DEEC
DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA E DE COMPUTADORES
TÉCNICO LISBOA

Object Oriented Programming, 2014/05/12

Laboratory evaluation - Duration: 45 min

Machine name _____

Student number _____ Student name _____

You must:

- Provide proof of your identity.
- Only have on your desk your identification and a pen.
- Ensure mobile phones or any other electronic devices are switched off and placed with personal belongings apart from the desk. Any student who does not switch off its phone, or who retains one in its possession, will be disqualified.
- Do not attempt to use the internet. The internet in your computer is off.
- Do not attempt to use any USB pen, to put/take info in/from your computer.
- **Do not turn this evaluation sheet before instructed**.

Examination instruction:

- The evaluation consists in a program to be developed within a package named

  lab2_NUMBER

  where NUMBER is the student number (e.g.: student number 12345 should create a package named lab2_12345 and provide java classes inside this package). This package must be found inside an Eclipse project named `LAB2`. Before stating ensures everything is in place (`/home/poo/workspace/LAB2/src/lab2_NUMBER`).
- When the evaluation ends leave the computer on and in session, that is,

  DO NOT LOGOUT,

  and **leave the examination sheet on the desk**.
- Skim all the evaluation before beginning to best plan your time.
- If the program does not compile, or crashes, it will be evaluated to a maximum of 0.6 points (out of 2.0 points).

The only classes you can use from `java.util` package are: `Iterator` and `Arrays`.

The program should implement the following functionalities:

[0.1] • Create an Eclipse project named LAB2. Inside this project create a package lab2_NUMBER, where NUMBER is your student number. Confirm that you have the following directories: `/home/poo/workspace/LAB2/src/lab2_NUMBER`.

[0.2] • Inside this package add a public generic class named `IterableArray<T>` that stores an array of objects of type `Object` named `array`. This `array` must have package visibility and must be built within a public constructor receiving the size of the array as parameter.

[0.2] • In `IterableArray` class, provide a public `add` method that adds an element of type `T` to the array. The array should behave as follows: (i) if the array is empty then all entries store the null object; (ii) if the array stores k objects then the indexes $0..k-1$ are used to store them and the remaining indexes store the null object; (iii) if the array is already full and `add` is invoked then an user exception named `IAException` is thrown.

[0.2] • The `IterableArray` must implement the `Iterable` interface.

> The `Iterable<T>` interface just provides the method:
>
> ```
> public Iterator<T> iterator();
> ```

[0.6] • Provide the required iterator in a public generic `IAIterator` class. If an object is removed from the array at an index k then objects at index r such that $k < r <$ `array.lenght` are moved to index $r-1$, and the last index of the array store the null object. Moreover, in the `remove` method, if the `next` method has not yet been called, or the `remove` method has already been called after the last call to the `next` method, an `IllegalStateException` should be thrown.

> The `Iterator<T>` interface provides three methods:
>
> ```
> public boolean hasNext()
> public T next()
> public void remove()
> ```

[0.1] • Provide a `Main` class with a `main` method. In the `main` method:

[0.1] – Provide an `IterableArray` of strings and add to it the following strings: “Hello”, “Iterable”, “Array”, “!”.

[0.1] – Using the iterator methods, iterate through the `IterableArray` and print its contents to the terminal.

[0.2] – Using the iterator methods, remove all elements from the `IterableArray` and print its contents to the terminal.

[0.2] • Provide an executable .jar file of your application with sources and place it in the following directory: `/home/poo/workspace/LAB2/src`.