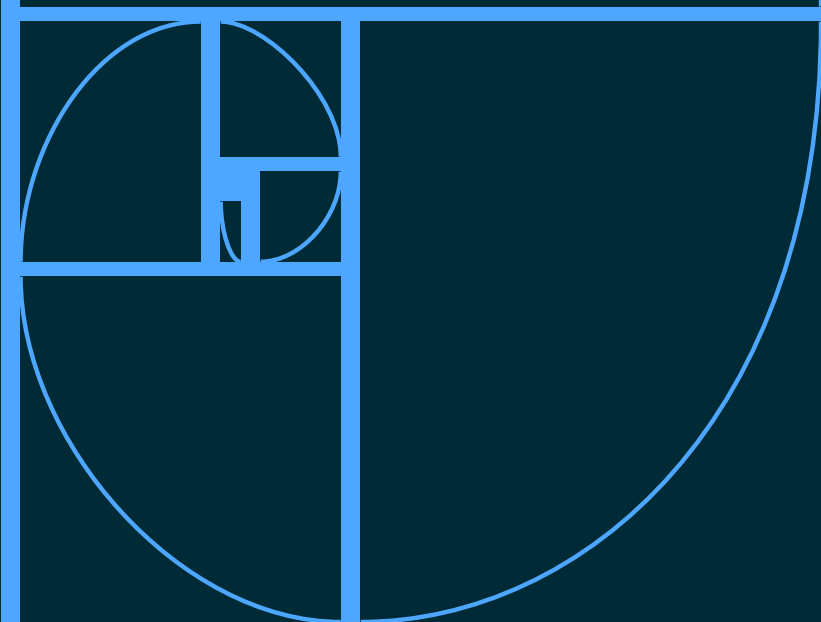
Инструментарий Gradle

Крапивина Ульяна, Иванов Александр

Политехнический университет, группа 5030102/30401



Что такое Gradle

Gradle - система сборки, которая автоматизирует:

- Компиляцию кода
- Управление Зависимостями
- Юнит Тестирование
- Создание дистрибутивов

Преимущества Gradle

Добавление зависимости в Maven:

```
// Maven (XML)
<dependencies>
 <dependency>
  <groupId>com.google.guava</groupId>
  <artifactId>guava</artifactId>
  <version>31.1-jre</version>
 </dependency>
</dependencies>
```

Преимущества Gradle

Эквивалент в Gradle:

```
// Gradle (Groovy) - короче и понятнее!
dependencies {
    implementation 'com.google.guava:guava:31.1-jre'
}
```

Стандартная структура

```
university-app/                  # Корень проекта (Project)
| build.gradle                   # Главный файл конфигурации
| settings.gradle                # Настройки проекта
| gradlew                        # Gradle wrapper (Unix)
| gradlew.bat                    # Gradle wrapper (Windows)
| src/
| | main/
| | | java/                  # Исходный код
| | | | com/university/
| | | | | Student.java
| | | | | Main.java
| | | resources/             # Ресурсы (properties, xml)
| | | | test/
| | | | | java/                  # Тесты
| | | | | | com/university/
| | | | | | | StudentTest.java
| build/                         # Результат сборки (создаётся автоматически)
| | classes/
| | libs/
```

"build.gradle"

Минимальный рабочий пример:

```
// 1. Плагины (что умеем делать)
plugins {
    id 'java'           // Умеем компилировать Java
    id 'application'    // Умеем запускать приложения
}
// 2. Информация о проекте
group = 'com.university'
version = '1.0.0'
```

"build.gradle"(Продолжение)

```
// 3. Главный класс для запуска
application {
    mainClass = 'com.university.Main'
}
// 4. Откуда скачивать библиотеки?
repositories {
    mavenCentral()  // Главный склад библиотек
}
// 5. Какие библиотеки нам нужны?
dependencies {
    implementation 'com.google.guava:guava:31.1-jre'
    testImplementation 'org.junit.jupiter:junit-jupiter:5.9.2'
}
```

"settings.gradle"Одномодульный проект

```
rootProject.name = 'university-app'  // Имя проекта
```

"settings.gradle"Многомодульный проект

```
rootProject.name = 'university-system'
// Подключаем модули
include 'core-module'
include 'web-module'
include 'data-module'
// Переименовываем если нужно
project(':core-module').name = 'university-core'
```

Что такое "Задача"

Задача - Атомарная операция сборки.
Примеры задач, встроенных в Gradle:

