

Build Large Scale Applications Using Java Modularity And Project Jigsaw

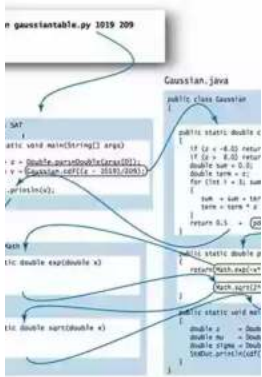
Java has long been hailed as one of the most popular programming languages due to its robustness, scalability, and versatility. However, as applications grow in size and complexity, managing dependencies and ensuring code maintainability becomes challenging. To address these issues, Java 9 introduced a groundbreaking feature called Project Jigsaw, which brings modularity to the Java platform. In this article, we will explore how Java modularity and Project Jigsaw can help developers build large-scale applications efficiently.

Understanding Java Modularity

Java modularity refers to breaking down a large application into smaller, manageable modules. Each module contains its own set of classes, interfaces, and resources, making it easier to understand and maintain code. Prior to Java 9, Java applications were built using a monolithic approach, where all code resided in a single module without any clear boundaries. This made it difficult to isolate and reuse components, resulting in code duplication and increased maintenance efforts.

With the of Project Jigsaw, Java developers can now define explicit module boundaries, allowing for better organization and separation of concerns. Modules can specify their dependencies and expose only the necessary APIs to other modules, reducing the chances of accidental dependencies and improving encapsulation.

Modular Programming in Java 9: Build large scale applications using Java modularity and Project



Jigsaw by Gerald M. Weinberg(1st Edition, Kindle Edition)

★★★★☆ 4.5 out of 5

Language : English
File size : 14962 KB
Text-to-Speech : Enabled
Screen Reader : Supported
Enhanced typesetting : Enabled
Print length : 298 pages



Benefits of Java Modularity

Improved Code Maintainability

By breaking down an application into smaller, self-contained modules, developers can easily focus on maintaining and enhancing specific features without affecting the rest of the codebase. This modular approach enables teams to work in parallel, ensuring faster development cycles and smoother code maintenance. Additionally, modules can be tested independently, reducing the testing effort required for the entire application.

Enhanced Code Reusability

Java modularity allows developers to reuse modules across different projects, saving time and effort. Modules can be shared as libraries or compiled into standalone JAR files, making it easier to integrate them into other applications. This not only promotes code reuse but also enhances collaboration among development teams, as they can share and rely on well-defined modules.

Better Dependency Management

Managing dependencies has always been a challenge in large-scale applications. With Java modularity and Project Jigsaw, dependency management becomes more explicit and deterministic. Modules declare their dependencies, allowing the Java runtime to resolve conflicts and ensure that the required dependencies are available at runtime. This eliminates the need for manual configuration or relying on external dependency management tools.

Improved Security

Java modularity enhances application security by providing clear module boundaries and encapsulation. Modules can expose or hide their implementation details, allowing only the necessary APIs to be accessed by other modules. This prevents unintended access to internal code and reduces the risk of introducing security vulnerabilities. Additionally, module boundaries promote better isolation, making it easier to apply security measures such as access control and auditing.

Using Project Jigsaw in Practice

To utilize Project Jigsaw and Java modularity, developers need to understand the new module system introduced in Java 9. The module system defines a specific structure for organizing code into modules, specifying dependencies, and providing access to external modules. Let's explore the key concepts and steps involved in using Project Jigsaw effectively.

Creating Modules

To create a module, developers need to define a module descriptor file (module-info.java) at the root of the module directory. This file declares the module name, dependencies, and other module-specific configurations. By explicitly defining module boundaries, developers can ensure that only essential APIs are exposed, promoting encapsulation and reducing accidental dependencies.

Resolving Dependencies

With Project Jigsaw, the Java runtime is responsible for resolving module dependencies. The runtime analyzes the module descriptor files and ensures that the required modules are present at runtime. This eliminates the need for developers to manually manage dependency configurations or rely on external tools like Maven or Gradle. The modularized application can run using the `java` command with the `--module` option, specifying the main module to be executed.

Migrating Existing Code

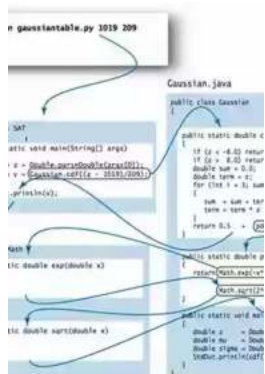
Migrating existing code to use the new module system can be a bit challenging, as it requires analyzing dependencies and modularizing the codebase. However, the benefits of modularity and enhanced code maintainability make it a worthwhile effort. Developers can start by identifying core functionalities and breaking them down into separate modules. Gradually, the codebase can be modularized, reducing the complexity and improving the overall design of the application.

