|  |
| --- |
|  |
| 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** |
| **«Технологии разработки программных приложений»**  **Тема: «Системы сборки проекта»** |

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-08-21 | Юнаков И.П. |
| Принял | Туманов М.Б. |

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

Москва 2023

Оглавление

[Часть 1 3](#_Toc137051286)

[**1.** **Теоретическая часть** 3](#_Toc137051287)

[**2.** **Выполнение задач в gradle** 3](#_Toc137051288)

[**3.** **Задания для выполнения** 4](#_Toc137051289)

[4. Вывод 8](#_Toc137051290)

# Часть 1

## **Теоретическая часть**

Gradle — система сборки проектов с открытым исходным кодом, в которой основной упор идет на гибкость и производительность. В настоящее время поддерживаются 2 языка программирования для написания скриптов сборки: groovy и kotlin.

Для работы с Gradle достаточно иметь установленную JDK, однако для повышения качества проектов рекомендуется использовать Gradle Wrapper - небольшой jar-файл, файл конфигурации и скрипты для запуска на Windows и \*nix. При первом запуске Wrapper загружает Gradle нужной версии и начинает свою работу. Это необходимо, так как версия системы сборки может отличаться у разных участников команды и на билд-сервере.

Gradle - не первая система сборки для Java и Android-приложений, поэтому он может использовать хранилища артефактов, созданные для Maven.

Для проверки установки можно вызывать команду:

root@yunikeil:/mnt/proj gradlew

В ответ получим версию, а также небольшую справку.

## **Выполнение задач в gradle**

В Gradle задачи выполняются путем вызова Gradle Wrapper и перечисления нужных задач в нужной последовательности, при этом некоторые задачи по умолчанию зависят друг от друга. Например, если вы запускаете задачу build, то задача compileJava будет выполнена автоматически, и вызывать ее отдельно не нужно. Среди наиболее популярных задач в Gradle можно выделить clean для очистки всех сборочных директорий, build для сборки приложения целиком, javadoc для генерации JavaDoc документации, shadowJar для сборки UberJar (архива со всеми зависимостями, для использования которого необходимо импортировать плагин shadow), run для запуска приложения и test для прогона тестов. Рассмотрим базовый скрипт:

1. plugins {
2. id 'java'}
3. group 'org.text'
4. version '1.0'
5. repositories {
6. mavenCentral()}
7. dependencies {
8. testImplementation 'org.junit.jupiter:junit-jupiter-api:5.6.0'
9. testRuntimeOnly 'org.junit.jupiter:junit-jupiter-engine'}

Gradle имеет несколько разделов в файле сборки проекта, которые важно учитывать. В разделе "plugins" перечисляются плагины, используемые при сборке проекта. В разделе "Пакет и версия" указываются данные для идентификации проекта. В разделе "repositories" перечисляются репозитории, используемые для поиска и загрузки артефактов. В разделе "dependencies" указываются зависимости проекта, такие как "annotationProcessor", "implementation" и "api".

## **Задания для выполнения**

Для выполнения нужно клонировать репозиторий согласно вашему варианту. Задания будут выполняться согласно варианту 1.

Root@yunikeil:/mnt/proj>git clone https://github.com/rtu-mirea/trpp-second-1

remote: Enumerating objects: 35, done.

Cloning into 'trpp-second-1'... remote: Counting objects: 100% (8/8), done. remote: Compressing objects: 100% (8/8), done. Receiving objects: 100% (35/35), 104.46 KiB | 906.00 KiB/s, done.

Root@yunikeil:/mnt/proj>cd trpp-second-1

Root@yunikeil:/mnt/proj\trpp-second-1>gradlew.bat

Downloading https://services.gradle.org/distributions/gradle-6.7-bin.zip .10%. .20%. ..100% .30%.. .......40%. .......50%.........60%. . . . . . . 80%. .90%.

Welcome to Gradle 6.7!

Here are the highlights of this release: File system watching is ready for production use

Declare the version of Java your build requires

Java 15 support

For more details see https://docs.gradle.org/6.7/release-notes.html

Starting a Gradle Daemon (subsequent builds will be faster)

> Task :help

Welcome to Gradle 6.7.

