Java SE 8 Programming Language

Lab Guides

Document Code	**25e-BM/HR/HDCV/FSOFT**
Version	**1.1**
Effective Date	**20/11/2012**

Hanoi, 06/2019

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	01/Oct/2018	Add the new labs	Create new	DieuNT1	VinhNV
2	12/May/2019	Update functional requirement	Update	DieuNT1	VinhNV
3	01/Jun/2019	Fsoft Template	Update	DieuNT1	VinhNV

Contents

CODE:	JPL.S.L202
TYPE:	SHORT
LOC:	200
DURATION:	60 MINUTES

Unit 3: Classes and Objects

Lab Guide 2: Inheritance, Encapsulation

Objectives: JPL-9

- ✓ Able to create Java-based applications that take advantage of Java object-oriented features, including encapsulation, inheritance, and polymorphism.

Problem Descriptions:

*This exercise will be developed from **JPL.S.L201** and adding an **Actionable** interface**,** an **EnglishTeacher** class**.***

Create a new package named **fa.training.entities** in **JPL.S.L202** project that contains:

The **Teacher** abstract class**:**

- ✓ Instance variables:
  - *designation*: for teacher designation
  - *collegeName:* the collegename that teacher do work
- ✓ Constructor:
  - public **Teacher**(): A default constructor, it should initialize the attribute to null or 0)
  - public **Teacher** (String designation, String collegeName): A constructor with parameters, it creates the teacher object by setting the two fields to the passed values
- ✓ Instance methods:
  - Getter/Setter methods: are used to get/set the value
  - public void teach(String content){}

The **MathTeacher** class that extends Teacher:

- ✓ Instance variables:
  - *mainSubject*: the main subject
- ✓ Constructor:
  - public **MathTeacher**(): A default constructor, it should initialize the attribute to null or 0)
  - public **MathTeacher** (String designation, String collegeName, String mainSubject): A constructor with parameters, it creates the teacher object by setting the three fields to the passed values.
- ✓ Instance methods:
  - Getter/Setter methods: are used to get/set the value
  - public void teach(String content){}: override the parent's method
  - public String toString(): This method allows the math teacher to be easily printed out to the screen

Create a new interface named **Actionable** inside package **fa.training.entities,** this interface contains an abstract method named **toSchool**(**),** Teacher class would implement the interface and override toSchool() method.

Create **EnglishTeacher** class inside **fa.training.entities** package, this class also extend above **Teacher** and implement **Actionable**. Add a new method **teach(int duration)** in **Teacher** class

Create package **fa.training.management** that contains **TeacherManagement** class:

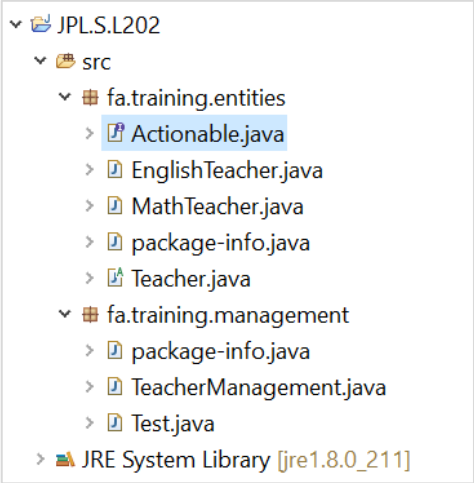
- ✓ Create some objects of MathTeacher and EnglishTeacher.
- ✓ Call methods of the class and explains the result.

Functional Requirements:

- ✓ Explain about snippet code: `(MathTeacher) teachers[i], (EnglishTeacher) teachers[i]` at lines of code 27, 29, 36, 38 and 42.
- ✓ Create some other objects Teacher from Actionable, call method and explains the results.

Guidelines:

Project struture:



- ✓ **Actionable interface**

```
1.  package fa.training.entities;
2.
3.  /**
4.   *
5.   * @author DieuNT1
6.   *
7.   */
8.  public interface Actionable {
9.      /**
10.      * An abstract method.
11.      */
12.     void toSchool();
13. }
```

- ✓ **Teacher** class

```
1.  package fa.training.entities;
2.
3.  public abstract class Teacher {
4.      private String designation;
5.      private String collegeName;
```

```
6.
7.      public Teacher() {
8.      }
9.
10.     public Teacher(String designation, String collegename) {
11.         super();
12.         this.designation = designation;
13.         this.collegeName = collegename;
14.     }
15.
16.     public String getDesignation() {
17.         return designation;
18.     }
19.
20.     public void setDesignation(String designation) {
21.         this.designation = designation;
22.     }
23.
24.     public String getCollegename() {
25.         return collegeName;
26.     }
27.
28.     public void setCollegename(String collegename) {
29.         this.collegeName = collegename;
30.     }
31.
32.     public abstract void teach();
33.
34.     public void teach(int duration) {
35.         System.out.println("Teaching in " + duration + " minutes");
36.     }
37.
38. }
```

