3. homework assignment; JAVA, Academic year 2011/2012; FER

A usual, please see the last page. I mean it! You are back? OK. This homework consists of eight problems.

Problem 1.

Open PowerPoint presentation java_tecaj_03_prezentacija.pdf available in repository in Ferko. Solve the problem from slide 19 (actually, solution is given starting from slide 20; write it, run it, and give in JavaDoc comment for that class explanation of generated output and differences among the three outputs).

Problem 2.

Open PowerPoint presentation java_tecaj_03_prezentacija.pdf available in repository in Ferko. Solve the problem from slide 30. Put the implementation class in the same package as class in problem 1, and name it `Names`. For reading files use following snippet of code:

```
BufferedReader br = new BufferedReader(
    new InputStreamReader(
        new BufferedInputStream(new FileInputStream("filename.txt")),
        "UTF-8"
    )
);

while(true) {
    String line = br.readLine();
    if(line==null) break;
    line = line.trim();
    if(line.isEmpty()) continue;
    // here do something with this line since it is non-empty
    // e.g. write it out
    System.out.println("Got line: "+line);
}
br.close();
```

Assume that both files are generated using UTF-8 codepage.

Problem 3.

Open PowerPoint presentation java_tecaj_03_prezentacija.pdf available in repository in Ferko. Solve the problem from slide 35. Put the implementation class in the same package as class in problem 1, and name it `NoDupLines`. The main program should read lines from console until user enters empty line and store them in appropriate collection from Java Collection Framework. The number of lines is unknown and is limited only by available memory. Convert it then to an array and call your line-reversing-duplicate-removing function.

Problem 4.

Open PowerPoint presentation java_tecaj_03_prezentacija.pdf available in repository in Ferko. Solve the problem from slide 37. Put the implementation class in the same package as class in problem 1, and name it

AboveAverage.

Problem 5.

Open PowerPoint presentation java_tecaj_03_prezentacija.pdf available in repository in Ferko. Solve the problem from slide 43. Put the implementation class in the same package as class in problem 1, and name it `NamesCounter`.

Problem 6.

Open PowerPoint presentation java_tecaj_05_prezentacija.pdf available in repository in Ferko. Solve the problem from slide 41 (copy the solution and check that it works). Put the implementation class in the same package as class in problem 1.

Problem 7.

Write a generic class `LikeMedian` and put the implementation class in the same package as class in problem 1. The class must be usable in following scenario:

```
LikeMedian<Integer> likeMedian = new  LikeMedian<Integer>();
likeMedian.add(new Integer(10));
likeMedian.add(new Integer(5));
likeMedian.add(new Integer(3));
Integer result = likeMedian.get();
System.out.println(result);
```

The class should by able to work with any type which has defined natural ordering (remember the last lecture?). If user adds an odd number of elements, the `get()` method must return median element. If user provides an even number of elements, the `get()` method must return the smaller from the two elements which would usually be used to calculate median element. When completed correctly, this should also work:

```
LikeMedian<String> likeMedian = new  LikeMedian<String>();
likeMedian.add("Joe");
likeMedian.add("Jane");
likeMedian.add("Adam");
likeMedian.add("Zed");
String result = likeMedian.get();
System.out.println(result); // Writes: Jane
```

Problem 8.

Write a class `PrimesCollection` and put the implementation class in the same package as class in problem 1. The class must be usable in following scenario:

```
PrimesCollection primesCollection = new PrimesCollection(5);
for(Integer prime : PrimesCollection) {
    System.out.println("Got prime: "+prime);
}
```

The previous snippet should produce output:

```
Got prime: 2
Got prime: 3
Got prime: 5
Got prime: 7
Got prime: 11
```

You must not use any preexisting Java collection (or arrays) for element storage. Constructor of the class accepts a number of consecutive primes that must be in this collection. You must write your own support that will allow objects of your class to be used in `for`-loops.

Hint: you can use nested classes. Also, if you do not declare them as static, instances of these classes will be able to “see” properties of your top-level class. Here is an example.

```
public static class TopLevelClass {

    private int number;

    //some code

    private void method() {
       InnerClass1 instance1 = new InnerClass1();
       InnerClass2 instance2 = new InnerClass2();
    }

    private class InnerClass1 {

      public void innerMethod() {
        int x = number; // I see it!
        // do something else...
      }

    }

    private static class InnerClass2 {

      public void innerMethod() {
        int x = number; // I dont see it! Compile time error!
        // do something else...
      }

    }
}
```

Info: to help you to properly design you class, be aware that the following code must also work (and any similarly arbitrarily-deep nested loops) :

```
PrimesCollection primesCollection = new PrimesCollection(2);
for(Integer prime : PrimesCollection) {
  for(Integer prime2 : PrimesCollection) {
    System.out.println("Got prime pair: "+prime+", "+prime2);
  }
}
```

which should produce the following output:

```
Got prime pair: 2, 2
Got prime pair: 2, 3
Got prime pair: 3, 2
Got prime pair: 3, 3
```

Please note. You can consult with your peers and exchange ideas about this homework *before* you start actual coding. Once you open you IDE and start coding, consultations with others (except with me) will be regarded as cheating. You can not use any of preexisting code or libraries for this homework (whether it is yours old code or someones else). Document your code!

In order to solve this homework, create a blank Eclipse Java Project and write your code inside. Once you are done, export project as a ZIP archive and upload this archive on Ferko before the deadline. Do not forget to lock your upload or upload will not be accepted.