BASE JAVA EXAMINATION

Goal of this paper

This paper is aimed to verify the level of understanding of the Java Base Course of the students. These exercises must be attempted from the student only after the final exercise of the Java Base Course (i.e. `Sample12`), without any time constrain.

Each exercise gives some points if the student manages to solve it: only students being able to reach **6 points** will pass the examination and will be ammitted to the Expert Java Course.

Exercise 1 (3 points)

Write the `Moving` applet, which shows a label on the applet's surface and 5 buttons: "Up", "Down", "Left", "Right" and "Where". Pressing the first four buttons the label will move in the direction corresponding to the button. To move the label use the method `setLocation()`, which must be implemented into a method named `drawIt()`.

Pressing the "Where" button will show on the applet's surface the X-Y coordinates of the label itself.

Exercise 2 (1 points)

Write the `Moving02` applet by extending the "Moving" applet (see exercise 1). This new applet must override the method `drawIt()` of the "Moving" applet since it will move the text using the `drawString()` method instead of the `setLocation()` method.

Exercise 3 (2 points)

Write the `Handling` application and other two very simple classes (name these classes `Foo1` and `Foo2`). The "Handling" application must invoke the method `doIt()` in the class "Foo1", which in turn must invoke the method `readNow()` in the class "Foo2": this method reads a key from the keyboard and returns it.

The "Handling" application must handle all the possible `Exceptions` that could be thrown by the `readNow()` method, by using a sequence of `try` and `catch` statements. For each possible exception the "Handling" application must print on the screen a different message and the stack trace for the exception (use the `printStackTrace()` method).

Hint: you must throw the exception in "Foo2" using the `throw new` clause to test this application.

Exercise 4 (4 points)

Write the `MyPrinter` application. This application contains the `PrintNow()` method, which prints a short text on the printer (at least three lines of text).

The `PrintNow()` method will do something like the following:

- Instanciates a `PrinterJob` object from the package java.awt.print.
- Instanciates a `Book` object from the package java.awt.print.
- Instanciates a `Painter` object (which is provided together with this exercise).
- Appends the "painter" object to the "book", and send the "book" object to the "printerjob" object.
- Shows the printer dialog on the screen and then prints the text on the printer.