Object-Oriented Programming
Laboratory 4

Exercise 1 (from Tutorial 3): Arrays

Question 1.1: Implement a matrix class that allows to manipulate two-dimensional matrices. Your class should implement:

- the initialization of a cell,
- the display of the full matrix,
- the addition of two matrices,
- the multiplication of two matrices (when possible).

Question 1.2: Write a class to manipulate a triangular data structure of the following shape:
#
##
###
A cell is accessed using (row,column). With this data structure, it is possible to access to cells (3,3) and (3,2), but not to cell (2,3). Ensure that you don't allocate unneeded cells.

Exercise 2 (from Tutorial 3): Lists

The main class to manipulate lists in Java is java.util.Vector. Among the most used methods of Vector are:

- void add(Object o);
- boolean contains(Object o);
- void remove(Object o);
- int size();

Question 2.1: How to add 1, 2 and 3 in the Vector? How to obtain these integer values from the Vector?

Question 2.2: Implement your own class in order to implement lists such as Vector does.

Question 2.3: Implement a reverse function, that reverses the list. For example, with this function, the list (a,b,c,d) becomes (d,c,b,a).

Exercise 3 (from Tutorial 3): Associative arrays

Associative arrays are arrays where the index is not an integer (unlike usual arrays), but an object. The main class in Java to manipulate associative arrays is java.util.Hashtable. A hashtable is implemented using an array of lists. A pair (key, value) is stored within the list of cell hashcode(key). Notice that in Java, there is a method "int hashcode()" in the Object class.

Among the most used functions of Hashtable, there are:

- void put(Object key, Object value);
- boolean contains(Object key);
- Object get(Object key);

Question 3.1: Implement your own class to manipulate associative arrays, such as Hashtable does. You can use the Vector class.

Question 3.2: Is it possible to modify your class so that there is a method “void put(Object key, int value)”? How to make the difference between an int as a value, and an Integer? Is it possible to have a method “int get(Object key)” used only when the value is an int, but not an Integer?

Exercise 4: Comparison of implementations

It is possible to store data in several structures. The choice of a good data structure has an impact on the performance of a program. Thus, we will compare the efficiency of several data structures on a simple example.

We will consider a set of 100 students (or more), each having a student number and a mark from 0 to 20. We will compare three data structures available in Java: arrays, vectors (from java.util) and hashtables (from java.util).

Question 4.1: Create the classes you need. Add the constructors and the accessors.

Question 4.2: Insert 100 randomly-generated students (using java.util.Random) in the three data structures, so that each data structure contains the same students, in order to make the comparison fair.

Question 4.3: Search in the Java documentation the class java.lang.System, which allows to obtain the current time (up to the millisecond).

Question 4.4: For each data structure, measure the time required to insertion 100 students from an empty data structure. Do the same with 1000 students, and with 10000 students.

Question 4.5: Write a method that allows the computation of the average mark of the students, for each data structure.

Question 4.6: Write a method to determine whether a given student ID exists in the data structure or not. Measure the time it takes for each data structure.