Travaux Dirigés n°1

Java Avancé

—M1—

OOP essentials

Objects, equality, encapsulation, mutability, overriding methods, etc.

▸ Exercice 1. Eclipse

1. Create a new project called `JavaAvance` and add a new package called `fr.dauphine.javaavance.td1`.
2. What happens when you type sysout and press Ctrl + space in a main method ?
3. Same question with toStr then Ctrl + space inside a class ?
4. Same question with main then Ctrl + space inside a class ?
5. Create a new `int` field called `foo` inside the class. What happens if you type Ctrl + space inside the class, what if you now type set then press Ctrl + space ?
6. Select the class name. Que What happens if you type Alt+Shift+R ? Same question with the int field `foo`.
7. (at home) It is sometime useful to check the source code from the JDK. Download the file called src.zip from Oracle's website. (on CRIO desktop stations it is already installed in /usr/local/jdk***/src.zip) and attach it in Window - Preferences – Installed JREs - Edit - rt.jar - Source Attachment. Now declare a String variable and clic on String with Ctrl key pressed. What happens ?

▸ Exercice 2. Point

The goal here is to specify a `Point` class to represent Cartesian coordinates.

1. Create a new class `Point` with two **private** fields x and y. Add a method with the following code :

```
1 Point p=new Point();
2 System.out.println(p.x+" "+p.y)
```

   Why does it work ?

2. Create a class `TestPoint` with a main and the same code as before. What happens ? How can we fix it ?

3. Why do you need to set all fields visibility to private ?
4. What is an accessor ? Do we have to do it here ?
5. Create a constructor with two arguments (called `px` and `py`). What is the problem ?
6. Modify the parameters of the constructor to call them `x` and `y`. What happens ?
7. We would like to keep track of the number of Points that have been created so far. How to proceed ?
8. Write a second contructor with a single `Point p2` argument that copies the coordinates from `p2` into the current Point. How does the compiler know which constructor to call ?
9. Update the class so that a call to `System.out.println(point);` will print the point coordinates as follows : (x, y).

▸ Exercice 3. Equality

We use the `Point` class from the previous exercise.

1.
```
Point p1=new Point(1,2);
Point p2=p1;
Point p3=new Point(1,2);

System.out.println(p1==p2);
System.out.println(p1==p3);
```

   What does this code print ? Why ?

2. Write a method `isSameAs(Point)` that will return true if the two points have the same coordinates.

3.
```
Point p1=new Point(1,2);
Point p2=p1;
Point p3=new Point(1,2);

ArrayList<Point> list = new ArrayList<>();
list.add(p1);
System.out.println(list.indexOf(p2));
System.out.println(list.indexOf(p3));
```

   What is the problem with this code ? Read the documentation of `indexOf` and check which method is called. Modify the `Point` class to fix this problem.

▸ Exercice 4. Polyline

We use the class `Point` from the previous exercise. We now want to write a class to represent a polygonal line. The polyline will have a maximum number of points that can vary from one instance to another (but is a constant for a single polyline instance).

1. You will use an array to store the Points of the polyline. Write the constructor for PolyLine.
2. Write a method `add` that can be used to add a new point to the line. What happens if we add more points that the maximum capacity of the array ? What to do about it ?

3. Write a method `pointCapacity()` and `nbPoints()` that will return the maximum capacity of the polyline and the number points currently in the polyline.
4. Write a method `contains` which will return true if a given point is in the polyline. Use a **for each** loop to do this (instead of a classical index based loop).
5. What happens if `null` is given instead of an actual Point object ? What if you do `add(null)` before ? Read about `Objects.requireNonNull(o)`.
6. Update the class and use a `LinkedList` instead of an array (and remove the maximum capacity limit). How to update `pointCapacity, nbPoints` and `contains` ?

▸ Exercice 5. Mutability and circle

1. Add a method `translate(dx, dy)` in `Point`. What are the different options to write this method ?
2. A circle can be represented with a center and a radius. Write a new `Circle` class. Don't forget the constructor.
3. Write the `toString` method.
4. Write the `translate(dx, dy)` that translate the circle.
5.
```
Point p=new Point(1,2);
Circle c=new Circle(p,1);

Circle c2=new Circle(p,2);
c2.translate(1,1);

System.out.println(c+" "+c2);
```
   What is the problem with this code ? How to avoid it ?
6. What would be the problem with a `getCenter()` method that would return the center ? To find out, consider the following code ?
```
Circle c=new Circle(new Point(1,2), 1);
c.getCenter().translate(1,1);
System.out.println(c);
```
   Modifier pour que cela soit correct.
7. Add `area()` and update toString() to print the area as well.
8. Add a `contains(Point p)` method to return true if p is inside the circle (hint : use Pythagoras theorem).
9. Add `contains(Point p, Circle...circles)` that will return true of the point is inside one of the circles ? What other change should you do about the method declaration ? Why ?

▸ Exercice 6. Anneaux

We now want to defined a ring.

1. Should you use inheritance ?
2. Write a new class `Ring`, with a center and two radius (beware, the inner radius must always be smaller than the outer one.
3. Write `equals`.
4. On veut afficher un anneau avec son centre, son rayon et son rayon interne. Quel est le problème si on fait `System.out.println(ring);` sans code supplémentaire ? Le corriger.
5. Write `contains(Point)`, avoid useless object construction.
6. Write `contains(Point p, Ring...rings)`.