The Java Language

Mark A. Austin

University of Maryland

austin@umd.edu ENCE 688P, Fall Semester 2020

September 28, 2020

Overview



- 2 Basic Stuff
 - Primitive Data Types, IEEE 754 Floating Point Standard
 - Three types of Java Variable
 - Arithmetic Operations
 - Control Statements
- Packages and Import Statements
- 4 Methods
 - Polymorphism and Class Methods
- Working with Arrays
 - One- and Multi-dimensional Arrays; Ragged Arrays

Part 2



Quick Review

•0

Quick Review

Popular Computer Languages

Tend to be designed for a specific set of purposes:

- FORTRAN (1950s today). Stands for formula translation.
- C (early 1970s today). New operating systems.
- C++ (early 1970s today). Object-oriented version of C.
- MATLAB (mid 1980s today). Stands for matrix laboratory.
- Python (early 1990s today). A great scripting language.
- HTML (1990s today). Layout of web-page content.
- Java (1994 today). Object-Oriented language for network-based computing.
- XML (late 1990s today). Description of data on the Web.



Packages

Java Packages

Purpose:

- Every class is part of a package, and every package is identified by its name.
- Packages provide a high-level layer of access protection and name-space management for collections of Java classes, interfaces, exceptions, and errors.
- Packages reduce the likelihood of name clashes because class and interface names are evaluated with respect to the package to which they belong.
- A package may include other packages (i.e., subpackages).

Java Packages

Simple Example

The statement

```
package fruit;
```

defines a package called fruit.

There needs to be a one-to-one correspondence between the package name and a hierarchy of folders containing the Java source code.

Import Statements

Definition

An import statement makes Java classes available to a program under an abbreviated name.

Import statements come in two forms:

```
import package.class;
import package.*;
```

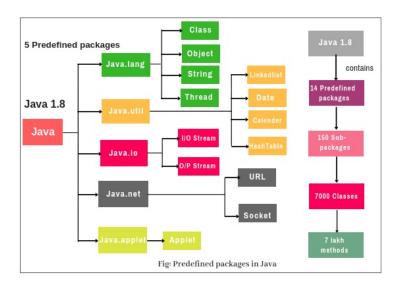
The first form allows a class to be referred to by its class name alone. The asterisk (*) in the second form references all the classes in the named package.

Importing classes from java.lang.System

The java.lang.System package is automatically imported into every Java program.



Packages in Java 1.8



Packages in Java 1.8

The Standard Edition of Java provides 14 predefined packages, e.g.,

java.lang. – contains the fundamental classes, methods, and interfaces of the Java language. It also contains the math library class java.lang.Math.

java.io. – contains classes and methods to read and write data to standard input and output devices, streams and files.

java.net. – contains classes to implement network operations.

java.text. – contains classes and interfaces to handle text.

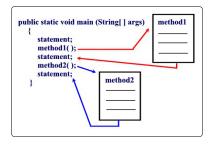
java.util. – contains collections framework, collection classes, classes related to time, dates and events.

Methods

Definition of Methods

Definition

A method is a set of code which is referred to by name and can be called (invoked) at any point in a program.



It is convenient to think of a method as a subprogram that acts on data and often returns a value.



Elements of Java Method

Name:

All Java methods will have a name.

Argument List:

- Most methods in Java will pass information from the calling method via an argument list.
- Occasionally we will encounter methods that have empty (void) argument lists.

Return Value:

- Most of the Java methods we will encounter will return information to the calling method via the return value.
- Occasionally we see functions that do not return a data type (void return type).



Syntax for Defining a Method

The syntax for making a method definition in Java is

```
modifier return-type name-of-method ( parameter-list ) {
    ... executable statements ....
} <=== end of the method body.</pre>
```

Key points:

- The modifier establishes the method type and its scope (i.e., what other methods can call it).
- The return-type specifies the type of information the method will return.
- Methods that do not return anything should use the return type void.



Class and Method Modifiers

	Modifier	Interpretation in Java
	=======	
	abstract	The method is provided without a body; the body will be provided by a subclass.
	final	The method may not be overridden.
	native	The method is implemented in C or in some other platform-dependent way. No body is provided.
	private	Method is only accessible from within the class that defines it.
	public	The method is accessible anywhere the class is accessible.
	static	Only one instance of a static member will be created, no matter how many instances of the class are created.

Passing Arguments to Methods

Pass-By-Value Mechanism (for basic data types):

Java passes all primitive data type variables and reference data type variables to a method by value. In other words, a copy of the variable's value is used by the method being called.

Example

See the TryChange.java code in java-code-basics

Polymorphism

Definition

Polymorphism is the capability of an action to do different things based on the details of the object that is being acted upon.

This is the third basic principle of object oriented-programming.

Polymorphism of Methods

See the DemoPolymorphism program in java-code-basics.

```
public static void doSomething() { ....
public static void doSomething( float fX ) { ....
public static void doSomething( double dX ) { ....
```

Three versions of a method with the same name!

Class Methods

Definition of Class Methods

A class method is a method that does not require an object to be invoked.

They are called in the same manner as instance methods except that the name of the class is substituted for the instance name.

Examples

```
System.out.println("Here is a line of text ...");
double dAngle = Math.sin( Math.PI );
```

Here, Math.sin() is a class method in the math library. Math.PI is a constant.