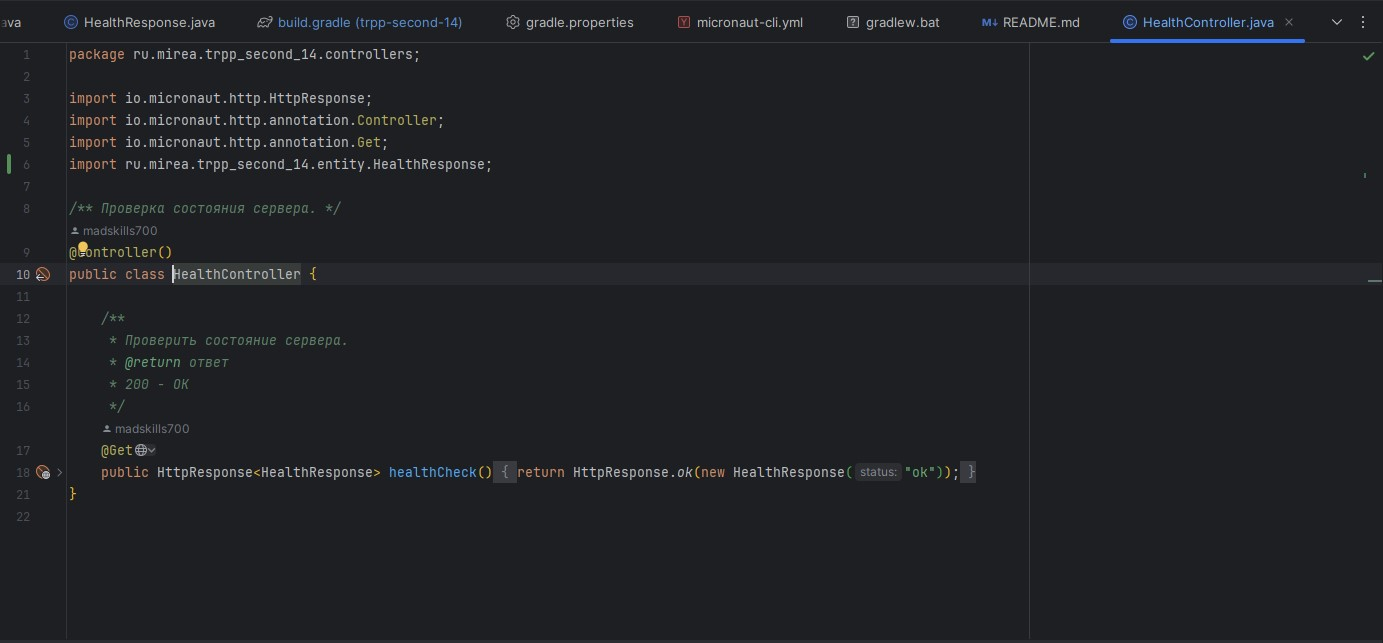


Рисунок 2 – Изменения в файле HealthController.java

Запустим проект с исправленными именами пакетов. Видим, что все собралось без проблем.

