Software and Programming I

Week 9 Lab -
Use of Classes
and Inheritance

Tobi Brodie (Tobi@dcs.bbk.ac.uk)

Lab 9: Objectives

Exercise 1 – Student & StudentTest classes

1. Reinforce Code Writing
   - **i.** **Class** structure
   - **ii.** **Accessor** (getter) / **mutator** (setter) methods.
2. Override a method
3. Use of **array** to create multiple objects
4. Use of the **Enhanced For Loop** / **String** comparison

Exercise 2 – Person, Tutor, Student & Test classes

1. Class Inheritance
2. Code Revision / Refactoring

Coursework

You need to have completed, and have shown to a lab demonstrator **all six** marked exercises to be entered for the in-class-test in Week 11, 26th March. The exercises are as follows:

1. **`LeapYear`**
   Lab 2 Marked Exercise 1
2. **`Powers`**
   Lab 3 Marked Exercise 2
3. **`BalancedParentheses`**
   Lab 3 Marked Exercise 3
4. **`ArrayExercises.repeat`**
   Lab 4 Marked Exercise 4
5. **`CashRegisterP816`**
   Lab 7 Marked Exercise 5
6. **`Team/Game/GameTest`**
   Lab 8 Marked Exercise 6
   (at least 0.5 should be recorded by 26 March)

Exercise 1:
Classes Student and StudentTest

Implement a class **`Student`**

The class requires the following attributes: name, year of birth and programme of study

1. Write the **class** declaration
2. Declare **variables**
3. Create **constructor**
4. Write three new methods: **`String getName()`**, **`int getYear()`** and **`String getProgramme()`**
5. Override the method **`String toString()`** from the class **`Object`** (a superclass of any class)

Note: as there are no mutator methods, the data for instances must be supplied on creation (through the constructor).

Exercise 1: Classes Student and StudentTest (2)

Implement a test class **`StudentTest`**

The class **`StudentTest`** is required to:

1. Create an **array** of 10 Students.
2. Create **instances** and receive user input to provide data to each **`Student` object**

   Hint: Use a loop and a **`Scanner`** object

3. Once data input is complete, print out the names of students of the programme "BSc ISM" only. Use an **enhanced for loop** for this.

Structure of Student Class

```
/**  Student class  **/
public class Student
{
    /* private data */
    private String name;
    private int year;
    private String programme;

    /* Constructor */
    public Student(String name, int year,
                                  String programme)
    {
      /* to do: assign values to instance variables
          on object instantiation */
    }
```

Structure of Student Class (2)

```
/* methods (public interface) */
public String getName()
{
   /* to do: write the code to return name */
 }

public int getYear()
{
   /* to do: write the code to return year */
}

public String getProgramme()
{
  /* to do: write the code to return programme */
}
```

Structure of Student Class (3)

```
    public String toString()
    {
        return  "Student " + name +
                     " programme " + programme;
    }
}  // end of class Student
```

Testing Student Class

```
import java.util.Scanner;

public class StudentTest
{

    public static void main(String[] args)
    {

      Scanner input = new Scanner(System.in);
      /* to do: Write code to:

      1. Declare and create an array of 10 Students.

      2. Fill  up the array by creating 10 instances
      of students where the name, year of birth and
      programme of study are entered at the
      keyboard.
       */
```

Testing Student Class (2)

```
          /* to do: Write more code to use an enhanced
          for loop to print out names of students of the
          programme "BSc ISM" only.
          */

     }   // end of method main

}    // end of class StudentTest
```

Note: an example of the **enhanced for loop** can be found on slide 19 of the 'Inheritance & Subclasses' lecture slides in Week 8.

