

TP1 on Java Programming

Qiong Liu

September 2024

1 Introduction

The objective of the first lab is to install Java on the laptop, get familiar with first example, class, object, review class diagram from first course.

The requirement of your report:

- Submit everything in a .zip file named: `JAVA_TP1_prenom_nom.zip`;
- Include a .pdf file to answer open questions, such as those in Section 3, questions 1 and 2;
- Name your project folder `TP1_Prenom_Nom`, and inside this project folder, include the following:
 - The following files: `.idea`, `.gitignore`, `.iml`, `out`, and `src` (your code should be inside the `src` folder);
 - Write comments in your code specifying which question you are answering, e.g., `section2-question3-a`;
 - Inside the `src` folder, include:
 1. `hello_world.java` [10%]
 2. A package called `Geometry`, containing the following classes [40%]:
 - 1) `Point.java`
 - 2) `Point_advanced.java`
 - 3) `Triangle.java`
 - 4) `Rectangle.java`
 3. A package called `University`, containing the following classes [40%]:
 - 1) `Student.java`
 - 2) `Teststudent.java`

2 Installation of Java compiler

2.1 Option 1: IntelliJ IDEA

All operating systems can install it.

1. Download integrated development environment (IDE) from <https://www.jetbrains.com/idea/>.
2. Download JDK 22 from <https://www.oracle.com/java/technologies/downloads/#java22>

2.2 Linux/MacOS

- Check if your laptop already have Java installed or not. in the terminal, type `'java -version'`
- Otherwise, install is from website <https://www.oracle.com/java/technologies/downloads/#jdk22-mac>

3 First example: Hello World!

Follow the steps to create your first java program "Hello World!":

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("Hello World !") ;  
    }  
}
```

3.1 Option 1 IDEA:

- Create new project "hello".
- Create new class "Hello" in the folder 'src'
- type the following code in the new created class file

3.2 Option 2 Terminal:

1. Go to the "Documents" folder.
2. Create a working folder for your source files (".java"), named "tpjava". Note: the "classes" folder will be used by the compiler, your programs (sources) with the ".java" extension should be placed in the "Documents" subfolder named "tpjava".
3. Verifying your configuration (step 1): Edit, compile and run the "Hello World" program (Hello.java file):
4. Note: You will use the terminal for this first simple program.

```
javac Hello.java  
java Hello
```

Questions:

1. If `println` is replaced by `print`, what can you get. What is the difference between `print` and `println`?
2. Does this class has methods or attributes? If yes, list them.
3. Print the following texts (pay attention to changing new lines):

```
Hello world,  
My name is XX.  
What is your name?
```

4 Second example: Points

Create new package "Geometry" in the folder "src".

Create a new class named Point, and type the following code in the Point.java file.

```
public class Point {  
    private int x;  
    private int y;  
  
    public Point(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
}
```

```

//Constructor, to initialize values of x and y
public Point() {
    x = 0;
    y = 0;
}

public String toString() {
    return "Point [x=" + x + ", y=" + y + "]";
}

public int getX() {
    return x;
}

public int getY() {
    return y;
}

public void setX(int x){
    this.x = x;
}
public void setY(int y){
    this.y = y;
}

public static void main(String[] args) {
}
}

```

Questions:

1. what is the purpose of this program?
2. Describe the class diagram from the given code, write the names of attributes and methods it contains.
3. In the `main` method,
 - (a) create a new instance using command:
`Point point = new Point(1, 1);`
 - (b) What will we get if we don't pass any values when creating new instance?
 - (c) Use method `getX()` and `getY()` to obtain value of `x` and `y` from the class `Point`. calling a method's format as follows:
`instance.method()`
 - (d) Use `System.out.println` to display the results returned from method `toString()`.
 - (e) Use method `setX()` and `setY()` to set `x=100`, `y=100`. Then verify it using method `toString()`.
4. What is the purpose of using `this`. What if we don't use `this`?
5. Advanced: Each point instance lies on a finite plane, with limit $-300 \leq x \leq 300$ and $-300 \leq y \leq 300$. Use `if` and `else` to successfully create instance or print "errors" respectively. (you can create a `Point-advanced` class)

5 Triangle

Let's consider the case of triangle. Create a new class file named "Triangle.java" under the same project "Geometry".

```

public class Triangle {
    private Point point1;
    private Point point2;
    private Point point3;

    public Triangle(int x1, int y1, int x2, int y2, int x3, int y3) {
        point1 = new Point(x1, y1);
        point2 = new Point(x2, y2);
        point3 = new Point(x3, y3);
    }

    public Triangle(Point point1, Point point2, Point point3) {
        this.point1 = point1;
        this.point2 = point2;
        this.point3 = point3;
    }

    public String toString() {
        return "Triangle [point1=" + point1.toString() + ", point2=" + point2.
            toString() + ", point3=" + point3.toString() + "]";
    }

    public Point getPoint1() {
        return point1;
    }

    public void setPoint1(Point point1) {
        this.point1 = point1;
    }

    public static void main(String[] args) {
        Triangle t1 = new Triangle(0,0,1,0,2,3);
    }
}

```

Questions:

1. What is relationship between Point class and Triangle class?
2. Create new method to set new values for three vertices of the triangle.

```

public void move(int x1, int y1, int x2, int y2, int x3, int y3) {
    point1.setX(x1);
    point1.setY(y1);

    point2.setX(x2);
    point2.setY(y2);

    point3.setX(x3);
    point3.setY(y3);
}

```

3. Test the following methods in main method, `toString()`, `getPoint1()`, `setPoint1()`,
4. **Advanced:** Create new class "Rectangle", triangle can be created by defining its four vertices coordinates.

6 Class "Student"

Create new package "University" in the folder "src".

We now want to define a "Student" class allowing us to represent the main characteristics of a student in a university's registration system. We will only consider the following information: last name, first name, student number, telephone number and email (university email address).

1. Define the UML class diagram of such a class.
2. Implement the "Student" class as well as a separate test class.
3. Add the packages.
4. You will make sure to specify the useful accessors(getter) and mutators(setter) as well as the different constructor variants.
5. Make useful improvements to your code ("toString()" method, use of "this." and "this()" and "this", non-redundancy of code, control of inputs at the mutators level).

Important: you will make sure to respect the principle of encapsulation (levels of accessibility to attributes and methods) and the presentation of your programs (indentation and comments).