CS260 Intro to Java & Android
02.Java Technology

Winter 2014

Getting Started: http://docs.oracle.com/javase/tutorial/getStarted/index.html

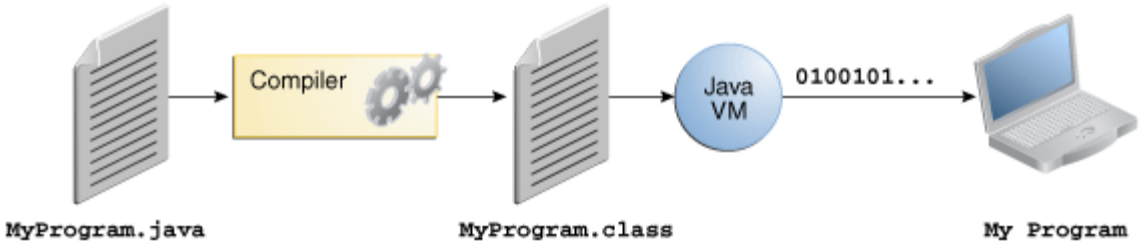
Java Technology is: (a) a programming language and (b) a platform

Java is a high-level programming language with the following characteristics:

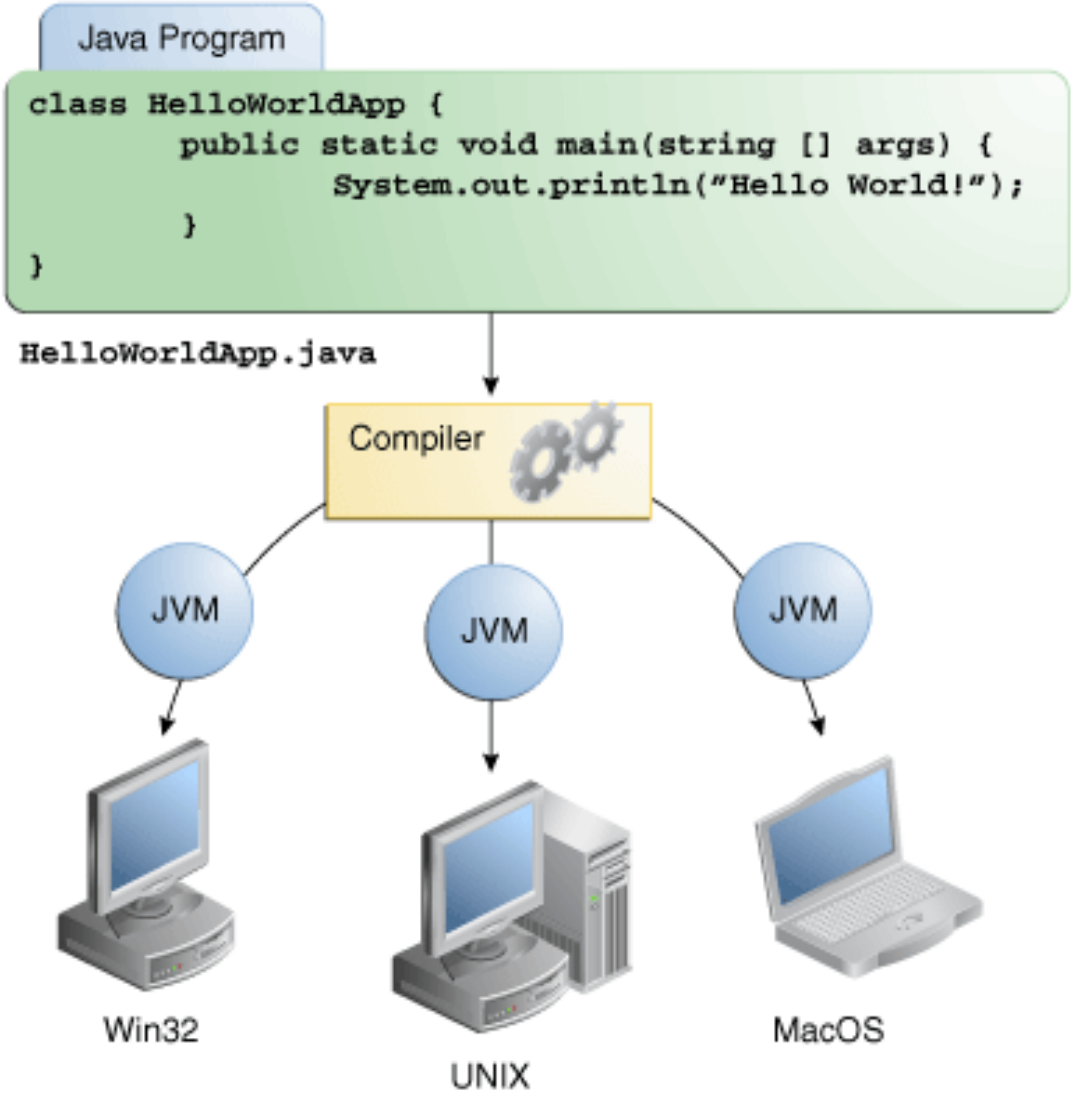
- Architecture neutral
- Object-oriented
- Portable
- Multithreaded
- Dynamic

In Java:

- all source code is first written in plain text files with a .java extension
- the source files are compiled into .class files by the compiler (javac)
- a .class is not code native to your processor
- a .class contains “bytecodes” which is the machine language of the Java Virtual Machine (JVM)
- the java launcher (java) runs you’re application with an instance of the JVM



Java “bytecode” can run on any JVM?

