Chapter 4: Java Tools

Java is designed as a language that allows for freedom of the developer to create however and whatever they want. Java is incredibly adaptable, and this is due in part to the many tools available for the developers to use. First, there are plenty of options for IDEs in Java development. The book mentions a few such as Eclipse, NetBeans, and IntelliJ. Others may include XCode (Apple exclusive), BlueJ, and JDeveloper. Tools for building projects are common as well, and the most popular are Apache Ant, Maven, and Gradle. Maven is described as essential for its simplicity and adaptability. Upon install, it does require some configuration as it is not fully functional out of the box. Plug-ins for the program allow many functionalities, such as standard unit testing using JUnit, building project websites to check progress of development, easily implementing an embedded application server, and using other languages inside the Java project. Version control software is another essential part of a professional Java development environment. These programs allow developers to view previous revisions of code, revert to old versions, create changelogs, patch the application, and create branches used for parallel development (allowing a team to collaborate on a project much easier). Java Virtual Machine (JVM) is useful for testing your applications in a real time environment, which is essential for debugging and compatibility verification. Java also makes it easy to convert JSON to other languages, such as Clojure, JavaScript, Jython, and Groovy. Typically, a Java application built for this purpose includes a main function that iterates through each argument, checking to see if the files listed are present. If not present, an exception will be thrown and the argument may be skipped by defaulting to the src/main/resources/scripts directory. These are just some examples provided by the book, but there are plenty of other options for developers to choose from.