To run a build, run gradlew <task>

To see a list of available tasks, run gradlew tasks

To see a list of command-line options, run gradlew --help

To see more detail about a task, run gradlew help --task <task>

For troubleshooting, visit https://help.gradle.org

BUILD SUCCESSFUL in 1m 5s

1 actionable task: 1 executed Root@yunikeil:/mnt/proj\trpp-second-1>,

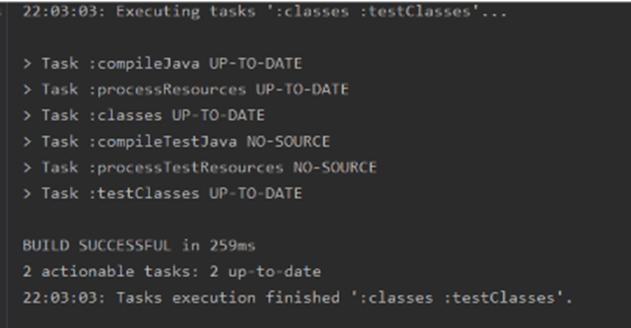
Для сборки репозитория необходимо устранить все ошибки, мешающие сборке. Необходимо было добавить новую зависимость в файл build.gradle - библиотеку OpenCSV для работы с .csv файлами. Также нужно было сделать импорт внутреннего класса (класс Employee) для корректной работы кода. Было исправлено название 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.

Добавление новой зависимости.

1. dependencies {
2. annotationProcessor 'org. projectlombok: lombok:1.18.26'
3. compileOnly 'org.projectlombok:lombok:1.18.26'
4. implementation("io.micronaut: micronaut-runtime")
5. implementation("io.micronaut: micronaut-validation")
6. implementation("io.micronaut:micronaut-http-client")
7. implementation("javax.annotation: javax.annotation-api")
8. implementation("org.apache.logging.log4j:log4j-core:2.20.0") // |
9. runtimeOnly("org.apache.logging.log4j:log4j-api:2.20.0") //
10. runtimeOnly("org.apache.logging.log4j:log4j-slf4j-impl:2.20.0") //
11. implementation(group: 'com.opencsv', name: 'opencsv', version: '5.6')
12. }

Импорт необходимого класса

1. package ru.mirea.trppsecond1.controllers;
2. Import com.opencsv.bean.CsvToBean Builder; //
3. import io.micronaut.http.HttpResponse;
4. import io.micronaut.http.annotation.Controller;
5. import io.micronaut.http.annotation.Get;
6. import ru.mirea.trppsecond1.entity.Employee;
7. //
8. import java.io.InputStreamReader; //
9. import java.util.List;
10. import java.util.Optional;
11. /\*\* Контроллер для работы с сотрудниками. \*/
12. 2 mad skills 700 \*
13. @Controller("/employee")
14. public class EmployeeController {
15. /\*\* Список сотрудников. \*/
16. 3 usages

Рисунок 1 – Успешная сборка проекта

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

Root@yunikeil:/mnt/proj>gradle javadoc

Starting a Gradle Daemon, 1 busy Daemon could not be reused, use --status for details

> Task : app: javadoc

Root@yunikeil:/mnt/proj\app\src\main\java\tn\App.java:6: warning: no comment

public class App {

A

Root@yunikeil:/mnt/proj\app\src\main\java\tn\App.java:7: warning: no comment

public String getGreeting() {

Root@yunikeil:/mnt/proj\app\src\main\java\tn\App.java:11: warning: no comment

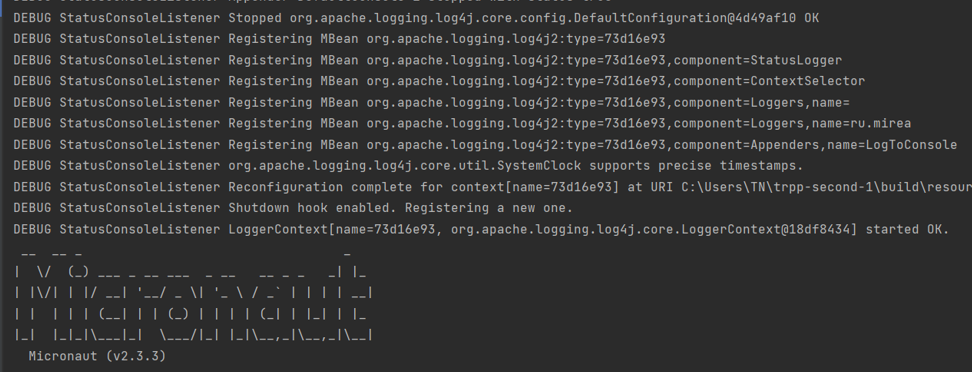
public static void main(String[] args) {

