

Lab Assignment 3: Packages, Arrays and Sorting

Course: Advanced Programming (EE423)

Institution: IGEE, Boumerdes University

October 28, 2022

1 Objectives

Within this lab, you will, in the first part, learn to organize your classes by putting them inside packages and, in the second part, you will learn to use arrays and matrices to solve different problems. You will also be introduced to sorting where you will be coding an algorithm called Sort by selection.

2 Assignment

2.1 Exercise 1: Understanding packages

Create a structure of packages and create the classes in those packages as illustrated in the following figure:

```
pkg1_vehicle
|-- Car.java
|-- Motorcycle.java
|-- pkg2_Air
    |-- Helicopter.java
    |-- Plane.java
```

1. What is the import statement that must be added in the class Helicopter in order to use the class Car?
2. What is the package statement that must be added in the class Helicopter?
3. Add the following function inside the class Plane in Plane.java

```
public void print(){
    System.out.println("I am a plane");
}
```

4. Add a main function inside the class Car in Car.java and create an object plane of type Plane and call the function print() of the object plane:

```
Plane plane = new Plane();
plane.print();
```

5. Add the following function inside the class `Motorcycle` in `Motorcycle.java`

```
public void print(){
    System.out.println("I am a motorcycle");
}
```

6. Add a `main` function inside the `Helicopter.java` file and create an object of type `Motorcycle` and call the function `print()` of `Motorcycle`
7. Using the command line interpreter, compile and run the class `Car`, you should get the output: `I am a plane`
8. Using the command line interpreter, compile and run the class `Helicopter`, you should get the output: `I am a motorcycle`

2.2 Exercise 2: Arrays, Methods and Sorting

Do you remember the `main` function?

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

It takes as an argument an array of strings that contains the parameters we pass when we call it. For example, when we run a program

```
java Exercise2 Good morning everybody
```

The method `main` of the class, receives 3 elements:

```
"Good", "morning", "everybody"
```

These are stored in an array structure.

Notes:

- In Java, to get the length of an array `args`, you can use this line of code:
`args.length`
- To compute the sum of an array `tab` you can use this code:

```
int sum = 0;
int i;
for(i = 0; i < tab.length; i++) {
    sum = sum + tab[i];
}
// this loop does this: sum = tab[0] + tab[1] + ... + tab[i - 1]
```

- We represent a set of integers using arrays. In Java, an array of `n` items can be declared as follows:

```
int[] array;//allocate a reference to an array
array = new int[n];//allocate n int in memory and assign it to array
NB: n is a number that must be known beforehand.
```

- To initialize an array, you can use either way:

```
int [] array2 = new int[]{0,1,2,3,4}; // this
int [] array3 = {0,1,2,3,4}; // or this
```

1. Write a function called `static String ping(String[] parms)` that returns a string of all the parameters passed to it separated by commas in the reverse order. If no parameters are passed, return `No parameters passed`.
2. Write a method called `readPositiveInteger` that asks the user to input a positive integer value and returns this value. This function should accept only positive integers (excluding 0). If the user inputs 0 or a negative value, it will be prompted again to input a positive number.
3. Write a method called `readIntegers` that takes an `int n` as parameter, and asks the user to input `n` positive integers (excluding 0), negatives values and zero should not be accepted. It stores the values in an array, and returns this array. This method must use the previous one `readPositiveInteger`
4. Write a function `getArray` that returns a string of the values stored in the array, separated by commas. The array will start with `[` and be closed with `]`. For example: `[1, 3, 4]` or `[]`
5. Write a function `getArrayReverse`, that returns a string of comma-separated values of the array in the reverse order they appear in the array. For example: `[4, 3, 1]` or `[]`
6. Write a function `findMinMaxAvg` that takes as input an array of integers and returns an array of size 3 containing the minimum value, the maximum value and the average value.
7. Write a method called `areTheSame` that takes as input two arrays of integers and tells if all of the elements of the two arrays are the same. The order does not matter. For example, `[4, 3, 1]` and `[1, 3, 4]` are the same.
8. Write a function `areSorted` that checks if all the elements in the array are sorted, either incrementally or decrementally.
9. Write a method `getIndexMinValue` that takes as parameters an array of `n` integers and an index `i` and returns the index of the minimum value found in the array starting from `i` till `n-1`. In case, the min value exists twice, return the smallest index. For example: if the array contains `[44, 1, 125, 38, 189]`, `getIndexMinValue(array, 2)` returns 3 because 38 is the minimum starting from index 2 i.e. `[125, 38, 189]`