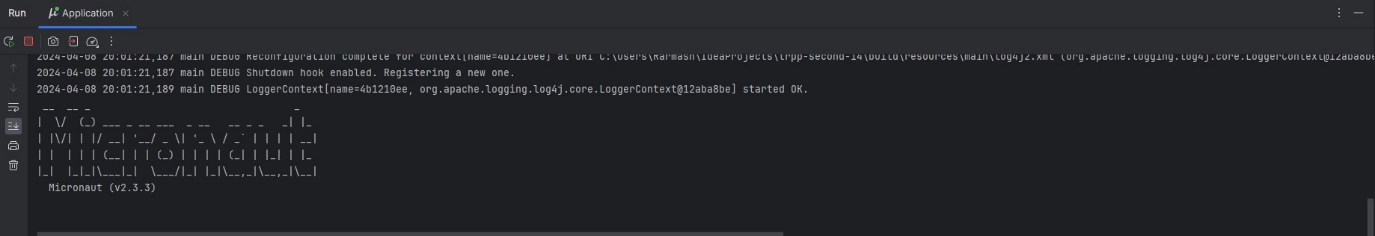


Рисунок 3 – Вывод консоли

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

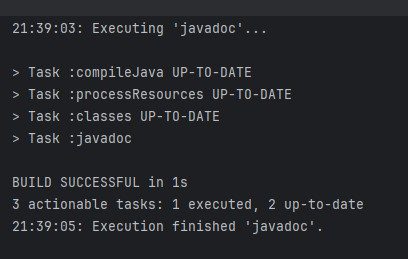


Рисунок 4 – Javadoc

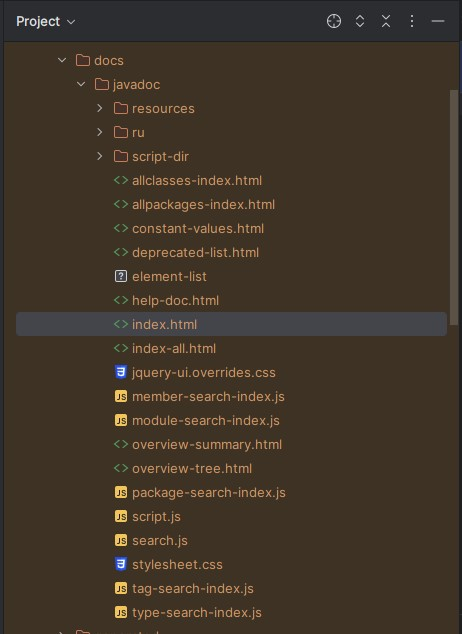


Рисунок 5 – Результат выполнения скрипта

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

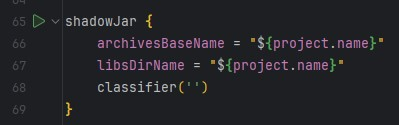


Рисунок 6 – shadowJar

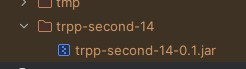


