Writing Java Programs

Exercises and solutions

1. What is the extension of the file that contains the source code for a Java program?

Typically, the file extension is .java. The Java Language Specification also allows .jav as the file extension.

1. What is a compilation unit?

A compilation unit contains the source code for type declarations. It consists of three parts: One package declaration, zero or more import declarations, zero or more type declarations in that order.

1. How many types can you declare in a compilation unit?

Typically, a compilation unit contains the declaration of one type. However, you can have multiple type declarations in one compilation unit. There is no upper limit on the number of type declarations you can have in a compilation unit.

1. How many public types can you declare in a compilation unit?

Maximum of one public type is allowed per compilation unit.

1. What is the restriction on naming a compilation unit if it contains a public type? What would be the name of the compilation unit if it contains the declaration of a public class named HelloWorld?

The name of the file of the compilation unit must be the same as the name of the public type. File name must be HelloWorld.java.

1. In what order the following constructs are specified in a compilation unit: type declaration, packages, and import statements?

One package declaration, followed by zero or more import declarations, followed by zero or more type declarations.

1. How many package statements can you have in one compilation unit?

A compilation unit can only have one package statement.

1. What is the extension of a file that contain Java compiled code?

The extension is .class.

1. What are the names of files that contain the source code and the compiled code for a Java module?

module-info.java and module-info.class

1. What keyword do you use to declare a module?

module

1. How many modules can you declare in one module-info.java file?

You can declare only one module in a module-info.java file?

1. What is an unnamed module? How many unnamed modules a class loader may have? When does a type (e.g. a class) become a member of an unnamed module?

Every class loader has a module called unnamed module. Types from the class path are loaded by a class loader into the class loader's unnamed module.

A class loader can have one unnamed module.

All types loaded from the class path become members of the unnamed module of their class loader.

1. What is a JAR? What is the difference between a JAR file and a ZIP file?

JAR is an acronym for Java Archive. Java compiled code in class files and application resources are packages into JAR files.

A JAR file is a ZIP file with a .jar extension with a MANIFEST.MF file in its META-INF directory.

1. What is a modular JAR and how does it differ from a JAR? Can you use a modular JAR as a JAR and vice versa?   
   Hint: A modular JAR is also a JAR and it can be used as such. A JAR placed on the module path acts as a modular JAR and in that case the module definition is automatically derived by the module system; such a module is known as an automatic module.

If a JAR file contains a module-info.class file, which is also known as a module descriptor, the JAR file is called a modular JAR.

1. What command do you use to start the JShell tool and where is the command located?

The command to start the JShell tool is jshell. JShell tool/command is in the JDK\_HOME\bin directory.

1. What command do you use to compile Java source code?

javac

1. What command do you use to package Java compiled code into a JAR or a modular JAR?

jar

1. Where is the module descriptor (module-info.class file) placed in a modular JAR?

Module descriptor is placed in root directory of a modular JAR.

1. You have a modular JAR saved at C:\lib\com.jdojo.test.jar. It contains a module named jdojo.test and a main class named com.jdojo.test.Test. Write the command to run this class in module mode and in legacy mode.

// In module mode

java –module-path C:\lib --module jdojo.test/com.jdojo.test.Test

// In lagacy mode

java --class-path C:\lib\com.jdojo.test.jar com.jdojo.test.Test

1. You have a modular JAR saved at C:\lib\com.jdojo.test.jar. Write the command using the jar command to describe the module packaged in this modular JAR.

jar –-file C:\lib\com.jdojo.test.jar –describe-module

1. What is a module descriptor? Can you specify the version for a module when you declare the module? How do you specify a module version?

The module-info.class in modular JAR is called module descriptor. You cannot specify the version of a module when you declare the module. You specify the version of a module when you package the module's code into a modular JAR using the jar tool. You can also specify the version of the module when you compile a module using the javac command. Both tools, jar and javac, uses the --module-version command-line option to specify a module's version.

1. What are observable modules? What are root modules and how are they used in a constructing a module graph?

Observable modules are the modules accessible to the module system at compile-time, link-time, and runtime.

Initial set of modules used in module resolution are called root modules.

Module resolution starts with root modules and follows the dependency chain until the java.base module is reached. A module graph is a set of resolved modules where each module is represented as a node. A directed edge from a module to another module exists if the first module depends on the second module.

1. Write the name of a command-line option that is used to add modules to the set of root modules.

Modules can be added to root modules using the --add-modules command-line option.

1. What command-line option do you use to print the list of observable modules?

A list of observable modules can be printed using the --list-modules command-line option with the java command.

1. What command-line option do you use to limit the set of observable modules?

The command-line option to limit the number of observable modules is --limit-modules.

1. The GNU-style option name for specifying the module path is --module-path. What is its equivalent UNIX-style option?

The equivalent UNIX-style option for --module-path is -p.

1. What are the options to print the help on a command? How do you print extra help for non-standard options available for a command?

For all standard options, run a tool with the --help option. For all non-standard options, run the tool with --help-extra option. The following commands will print standard and non-standard options for the java command:

java --help

java --help-extra