

ТЕХНОЛОГИЯ СОЗДАНИЯ ВЕБ-ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ GRADLE

А.Д. КУШНЕРЕВИЧ, Е.А. САЛЬНИКОВА

*Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники» филиал «Минский радиотехнический
колледж»*

Аннотация: создание веб-приложений на Java стало доступнее благодаря современным инструментам и технологиям. Одним из таких инструментов является Gradle, который представляет собой систему автоматизации сборки. Gradle упрощает процесс разработки, тестирования и развертывания приложений, позволяя разработчикам сосредоточиться на написании кода, а не на управлении зависимостями и конфигурацией сборки. В этой статье будет рассмотрено, что такое Gradle, процесс работы Gradle и отличия от Maven.

Gradle — это инструмент автоматизации сборки, предназначенный для управления проектами, написанными на языках, работающих на платформе Java Virtual Machine (JVM), таких как Java, Groovy и Kotlin. Он позволяет автоматизировать множество процессов, включая компиляцию кода, тестирование и развертывание приложений. Gradle использует декларативный подход, что делает конфигурацию сборки более понятной и гибкой. Это означает, что разработчики могут описывать, что именно нужно сделать, без необходимости прописывать все шаги процесса. [1]

Работа Gradle строится вокруг концепции задач. Каждая задача представляет собой атомарную единицу работы, которая выполняет определенные действия, такие как компиляция кода, запуск тестов или создание документации. Задачи могут иметь входные и выходные данные, что позволяет Gradle управлять зависимостями между ними. При запуске сборки Gradle выполняет задачи в соответствии с их зависимостями, что позволяет избежать выполнения ненужных процессов. [1]

Проект Gradle обычно содержит два основных файла: “build.gradle” и “settings.gradle”. Файл “build.gradle” определяет конфигурацию сборки, в том числе плагины и зависимости, необходимые для приложения. В этом файле указывается, какие библиотеки нужно подключить и как настроить проект. Файл “settings.gradle” используется для управления модулями проекта, позволяя определить, какие компоненты будут включены в сборку. Главную роль в расширении функциональности Gradle играют плагины. Они добавляют новые задачи и конфигурации, которые могут использоваться в проекте. Плагины необходимо добавить в файл “build.gradle”, чтобы интегрировать необходимые технологии в проект.

Когда запускается сборка Gradle, процесс проходит три основных этапа: инициализация, конфигурация и выполнение. На этапе инициализации Gradle определяет, какие проекты и модули будут включены в сборку. Затем в ходе конфигурации он регистрирует задачи и выполняет файлы конфигурации, чтобы подготовить проект к сборке. На последнем этапе выполнения происходит

фактическое создание артефактов, таких как WAR или JAR файлы, в результате чего приложение становится готовым к развертыванию. [2]

Maven использует XML для конфигурации, что делает его более статичным. В XML-файле (POM) можно описать зависимости, плагины и другие настройки. Однако добавление пользовательской логики в Maven требует использования плагинов, которые написаны на языке Java. Это значит невозможность просто написать код в POM-файле, как в Gradle. Вместо этого необходимо использовать заранее определенные плагины и их конфигурации. Конфигурация в Maven требует больше строк кода и может быть сложнее для понимания, особенно при добавлении пользовательской логики. [2]

Gradle поддерживает динамическое управление зависимостями. Он позволяет использовать такие функции, как: динамическая замена проектных зависимостей на внешние и наоборот, правила подстановки и замены для совместимых библиотек, централизованное версионирование и поддержка метаданных IVY. Maven использует статический подход к управлению зависимостями, где версии фиксируются в POM файле. Это может привести к проблемам с конфликтами версий и сложностями в управлении зависимостями при большом количестве библиотек.

Gradle использует инкрементальную сборку, что означает, что он выполняет только те задачи, которые были изменены с последнего запуска. Это значительно ускоряет процесс сборки, особенно в больших проектах. Maven выполняет полную сборку каждый раз, когда запускается, что медленнее по сравнению с Gradle.

Gradle поддерживает множество плагинов и позволяет разработчикам создавать свои собственные плагины. Maven также имеет множество плагинов, но добавление пользовательской логики может быть более сложным из-за ограничений XML и статического характера конфигурации.

Оба инструмента поддерживают Java, но Gradle также совместим с C, C++, Groovy и Kotlin, тогда как Maven чаще используется с Java, Scala, Ruby и C#. Это делает Gradle более универсальным инструментом для проекта, где могут использоваться различные языки. [2]

Gradle является мощным инструментом для создания веб-приложений на Java, который обеспечивает гибкость и удобство в управлении зависимостями и сборкой. Его возможности позволяют разработчикам сосредоточиться на написании качественного кода, минимизируя время, затрачиваемое на конфигурацию. Использование Gradle в сочетании с фреймворками, такими как Spring, открывает новые горизонты для создания современных веб-приложений, делая процесс разработки более эффективным и организованным.

Список использованных источников

1. Урванов Ф. В. Java. Состояние языка и его перспективы. / Ф.В. Урванов. Санкт-Петербург : БХВ-Петербург. – 2023.
2. Эванс Бенджамин. Java для опытных разработчиков. 2-е изд. – Санкт-Петербург : Питер. – 2024.