Рисунок 7 – Результат выполнения скрипта

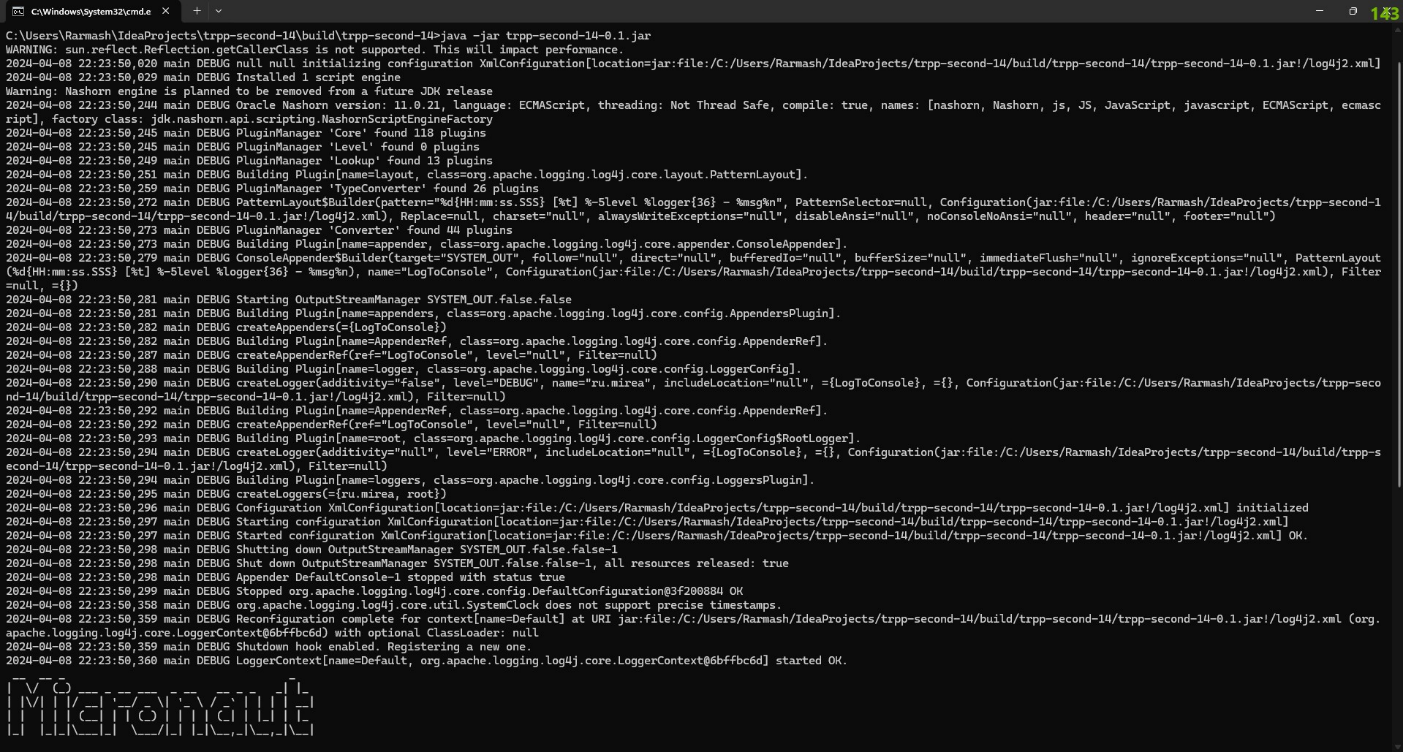


Рисунок 8 – Результат выполнения сборки

# Скриншоты результатов, полученного в п.5-8 задания

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

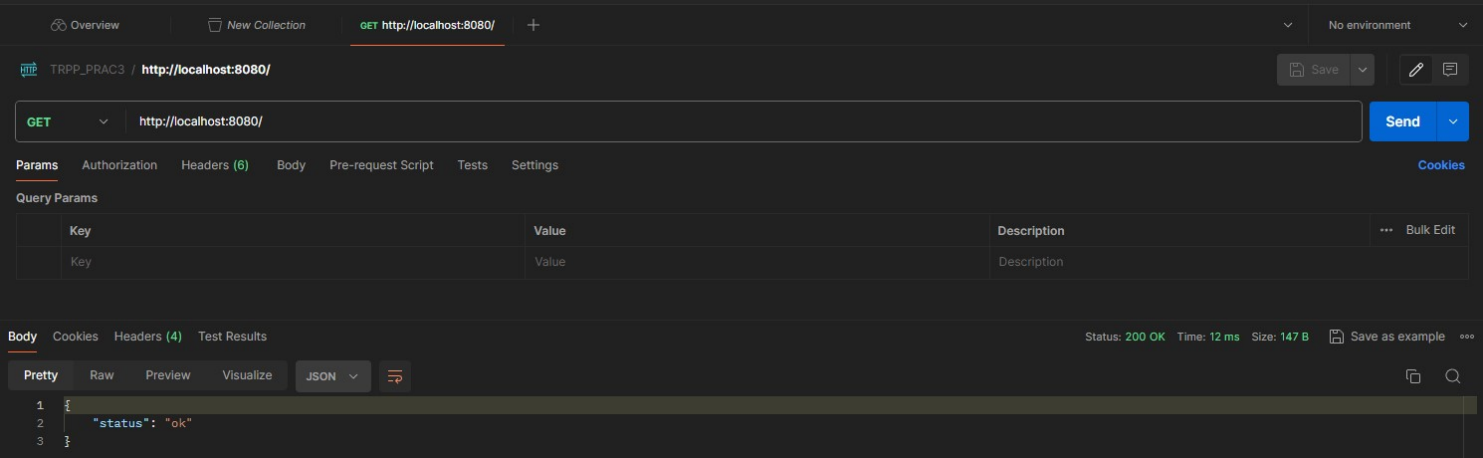


Рисунок 9 - Состояние запущенного сервера

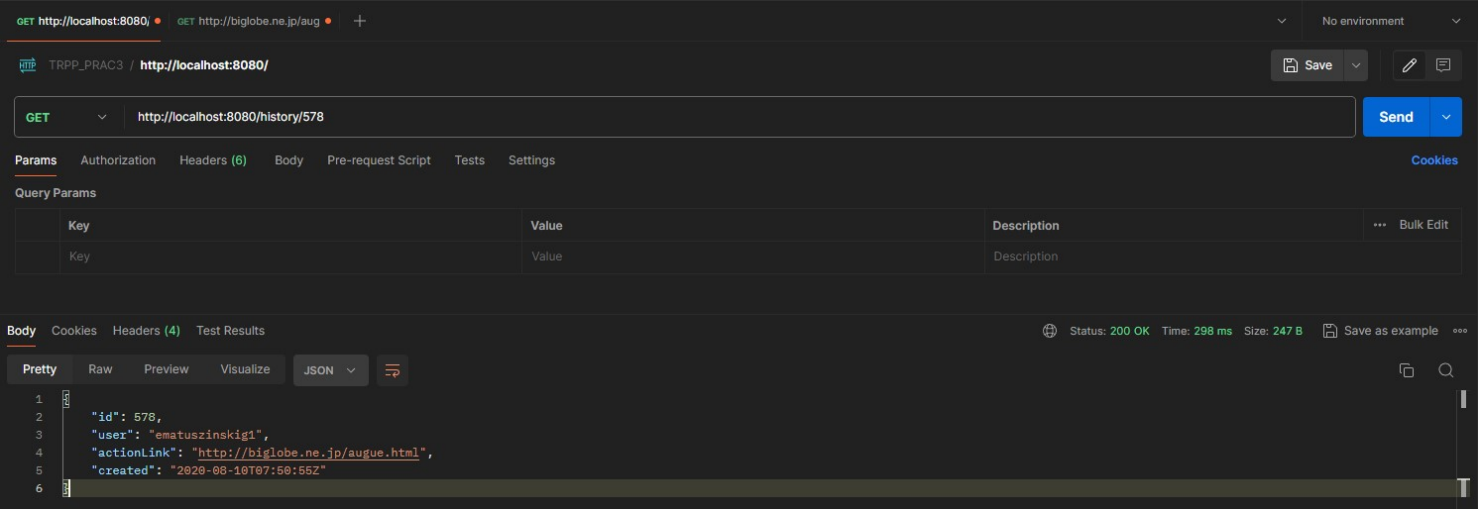
1. Запросить сущность по идентификатору (GET запрос по адресу: http://localhost:8080/сущность/идентификатор