- compileJava - Компиляция Java кода
- test - запуск юнит тестов
- jar - сборка JAR файла
- clean - очистка папки сборки

Просмотреть все доступные задачи можно командой bash:
./gradlew tasks
Запустить задачу можно командой bash:
./gradlew taskname

Создание собственных задач

Собственные задачи определяются в файле build.gradle, например:

```
task buildReport {
    dependsOn build  // Зависит от задачи build
    doLast {
        def jarFile = file("build/libs/${project.name}.jar")
        println "=== ОТЧЕТ О СБОРКЕ ==="
        println "Проект: ${project.name}"
        println "Версия: ${project.version}"
        println "JAR файл: ${jarFile.name}"
        println "Размер: ${jarFile.length()} байт"
        println "Время сборки: ${new Date()}"
    }
}
```

Зависимость задач друг от друга

```
task compile {
    doLast {
        println 'Компилируем код...'
    }
}
task test {
    dependsOn compile  // Сначала выполни compile
    doLast {
        println 'Запускаем тесты...'
    }
}
```

При запуске задачи "test"сначала запустится задача "compile"

Типы зависимостей

```
dependencies {
// Основная зависимость - попадает в финальный JAR
implementation 'com.google.guava:guava:31.1-jre'
// Только для компиляции (например, аннотации Lombok)
compileOnly 'org.projectlombok:lombok:1.18.26'
annotationProcessor 'org.projectlombok:lombok:1.18.26'
// Только для тестов
testImplementation 'org.junit.jupiter:junit-jupiter:5.9.2'
// Для runtime (например, драйвер БД)
runtimeOnly 'mysql:mysql-connector-java:8.0.33'
// Зависимость от локального JAR
implementation files('libs/local-library.jar')
// Зависимость от другого модуля
implementation project(':core-module')
}
```

Репозитории

```
repositories {
mavenCentral() // Основной репозиторий
maven {
   url 'https://repo.spring.io/milestone' // Специальный репозиторий
}
jcenter() // Альтернативный репозиторий (устаревший)
// Локальная папка с JAR
flatDir {
    dirs 'libs'
}
// Корпоративный репозиторий
maven {
    url 'http://company-repo:8081/repository/maven-public/'
}
}
```

Анализ зависимостей

Полезные команды:

- ./gradlew dependencies - Показать все зависимости
- ./gradlew dependencies –configuration implementation - Показать зависимости только для компиляции
- ./gradlew dependencies –configuration compileClasspath -Показать дерево зависимостей
- ./gradlew dependencyInsight –dependency guava - Поиск конфликтов версий

Ошибки

1. Зависимость не найдена
   > Could not resolve: com.example:unknown-library:1.0
   Решение: Проверить название в Maven Central
2. Не та версия Java
   > incompatible types: java.lang.String cannot be converted to int
   Решение: Указать версию Java в build.gradle

```
java {
    toolchain {
        languageVersion = JavaLanguageVersion.of(17)
    }
}
```

Ошибки (Продолжение)

3. Конфликт версий > Conflict: version 2.5.0 of library conflicts with version 2.4.0
   Решение: Принудительно указать версию

```
dependencies {
    implementation('com.example:library:2.5.0') {
        force = true
    }
}
```

4. Задача не найдена
   > Task 'helloWorld' not found in project
   Решение: Проверить имя задачи: ./gradlew tasks

Ошибки (Продолжение)

5. Плагин не найден
   > Plugin with id 'java' not found
   Решение: Добавить настройки плагинов:

```
plugins {
    id 'java'
}
// Или для кастомных плагинов
buildscript {
    repositories {
        gradlePluginPortal()
    }
    dependencies {
        classpath 'com.example:custom-plugin:1.0'
    }
}
```

Что такое Gradle Wrapper

Gradle Wrapper - скрипт, вызывающий конкретную скачанную версию gradle.
Преимущества:

- Не требует установки Gradle на машине
- Использует конкретную версию Gradle
- Гарантирует одинаковые результаты сборки

Генерация wrapper командой bash:
gradle wrapper –gradle-version 8.5

Заключение

Gradle помогает с:

- Автоматизация сборки
- Управление зависимостями
- Масштабируемость проектов

Термины:

- Project - build.gradle + settings.gradle
- Task - действия (compile, test, jar)
- Dependency - внешние библиотеки
- Plugin - готовые наборы задач

Список Источников

- Официальная Документация Gradle
- Gradle user manual
- Посты на Различных Форумах