Introduction to Programming I (F2019)

Assignment III - week 6

1. Consider the following code.

```
package week6;
public class CauseProblem {
      public static String m1() {
            String result;
            // TODO
            return result;
      }
}
```

In the method `m1`(), you are asked to throw an object of the class `Exception` inside a `try` block. Give to the constructor of the exception the following String argument: “question 1”. Then, catch the exception inside a `catch` clause and set the variable `‘result’` with the value of the message contained in the caught exception. Add a `finally` clause that appends the string “; with finally” to the variable `‘result’`.

2. Create your own exception (ProblemException in the same package week6) by extending the class Exception. Write a constructor for this class that takes a String argument and stores it in the attribute message. Create a method m2 in CauseProblem (similar to m1) to demonstrate the use of your new exception.

    a. Again, use try-catch blocks and the throw keyword.

    b. Throw an exception of the new type that you have just created

    c. The string passed to the constructor should be “question 2”

    d. There should be a specific catch for your new exception and a catch for all the other possible exceptions, with the same behavior for simplicity sake.

3. Consider the method m3 that should be added to your class CauseProblem.

```
public static void m3() {
      // TODO
}
```

Add to the body of this method one single instruction throwing your new type of exception created in question 2. An error will appear after that. Write a comment within m3 explaining the reason for the error.

Then, fix the method using the `throws` keyword.

4. Create a method m4 (similar to m3) in which you must define an object reference of the class Object and initialize it to null. Try to call the method hashCode() through this reference. What’s the error? Write a comment within m4 explaining the error and then wrap the code in a try-catch clause to catch the exception and solve the problem.

5. Add the following two public static void methods: f( ) and g( ). In g(), throw an exception of a new type that you have defined in question 2. In f( ), call g( ), catch its exception and, in the catch clause, throw a different type of exception (RuntimeException).

6. Given the following code:

```
public static void m6() {
      // TODO
}
```

Add some lines of code to cause an ArithmeticException, an ArrayIndexOutOfBoundsException, a NullPointerException, a NumberFormatException and a FileNotFoundException exception **in this order**.

This code should be contained in a single `try` block. Next, add one specific `catch` block for each type of exception plus a `catch` for all the other types of exceptions. The behavior should be the same for all the `catch` blocks, that is to print the message in the `Exception` object.

Additional instructions:

- Your code should contain no errors and compile (-100% of the grade if it doesn't compile)
- Format your code using proper indenting (-50% of the grade if your code doesn't follow this good practice)
- Again, plagiarism will be automatically checked and not tolerated (-100% with further implications for the involved students)
- Submit **one single Java file** containing the public class CauseProblem and a class ProblemException (-50% if this requirement is not attended)
- Add a comment somewhere in your code with your name (-50% if this requirement is not attended)