Homework 3

Static Methods

Do this homework for attendance at the third laboratory session

Exercise 1

Write with Notepad a program `Password` in your directory. The program generates a password from your first and last name, and your birthday. The program carries out the following operations:

1. Asks the user to insert her/his first name (i.e. John).
2. Asks the user to insert her/his second name (i.e. Walker).
3. Asks the user to insert her/his birthday with a format `dd/mm/yyyy` (i.e. 12/04/1984).
4. Print using `System.out` a message which informs the user on the generated password (i.e*. Hello, John Walker! You password is: jo84er*).

Note.

The password is calculated by concatenating the first two letters of the first name, the last two digits of the birthday year and the last two letters of the second name, all of them in lower cases.

Exercise 2

Modify the previous program in a new program `Password1` which generates the password from the user's first and last names, and birthday. The program carries out the following operations:

1. Asks the user to insert her/his first name (i.e. John).
2. Asks the user to insert her/his second name (i.e. Walker).
3. Asks the user to insert her/his birthday with a format `dd/mm/yyyy` (i.e. 12/04/1984).
4. Print using `System.out` a message which informs the user on the generated password (i.e*. Hello, John Walker! You password is: jo21er*).

Note.

1. The password is calculated concatenating:
   a) the first two letters of the first name,
   b) the age calculated as a difference of the current year (use GregorianCalendar) and the birthday year,
   c) the last two letters of the last name, all in lower case.

2. To convert a string in an integer (if the string is a sequence of digits) use the method:

```
int Integer.parseInt(String s)
```

where s is the string to be converted (i.e. `Integer.parseInt("1984")` has as a result `1984`).

Exercise 3

Write with Notepad a program `Tickets` in your directory. The program calculates the total price of several train tickets. The program carries out the following operations:

1. Asks the user for the number of tickets (i.e. `3`).
2. Asks the user for the cost of a ticket in lei (i.e. `34.85`).
3. Write using `System.out` a message which informs the user of the total cost of the tickets (in this case: `104.55 lei`).

Exercise 4

Write a program `Distance1` in your directory. The program uses the class `java.awt.Point` (you must import it!):

1. To create two objects Point: A(6, 4) and B(10, 7).

2. To calculate the distance between the two points with the formula:

$$dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the points,

3. To display the result.

Exercise 5

Write a version of the previous program, `Distance2`, in your directory. The program calculates the distance between the two points with a static method `distance()` which provides as a result a `double` value. The method declaration has the next structure:

```
public static double distance(int x1, int y1, int x2,
int y2){
   . . .
}
```

Exercise 6

Write a new version of the last two programs, `Distance3`, which uses the following method

```
public static double distance(Point p1, Point p2){
   . . .
}
```

Note differences in how the method parameters are passed between `Distance2` and `Distance3`.

Exercise 7

Let write our first program having more than one class. Follow the following steps:

1. Write and save in your directory the following class `Geom` containing five static methods:

```
public class Geom {
  public static double distance(Point p1, Point
p2){
 . . .
  }
  public static double triangleSemiPerimeter(Point
a,Point b,Point c){
 . . .
  }
```

```
  public static double triangleArea(Point a,Point
b,Point c){
 . . .
  }
  public static double rectanglePerimeter(Point a,
Point b){
 . . .
  }
  public static double rectangleArea(Point a, Point
b){
 . . .
  }
}
```

Hint: Use Heron's formula for the area of a triangle:

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

where a, b, c are the triangle edges and s is its semiperimeter.

2. Write and save in your directory a class `GeomTest` having a method `main` which:
   - creates three objects `Point` located in the plane;
   - print their distances to the origin;
   - print the area of the triangle formed by the 3 points;
   - print areas of the three rectangles formed by each pair of points.

   `GeomTest` is structured as in the followings:

```
public class GeomTest {
  public static void main(String[] args){
   . . .
  }
}
```

   Use in the `main()` algorithm the static methods of `Geom` by calling them with the class name:

`Geom.distance(..)`, `Geom.traingleSemiPerimeter(..)`, `Geom.triangleArea(..)` and so on.

3. Compile each of the two classes in the your directory.

```
javac Geom.java

javac GeomTest.java
```

   Two files `Geom.class` and `GeomTest.class` should be found in the directory.

4. Execute the class `GeomTest` with:

```
java GeomTest
```

Our class `Geom` is a utility class now ready to be used in geometrical applications.