Java modularity and Project Jigsaw bring a paradigm shift in how large-scale applications are developed and maintained. By embracing modularity, developers can improve code maintainability, enhance code reusability, and simplify dependency management. Project Jigsaw provides the necessary tools and runtime support to make modularity a reality in Java applications. To stay ahead in the ever-evolving software landscape, developers should leverage Java modularity and Project Jigsaw to build scalable and maintainable applications.



“Unlock the power of Java modularity and create large-scale applications with ease using Project Jigsaw!”

Keywords for alt attributes: build large scale applications using Java modularity, Project Jigsaw, Java 9, code maintainability, code reusability, dependency management, application security.



Modular Programming in Java 9: Build large scale applications using Java modularity and Project Jigsaw

by Gerald M. Weinberg(1st Edition, Kindle Edition)

★★★★☆ 4.5 out of 5

Language : English
File size : 14962 KB
Text-to-Speech : Enabled
Screen Reader : Supported
Enhanced typesetting : Enabled
Print length : 298 pages



Key Features

- Master design patterns and best practices to build truly modular applications in Java 9
- Upgrade your old Java code to Java 9 with ease
- Build and run a smooth functioning multi-module application.

Book Description

The Java 9 module system is an important addition to the language that affects the way we design, write, and organize code and libraries in Java. It provides a new way to achieve maintainable code by the encapsulation of Java types, as well as a way to write better libraries that have clear interfaces. Effectively using the module system requires an understanding of how modules work and what the best practices of creating modules are.

This book will give you step-by-step instructions to create new modules as well as migrate code from earlier versions of Java to the Java 9 module system. You'll be working on a fully modular sample application and add features to it as you learn about Java modules. You'll learn how to create module definitions, setup inter-module dependencies, and use the built-in modules from the modular JDK. You will also learn about module resolution and how to use jlink to generate custom runtime images.

We will end our journey by taking a look at the road ahead. You will learn some powerful best practices that will help you as you start building modular applications. You will also learn how to upgrade an existing Java 8 codebase to Java 9, handle issues with libraries, and how to test Java 9 applications.

What you will learn

- Get introduced to the concept of modules and modular programming by working on a fully modular Java application
- Build and configure your own Java 9 modules
- Work with multiple modules and establish inter-module dependencies
- Understand and use the principles of encapsulation, readability, and accessibility
- Use jlink to generate fully loaded custom runtime images like a pro
- Discover the best practices to help you write awesome modules that are a joy to use and maintain
- Upgrade your old Java code to use the new Java 9 module system

About the Author

Koushik Kothagal is the founder of Java Brains, an online training website that offers courses on various enterprise Java and JavaScript technologies entirely for free. He works as a Senior Staff Engineer at Financial Engines. He has over 14 years of professional experience working on full-stack web applications and has worked extensively with technologies such as Java, Spring, Java EE, JavaScript, and Angular. He loves teaching, and when he's not coding Java and JavaScript, he's probably teaching it! He currently lives in the Bay Area.

Table of Contents

1. What's Java modularity all about?
2. Writing your first Java module
3. Handling inter-module dependencies
4. Introducing the modular JDK
5. Using Platform APIs
6. Module resolution, accessibility and readability
7. Introducing Services
8. Understanding linking and using jlink
9. Module design patterns and strategies
10. Upgrading your code to Java 9
11. Java 9 and OSGi
12. Using build systems and testing Java modules



The Secrets of Chaplaincy: Unveiling the Pastoral Theology of Inquiry Haworth

Chaplaincy is a field that encompasses deep empathy, understanding, and spirituality. It is a profession where individuals provide spiritual care and support to those in...



Animales Wordbooks: Libros de Palabras para los Amantes de los Animales

Si eres un amante de los animales como yo, entonces seguramente entenderás la fascinación que sentimos hacia estas increíbles criaturas. Ya sea que se trate de majestuosos...



Let's Learn Russian: Unlocking the Mysteries of the Cyrillic Script

Are you ready to embark on a linguistic adventure? Have you ever been curious about the beautiful Russian language? Look no further - this article is your...



The Incredible Adventures of Tap It Tad: Collins Big Cat Phonics For Letters And Sounds

Welcome to the enchanting world of phonics where learning to read becomes a captivating journey! In this article, we will explore the marvelous educational resource,...



Schoolla Escuela Wordbookslibros De Palabras - Unlocking the Power of Words!

Growing up, one of the most significant milestones in a child's life is learning how to read. It opens up a whole new world of possibilities, imagination, and knowledge. A...



15 Exciting Fun Facts About Canada for Curious Kids

Canada, the second-largest country in the world, is famous for its stunning landscapes, diverse wildlife, and friendly people. As children, it's essential to...



What Did He Say? Unraveling the Mystery Behind His Words

Have you ever found yourself struggling to understand what someone really meant when they said something? Communication can often be clouded with ambiguity, leaving us...



A Delicious Journey through Foodla Comida Wordbookslibros De Palabras

Welcome to the world of Foodla Comida Wordbookslibros De Palabras, where colorful illustrations and engaging words come together to create a delightful learning...