Introduction to Java

Clare So

socm@mcmaster.ca

SE2AA4 Tutorial. McMaster University. January 24, 2007.

What is Java?

- An object-oriented programming language
- Developed at Sun Microsystems
- Project started by James Gosling in 1991
- First released in January 1996
- “Write once, run everywhere”
- `http://java.sun.com`

Some Features of Java

- Object-oriented programming
- Exception handling
  - The program may not crash because of a run-time error
- Automatic garbage collection
  - No need to `free()` the memory manually
- Platform independence
  - Programs are compiled into bytecode
  - Bytecode is executed by the Java virtual machine

A First Java Program

- Let’s examine `Hello.java`

```
public class Hello {
   public static void main(String[] args) {
      System.out.println("Hi mom!  I can program in Java.");
   }
}
```

- Steps to compile and run `Hello.java`

```
[socm@birkhoff ~]$ javac Hello.java
[socm@birkhoff ~]$ java Hello
Hi mom!  I can program in Java.
[socm@birkhoff ~]$
```

A First Java Program

- The `.java` file must match the name of the class (ie. `Hello.java` must only include the `Hello` class)
- `public static void main` indicates the main function/procedure
- `System.out` is a built-in object in Java for handling outputs
- `println` is a built-in function/procedure of the `System.out` object

Creating Classes (Modules)

- Let's create a representation of a circle
- A circle only has its diameter as its attribute
- We can
  - Initialize the circle (constructor)
  - Grow/Shrink the circle (mutator)
  - Get the diameter (selector)
- Additionally, we can
  - Calculate the circumference of the circle
  - Create a string representation of the object

Our Circle Class (`Circle.java`)

```
public class Circle {

   private double diameter;  // The diameter of the circle

   // Default Constructor
   public Circle() {
      diameter = 10.0;
   }

   // Constructor
   public Circle(double r) {
      diameter = r;
   }
```

Our Circle Class (`Circle.java`)

```
   // Grow the circle by "factor" units.
   // The new diameter of the circle is returned.
   public double grow(double factor) {
      diameter = diameter + factor;
      return diameter;
   }

   // Shrink the circle by "factor" units.
   // The new diameter of the circle is returned.
   public double shrink(double factor) {
      diameter = diameter - factor;
      return diameter;
   }
```

Our Circle Class (`Circle.java`)

```
// Access the diameter of the circle
// without modifying it
public double getDiameter() {
   return diameter;
}

// Calculate the circumference of the circle
public double calculatetCircumference() {
   double PI = 3.14;
   return diameter*PI;
}

// Tell us how big the cirlce is!
public String toString() {
   return "The diameter of this circle is " + diameter + " cm.";
}
```

Using our Circle Class (`Test.java`)

- Here is how we test our newly created `Circle` class

```
public class Test {
   public static void main(String[] args) {

      Circle a = new Circle();

      System.out.println(a.toString());

      System.out.println("getDiameter gives us "+a.getDiameter);

      a.grow(2.4);
      System.out.println("After grow: "+a.toString());
```

Using our Circle Class (`Test.java`)

```
      a.shrink(5);
      System.out.println("After shrink: "+a.toString());

      System.out.println("The circumference of this circle is "
         + a.calculatetCircumference() + "cm.");
   }
}
```

Using our Circle Class (`Test.java`)

- Compiling and running `Test.java`

```
[socm@birkhoff ~]$ javac Test.java
[socm@birkhoff ~]$ java Test
The diameter of this circle is 10.0 cm.
getDiameter gives us 10.0
After grow: The diameter of this circle is 12.4 cm.
After shrink: The diameter of this circle is 7.4 cm.
The circumference of this circle is 23.236cm.
[socm@birkhoff ~]$
```