10. Write a method called `swapValues` that takes as an argument an array of integers and two indexes, and exchanges the elements pointed to by the indexes. For example, if the array is `[44,1,125,38,189]` `swapValues(tab,0,2)` transforms it into `[125,1,44,38,189]`

NB: when you pass an array to a function, it is passed by reference, not by value. Because arrays can be quite big, and java, doesn't want to allow big arrays to be copied by default, if you want to pass by value you have to create your own copies by yourself using the clone method:

```
int[] copyOfArray = array.clone()
```

Now we have reached the sorting part of this lab. Using the methods we have coded so far, you will code your first sorting algorithm in Java called *Sort by selection*. This sorting method is the most intuitive one to humans, but it is inefficient. There are better methods. For now let's just code this one.

Using the previous methods you wrote, you can now sort an array of integers.

11. Write a method called `sortValues` that takes as input an array of integers and sort it. You can use the function we coded before `readIntegers` to read an array of `n` integers and pass it to the method `sortValues`. You can also use `getArray` to print it so that you are sure everything is working correctly. The algorithm is simple. It is as follows:

- 1- First start at position 0 of the array and find the index of the minimum value between 0 and `n-1`, and return its index. You can use the function `getIndexMinValue(array, 0)`
- 2- Then swap the value at position 0 with the value at position index,
- 3- Increment one step (position should be now 1), search for the minimum value between 1 and `n-1`, and return its index. You can use the function `getIndexMinValue(array, 1)`
- 4- Then swap the value at position 1 with the value at position index
- 5- Repeat the process until the position is at `n-1` (last element).
- Finally you print the sorted array using `getArray` to check your result.

2.3 Exercise 3 : Creating a histogram

An exam is graded on 10 (integers). Only decimals are allowed, this means the possible values are (0,1,2,3,4,5,6,7,8,9,10). We would like to know how many students got a specific grade. We suppose we have `N` students. We have an array of `N` elements containing the grades of the students: `grades[i]` (Grade of student `i`)

1. Write a method `getStats()` that returns an array `stats`, such that `stats[g]` contains the number of students that got the grade `g`; You should not allow the user to input negative numbers.
2. Write a method `showDiag(int[] stats)`, that returns a string representing a diagram of the grades obtained by the students as follows:

```

0
1
2 ***
3 ****
4 *
5 ****
2
6 *****
7 *****
8 ***
9 ***
10 *
```

This reads like this: `stats[0] = 0`, `stats[5]=4`, `stats[6] = 8`, ...

3. Write a method `showDiagSorted(int[] stats)`, that returns a string of the diagram but this time in a sorted manner, from the most obtained grade to the least one. In case of equality, start with the highest grade first. Example:

```

6 *****
7 *****
5 ****
3 ****
9 ***
8 ***
2 ***
10 *
4 *
1
0
```

2.4 Exercise 4 : Matrices (2d arrays)

The syntax for manipulating matrices in java:

```

int [][] a;           // declares a matrix of integers
a = new int [l][c];   // allocate a matrix of l lines and c columns
a[i][j]               // access the element j of line i
a.length              // number of lines in the matrix
```

```

a[0].length          // number of columns in the matrix
// The lines of matrices are numbered from 0 til l-1
// and the columns from 0 til c-1.

```

1. Write a method called `readMatrix(int l, int c)` that takes two parameters as input, `l`: number of lines and `c`: number of columns and returns a matrix of `l*c` integers. The user should input all the integers of the matrix.
2. Write a method called `printMatrix(int[] [] mtx)` that takes as input a matrix and prints it inside a table. You can use the characters `_` and `|`. For example,

```

-----
| 1 | 2 | 3 |
-----
| 3 | 5 | 5 |
-----
| 2 | 4 | 5 |
-----

-----
| 10 | 2 | 3 |
-----
| 3 | 5 | 5 |
-----
| 2 | 4 | 51 |
-----

```

3 Conclusion

After finishing this assignment, you should now know how to use packages to organize your classes. You should also know how to use methods and arrays to solve different algorithmic problems.