Introduction to Programming

UCM/MSP

Tutorial 6

1a. Given the following two classes (ignore their constructors):

```
public class Car
{
   public void m1()
   {
      System.out.println("car 1");
   }

   public void m2()
   {
      System.out.println("car 2");
   }

   public String toString()
   {

      return "vroom";
   }
}
```

```
public class Truck extends Car
{
   public void m1()
   {
      System.out.println("truck 1");
   }
}
```

What will be the output of the following code?

```
Truck mycar = new Truck();
System.out.println(mycar);
mycar.m1();
mycar.m2();
```

1b. If we add some extra methods to the Truck class:

```
public class Truck extends Car
{
   public void m1()
   {
      System.out.println("truck 1");
   }
   public void m2()
   {
      super.m1();
   }

   public String toString()
   {
       return super.toString() +
              super.toString();
   }
}
```

What will that change?

2. Assume that the following classes A, B, C and D have been defined. What will be the outcome of the code (below the classes)? Try them on paper first, then check your answers by running them.

```
public class A extends B
{
     public void method2()
     {
           System.out.println("a 2");
     }
}

public class D extends B
{
     public void method1()
     {
           System.out.println("d 1");
      }
}

public class C
{
     public String toString()
     {
           return "c";
     }
```

```
	public void method1()
	{
		System.out.println("c 1");
	}

	public void method2()
	{
		System.out.println("c 2");
	}
}

public class B extends C
{
	public String toString()
	{
		return "b";
	}

	public void method2()
	{
		System.out.println("b 2");
	}
}

// Somewhere in a class file (in a main method)
final C[] elements = {new A(), new B(), new C(), new D()};

for (int i = 0; i < elements.length; i++)
{
	System.out.println(elements[i]);
	elements[i].method1();
	elements[i].method2();
}
```

3. Animals

Download the file I2P_Animals.zip from Canvas and unzip it to a new Java project called I2P_Animals. This is the code for the Animals example in Lecture 6. Run AnimalsTest.java and you should see the following output:

```
Is a cow named Daisy. (moo)
Is a cow named Bessie. (moo)
Is a cow. (moo)
Is a snake. (hiss)
Is a snake named Monty. (hiss)
Is a lizard named Liz.
```

3a. Add an interface called `Swims` for animals that can swim. This interface can be empty; it does not require any constants or methods.

3b. Add two new concrete subclasses of animals: `Dolphin` and `WaterSnake`. Both are phylum Animalia and class Chordata, and both swim so should implement the `Swims` interface (as well as the `Noisy` interface). Add a couple of test objects to the `animals` array in `AnimalsTest` to test your new classes.

3c. In Animal.toString(), append " (swims)" to the text description of animals that swim. You can do a `(this instanceof Swims)` test to check whether an animal swims.

3d. Now add a class `Organism` that is a superclass of `Animal` and move the `String name` variable from `Animal` up to `Organism`. Make `Animal` extend `Organism`. You will also need to move the `kingdom(), phylum(), clss()` and `type()` methods up to `Organism`. Implement a mechanism for counting the number of organisms created, using a variable `private static int numOrganisms`. Why is this variable static?

3e. Add an abstract `Plant` class that is a subclass of `Organism`. This class represents members of the kingdom Plantae, and should perform similar functionality as `Animal` but for plants (you can copy `Animal` and just change the relevant class, variable and method names).

3f. Add two concrete subclasses of plants: `Oak` and `Cactus`. Both are phylum Magnoliophyta and class Magnliopsida. Add a couple of test objects to your array in `AnimalsTest` to test your new classes (you should change the array type and name to `Organism[] organisms`). Your output should now look something like the following:

```
Is a cow named Daisy. (moo)
Is a cow named Bessie. (moo)
Is a cow. (moo)
Is a snake. (hiss)
Is a snake named Monty. (hiss)
Is a lizard named Liz.
Is a dolphin named Flipper. (squeak) (swims)
Is a water snake. (hiss) (swims)
Is a oak.
Is a cactus named Spike.
```

Additional Tasks

Here are additional tasks to do at home. You are not expected to do this during the tutorial and they will not be marked.

4. What is the output of the following programs? Try them on paper first, then check your answers by running them.

4a.

```
class A
{
    int i = 10;
}

class B extends A
{
    int i = 20;
}

public class MainClass
{
    public static void main(String[] args)
    {
        final A a = new B();
        System.out.println(a.i);
    }
}
```

4b.

```
class A
{
    {
        System.out.println(1);
    }
}

class B extends A
{
    {
        System.out.println(2);
    }
}

class C extends B{
    {
        System.out.println(3);
    }
}

public class MainClass
```

```
{
    public static void main(String[] args)
    {
        final C c = new C();
    }
}
```

4c.

```
class A
{
    String s = "Class A";
}

class B extends A
{
    String s = "Class B";
    {
        System.out.println(super.s);
    }
}

class C extends B
{
    String s = "Class C";
    {
        System.out.println(super.s);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        final C c = new C();
         System.out.println(c.s);
    }
}
```