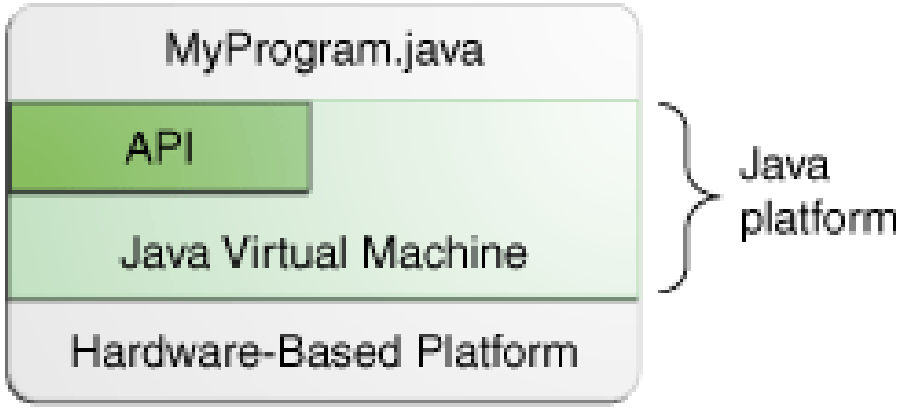


The Java Platform consists of two components:

- The JVM
- The Java Application Programming Interface (API)

Java API - a large collection of ready-made software components

Java API - grouped into libraries of related classes and interfaces; these libraries are known as *packages*



The Java Platform can be slower than native code.

Advances in compiler and VM technologies are greatly improving performance.

Example - Critical sections of bytecode can be compiled into native code and executed.

“HelloUser” for Microsoft Windows

Download the most recent Java Development Kit (JDK) which includes:

a Java Runtime Environment (JRE)
http://www.oracle.com/technetwork/java/javase/downloads/index.html

Q1: What is the JDK?

A: Command-line programs used to:

- create programs
- compile programs
- run Java programs

P1: Go to the Start menu and Run the command **cmd** to bring up a shell or “command” window.

Then enter the following command: > **javac –version**

javac 1.7.0_XX

P2: Enter the following command: > **java –version**

java version "1.7.0_XX"
Java(TM) SE Runtime Environment (build 1.7.0_XX)
Java HotSpot(TM) 64-Bit Server VM (build XX, mixed mode)

P3: Create a folder in your “My Documents” folder called “CS260\Java”

Learning the Java Language:

http://docs.oracle.com/javase/tutorial/java/TOC.html

Read:

1. Object-Oriented Programming Concepts (ALL but “What is an Interface”)
2. Language Basics (ALL)
3. Classes and Objects (ALL stopping at Nested Classes)
4. Numbers and Strings (ALL)

OOP Review

- Objects have two characteristics: (1) state and (2) behavior
- An object stores its state in fields and exposes its behavior through methods
- data encapsulation is the hiding of the internal state of an object and the requiring of interaction to be performed through an objects methods

What is the state of a Rational number?

What is the behavior or a Rational number?

class - blueprint from which objects are created

Rational.java

```
1  package edu.pacificu.cs.Rational;
2
3  /**
4   * Creates a Rational class where all operations work on actual
5   * rational numbers of the form numerator / denominator. Assumes all
6   * Rationals are positive and there is no error checking.
7   *
8   * @author  CS, Pacific University
9   */
10
11
12
13 public class Rational
14 {
15   private int mNumerator;
16   private int mDenominator;
17
18   /**
19   * Initializes data members to default values representing 0
20   */
21
22   public Rational ()
23   {
24     mNumerator = 0;
25     mDenominator = 1;
26   }
27
28   /**
29   * Initializes Rational number using parameter list values
30   *
31   * @param numerator  the numerator of the Rational number
32   *
33   * @param denominator  the denominator of the Rational number
34   */
35
36   public Rational (int numerator, int denominator)
37   {
38     this.mNumerator = numerator;
39     this.mDenominator = denominator;
40   }
```

Inheritance - each class can “inherit” state and behavior from one superclass

Each superclass can have unlimited subclasses

```
35  class MovingSprite extends Sprite
36  {
37      //new fields
38
39      // new methods
40  }
```

Interface - collection of abstract methods

A class

1. implements an interface
2. inherits the abstract methods

An interface is NOT a class

```
42  interface Animal
43  {
44    public void eat ();
45    public void move ();
46  }
47
48  public Mammal implements Animal
49  {
50    private int mNumberOfLegs;
51
52    public void eat ()
53    {
54      System.out.println ("Mammal Eat");
55    }
56    public void move ()
57    {
58      System.out.println ("Mammal Move");
59    }
60    public void setNumberOfLegs (int numLegs)
61    {
62      mNumberOfLegs = numLegs;
63    }
64    ...
65  }
```

Interface similar to Class

1. can contain any number of methods
2. has .java extension
3. filename matches interface name
4. .class contains bytecode

Class different from Interface

1. cannot instantiate interface
2. cannot have constructors
3. all methods are abstract
4. cannot contain instance fields
5. not extended by a class
6. can extend “multiple” interfaces

package - is a namespace that organizes a set of related classes and interfaces

Conceptually think of packages as different folders on your computer where related classes and interfaces reside.

The Java API is broken up into many packages

1. java.lang - bundles the fundamental classes
2. java.io - classes for input and output

The Java 7 Platform API Specification can be found at:

http://docs.oracle.com/javase/7/docs/api/index.html