Рисунок 10 – Запрос сущности по идентификатору

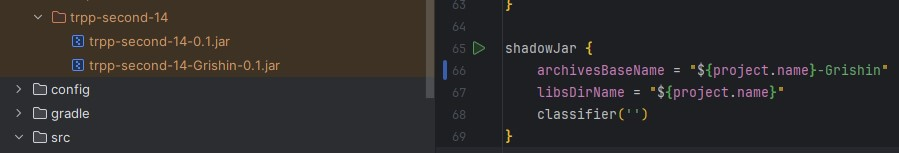
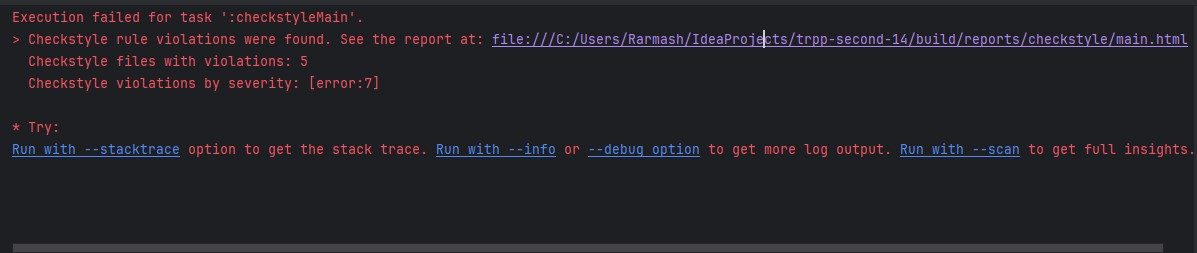
1. В задаче shadowJar добавить к jar-файлу вашу фамилию