A

3 warnings

BUILD SUCCESSFUL in 41s

2 actionable tasks: 2 executed

Рисунок 2 – Запуск сервера

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

root@yunikeil:/mnt/proj# curl <http://localhost:8080>

{

status: «ok»

}

Далее необходимо запросить сущность по идентификатору, а именно выполнить GET запрос по адресу <http://localhost:8080/> сущность/идентификатор. Идентификатором будет 3 последних цифры в студенческом билете.

root@yunikeil:/mnt/proj# curl <http://localhost:8080/eployee/550>

{

id: 550

email: "despinaf9@alibaba.com "

firstName: "Denni"

lastName: "Espina"

birthDate: "2007-11-19T12:17:34Z"

company: "Lind-Schimmel"

department: "Product Management"

}

Добавление фамилии к названию jar файла

1. javadoc {
2. options.encoding = 'UTF-8'
3. shadowJar {{
4. archivesBaseName = "${project.name}-" + "YUNAKOV\_IKBO\_08\_21"
5. libdir name = "${project.name}"
6. classifier('')

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

> Task : compileJavacond-1>gradle checkstyleMain

Note: Creating bean classes for 1 type elements

warning: No processor claimed any of these annotations: /io.micronaut.http.annotation.Controller,/io.micronaut.http.annotation.Get

1 warning

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

See https://docs.gradle.org/7.5.1/userguide/command\_line\_interface.html#sec:command\_line\_warnings

BUILD SUCCESSFUL in 1s

3 actionable tasks: 2 executed, 1 up-to-date

Root@yunikeil:/mnt/proj

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.

> Connecting to Daemon

You can use '

'--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

See https://docs.gradle.org/7.5.1/userguide/command\_line\_interface.html#sec:command\_line\_warnings

BUILD SUCCESSFUL in 1s

3 actionable tasks: 3 up-to-date

--> 0% WAITING

IDLE

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.

> Connecting to Daemon

You can use '

'--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.

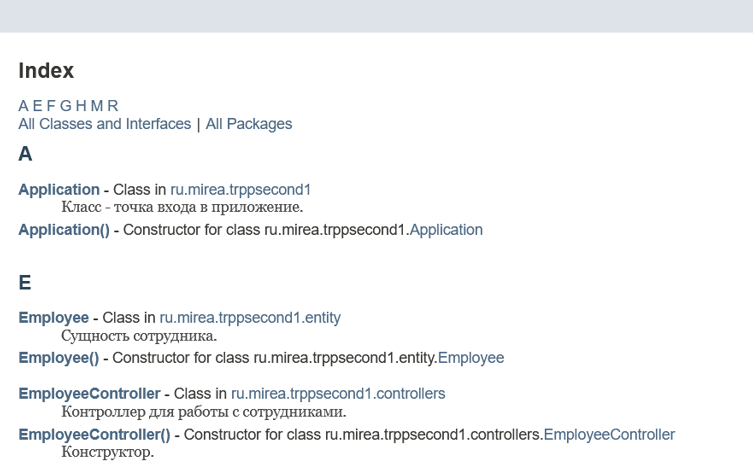
See https://docs.gradle.org/7.5.1/userguide/command\_line\_interface.html#sec:command\_line\_warnings

BUILD SUCCESSFUL in 4s

3 actionable tasks: 1 executed, 2 up-to-date

Root@yunikeil:/mnt/proj

Получить доступ к результатам сборки документации можно, открыв файл index.html, который находится в этой же папке.

Рисунок 3 – Просмотр документации

1. Вывод

В ходе данной работы я изучил инструмент автоматизации сборки Gradle. Были рассмотрены основные задачи, которые могут выполняться с помощью Gradle, такие как очистка проекта, сборка проекта, генерация документации, запуск тестов и другие. Также были рассмотрены различные разделы Gradle-файла, такие как "plugins", "Package and Version", "repositories" и "dependencies".