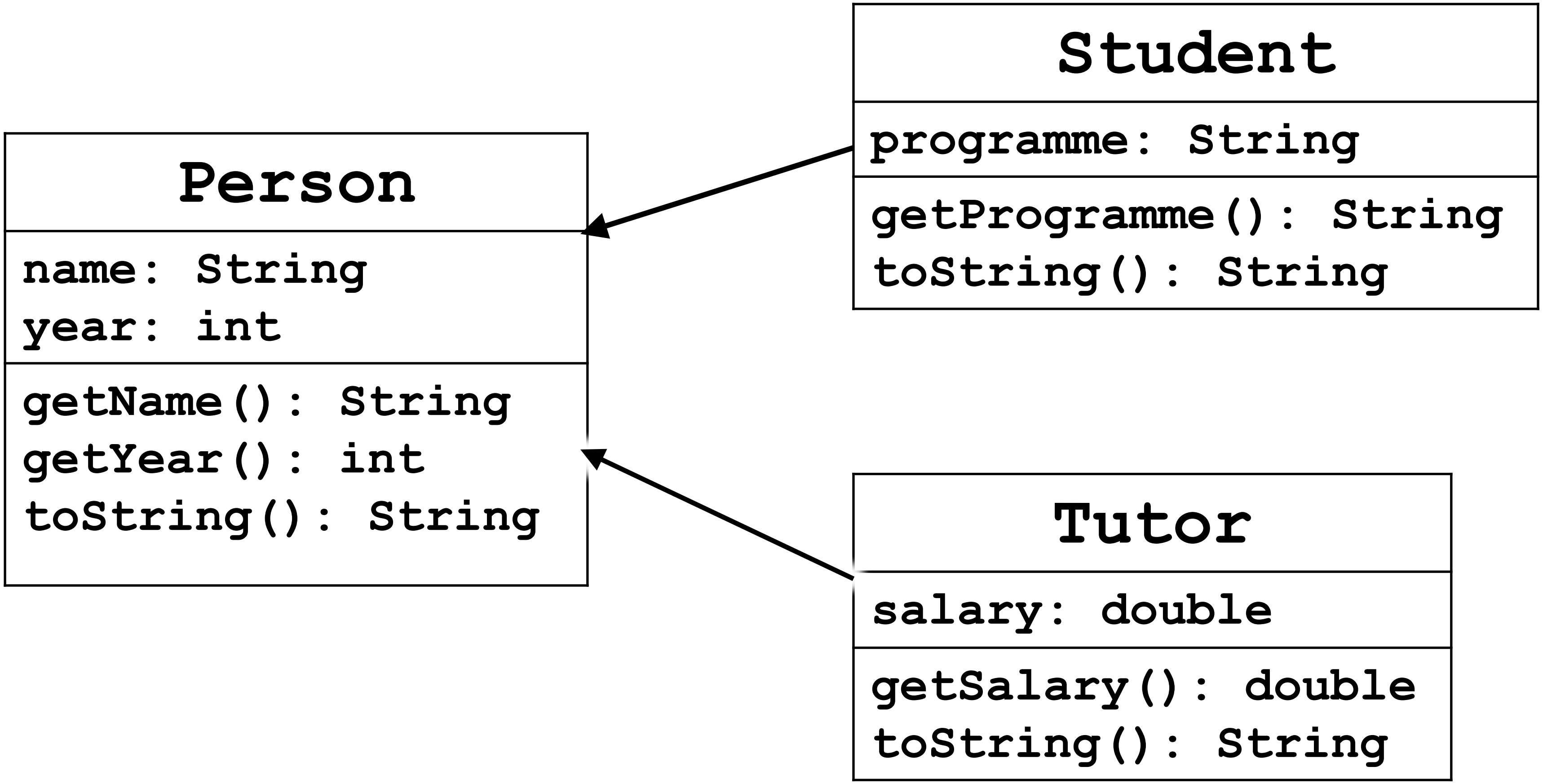
Exercise 2: Classes Person, Tutor, Student (revised) and Test

Implement a class **`Person`** (a person has a name and a year of birth). Change the class **`Student`** so that it *extends* **`Person`**. Then implement another subclass, **`Tutor`**, of **`Person`** (a tutor has a salary).

For the classes, write the class declarations, the constructors and the methods **`toString()`** for all classes.

Implement a program that tests these classes & methods.

Inheritance hierarchy



Revised Student Class

```
/**  Revised Student class  */
public class Student extends Person
{
     private String programme;

      public Student(String name, int year,
                                    String programme)
      {
          // call the constructor of Person
          super(name, year);
          this.programme = programme;
      }
```

Revised
Student Class (2)

```
     public String getProgramme()
     {
       /* to do: write the code to return programme */
     }

     public String toString()
     {
          return "Student " + getName() +
                 " programme " + programme;
     }

}  // end of class Student
```

Home Work
Java for Everyone by C. Horstmann

Read Sections 9.1–9.5, which are available online from

http://vufind.lib.bbk.ac.uk/vufind/Record/566484

and complete the following exercises:

- Exercise R9.10
- Exercise P9.12
- Exercise P9.21
- Exercise P9.22