Рисунок 11 – Добавление фамилии к jar-файлу

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

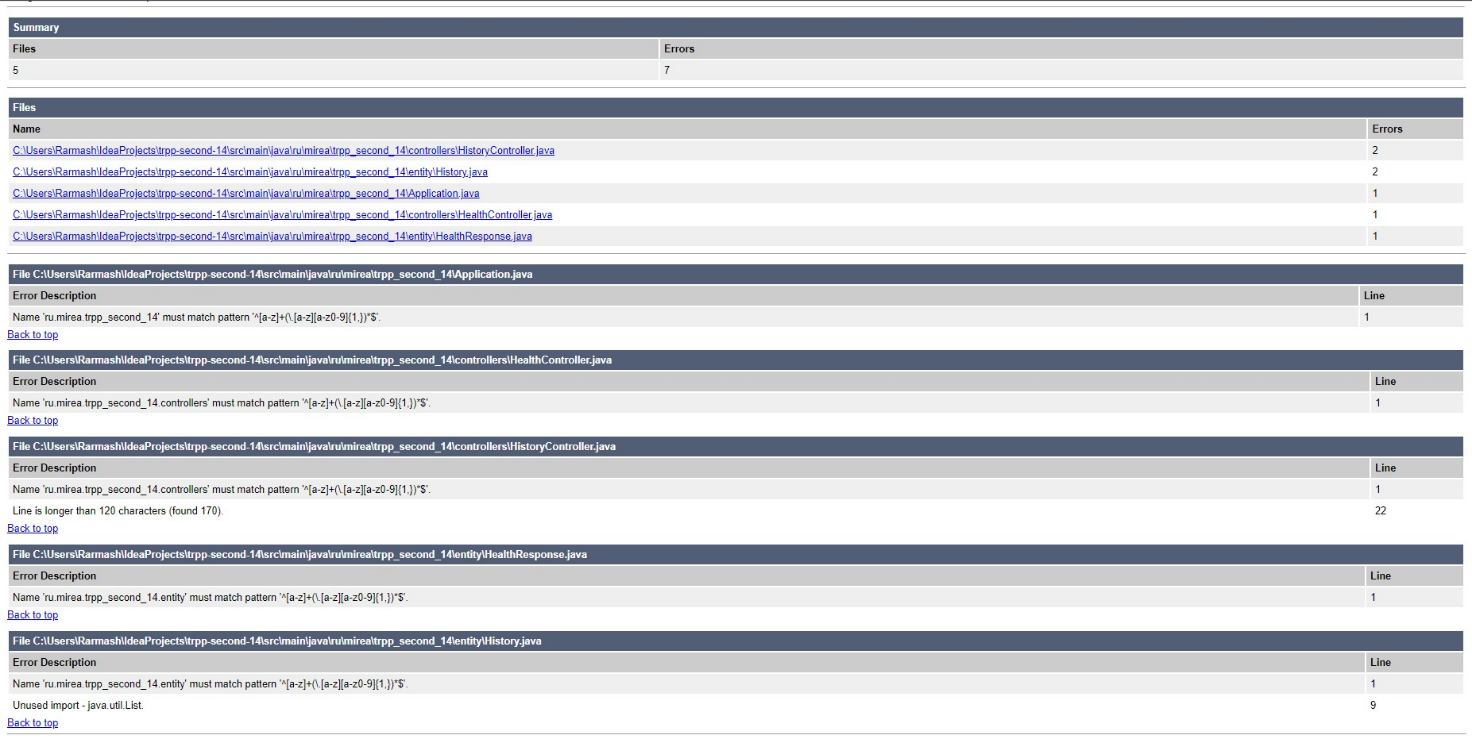
Рисунок 12 – Запуск checkstyleMain

Рисунок 13 – Выявленные ошибки

Рисунок 14 – Теперь нет ошибок

# Ответы на вопросы

5) Что такое Gradle?

* **Gradle** - это инструмент сборки проектов, который используется в основном для разработки программного обеспечения на языке Java, хотя его также можно использовать и для других языков программирования. Он предоставляет гибкий способ управления проектами и их зависимостями, а также автоматизации сборки, тестирования и развертывания приложений. Gradle использует DSL (Domain Specific Language) на основе Groovy или Kotlin для описания сборочных скриптов, что делает его более удобным и гибким по сравнению с другими инструментами сборки.

6) Что такое Maven?

* **Maven** - это инструмент управления проектами, который обеспечивает сборку, отчетность и управление зависимостями в проектах на Java. Он использует XML-файлы для описания структуры проекта, его зависимостей и сборочных целей. Maven автоматически загружает зависимости из центрального репозитория и обеспечивает стандартизированные структуры проектов, что делает его широко используемым инструментом в сообществе Java.

9) Что делает задача build?

* Задача **build** (сборка) в инструментах сборки, таких как Gradle или Maven, отвечает за компиляцию и сборку исходного кода проекта в исполняемый или дистрибутивный файл. Это включает в себя выполнение различных этапов, таких как компиляция исходного кода, копирование ресурсов, создание исполняемых файлов, запаковку и т. д.

10) Что делает задача compileJava?

* Задача compileJava отвечает за компиляцию исходного кода Java в байт-код Java (файлы .class). Она часто является частью процесса сборки проекта и выполняется перед другими этапами сборки, такими как создание исполняемых файлов или дистрибутивов.

13) Что такое javadoc?

* Javadoc - это инструмент, входящий в состав JDK (Java Development Kit), который автоматически генерирует документацию API на основе комментариев в исходном коде Java. Эти комментарии должны следовать определенным соглашениям о стиле и содержать информацию о классах, методах, полях и других элементах API. Сгенерированная документация предоставляет полезную справочную информацию для разработчиков, использующих библиотеку или API.

14) Что такое checkstyle?

* Checkstyle - это инструмент статического анализа кода для языка Java, который используется для обеспечения согласованности и соответствия кода определенным стандартам оформления. Он проверяет исходный код на соответствие правилам, определенным в конфигурационных файлах, и предоставляет отчет о найденных стилистических или структурных нарушениях. Checkstyle помогает обеспечить высокое качество кода, улучшить его читаемость и уменьшить количество потенциальных ошибок.

# Вывод

В ходе выполнения данной работы были приобретены и освоены навыки работы с gradle, сборкой проекта, работы с сервером.