✓ **MathTeacher** class

```
1.  package fa.training.entities;
2.
3.  /**
4.   *
5.   * @author DieuNT1
6.   *
7.   */
8.  public class MathTeacher extends Teacher implements Actionable  {
9.      protected String mainSubject;
10.
11.     public MathTeacher() {
12.     }
13.
14.     public MathTeacher(String designation, String collegename,
15.                                                         String mainSubject) {
16.         super(designation, collegename);
17.         this.mainSubject = mainSubject;
18.     }
19.
20.     public String getMainSubject() {
21.         return mainSubject;
22.     }
23.
24.     public void setMainSubject(String mainSubject) {
25.         this.mainSubject = mainSubject;
26.     }
27.
```

```
28.     /**
29.      * The method return sum of all two numbers.
30.      *
31.      * @param number1
32.      * @param number2
33.      * @return an integer value.
34.      */
35.     public int sum(int number1, int number2) {
36.         return (number1 + number2);
37.     }
38.
39.     @Override
40.     public void toSchool() {
41.       System.out.println("Math teacher go to school by car!");
42.     }
43.
44.     @Override
45.     public void teach() {
46.         System.out.print("Teaching math subject:");
47.     }
48.
49.     @Override
50.     public String toString() {
51.         return "MathTeacher [mainSubject=" + mainSubject +
52.              ", getDesignation()="+ getDesignation() +
53.              ", getCollegename()=" + getCollegename() + "]";
54.   }
55. }
```

✓ EnglishTeacher **class**

```
1.  package fa.training.entities;
2.
3.  /**
4.   *
5.   * @author DieuNT1
6.   *
7.   */
8.  public class EnglishTeacher extends Teacher implements Actionable {
9.
10.     private String mainSubject;
11.
12.     public EnglishTeacher() {
13.     }
14.
15.     public EnglishTeacher(String designation, String collegename,
16.                                                 String mainSubject) {
17.         super(designation, collegename);
18.         this.mainSubject = mainSubject;
19.     }
20.
21.     public String getMainSubject() {
22.         return mainSubject;
23.     }
24.
25.     public void setMainSubject(String mainSubject) {
26.         this.mainSubject = mainSubject;
27.     }
28.
29.     @Override
30.     public void teach() {
```

```
        System.out.println("Teaching English subject");
    }

    @Override
    public void toSchool() {
        System.out.println("English teacher go to school by motorbike");
    }

    public String translate(String en, String vi) {
        return en + " in Vietnamese " + vi;
    }

    @Override
    public String toString() {
        return      "EnglishTeacher [mainSubject=" + mainSubject +
                    ", getDesignation()=" + getDesignation() +
                    ", getCollegename()=" + getCollegename() + "]";
    }
}
```

✓ **TeacherManagement** class

```
package fa.training.management;

import fa.training.entities.EnglishTeacher;
import fa.training.entities.MathTeacher;
import fa.training.entities.Teacher;

public class TeacherManagement {

  public static void main(String[] args) {

    MathTeacher mathTeacher = new MathTeacher("Teacher", "FU", "Math");
    MathTeacher mathTeacher2 = new MathTeacher("Teacher", "PTIT", "Math");
    EnglishTeacher englishTeacher = new EnglishTeacher("Teacher", "PTIT", "English");

    Teacher[] teachers = new Teacher[3];
    teachers[0] = mathTeacher;
    teachers[1] = mathTeacher2;
    teachers[2] = englishTeacher;

    int number1 = 100, number2 = 20;

    for (int i = 0; i < teachers.length; i++) {
      System.out.println("----------TEACHER " + (i + 1) + "-------------");
      System.out.println("Colleage name: " + teachers[i].getCollegename());
      System.out.println("Designation: " + teachers[i].getDesignation());
      if (teachers[i] instanceof MathTeacher) {
        System.out.println("Main subject: " + ((MathTeacher) teachers[i]).getMainSubject());

        ((MathTeacher) teachers[i]).toSchool();

        teachers[i].teach();
        System.out.println("SUM(" + number1 + ", " + number2 + ") = " +
                                  mathTeacher.sum(number1, number2));

      } else {
        System.out.println("Main subject: " + ((EnglishTeacher) teachers[i]).getMainSubject());

        ((EnglishTeacher) teachers[i]).toSchool();

        teachers[i].teach();

        ((EnglishTeacher) teachers[i]).translate("Hello", "Xin chao!");
      }

    }

  }
}
```

- ✓ How to run:

  Click **Run** menu | choose **Run as:**

Results:

```
----------TEACHER 1-------------
Colleage name: FU
Designation: Teacher
Main subject: Math
Math teacher go to school by car!
Teaching math subject!SUM(100, 20) = 120
----------TEACHER 2-------------
Colleage name: PTIT
Designation: Teacher
Main subject: Math
Math teacher go to school by car!
Teaching math subject!SUM(100, 20) = 120
----------TEACHER 3-------------
Colleage name: PTIT
Designation: Teacher
Main subject: English
English teacher go to school by motorbike
Teaching English subject
```

-- THE END --