Programmierpraktikum

Exercise Sheet #3

WS, 2012/2013

Intructions:

The exercises must be sent to your tutor in a *tar* file via email until midday, the day before the practice lesson (unless you arrange differently with your tutor).

The file you are sending must be named like in this:

```
pp-ex<number>-<firstname1>_<lastname1>-<firstname2>_<lastname2>-<firstname3>_<lastname3>.tar.gz
for example: pp-ex3-john_doe-max_mustermann.tar.gz
```

To generate the file, you have to put all your code in a directory, for the previous example:

```
mkdir pp-ex3-john_doe-max_mustermann
```

Then you work inside of that directory, and when you are ready, you save it:

```
tar -czf pp-ex3-john_doe-max_mustermann.tar.gz pp-ex3-john_doe-max_mustermann/
```

Prepare the *tar* file with all the code of this exercise in an ordered way (make a file explaining which file does what if you use many files and directories). The email subject should have the format:

```
[progprak] ex3, firstname1 lastname1, firstname2 lastname2, firstname3 lastname3
```

Please, when sending your solutions, send only *one* email with your *definitive* solutions.

The code must compile and run to receive any points. Comment all your code properly (i.e. put a comment explaining what each non-obvious part of the code does.) Write also the answers to the questions in the exercises as comments.

Printing and formating

As in the last sheet, your program can print to the standard output via the `println` command:

```
System.out.print("Hello, World!");  //this prints the phrase "Hello, World!"
System.out.println("Hello, World!");  //this prints the phrase "Hello, World!"
                                      //    and then starts a new line
```

You can also use `println` to print values of variables.

- Define an `integer`, a `float` and a `boolean` and assign values to them. Then make a program that prints sentences using each of these variables like e.g. "The value of variable a is 3". You can do this by using the "+" operator. To do this, look for examples in the class and check the documentation.

Specially in the case of floats and doubles, you might want to define a format for the numbers. For this you can use the `format` method. For example:

```
System.out.format("The value of pi is %.2f", Math.PI);
```

will print the value of π converted to float (the `f`) with two decimals (the `.2`).

Using the `format` method, write a program that prints the value of π:

- with 15 decimals.
- with 2 decimals.
- the closest integer value.
- in scientific notation.
- filling exactly 20 characters, of which 10 are decimals.

hint: Check the documentation for the class `format`. For instance
http://docs.oracle.com/javase/tutorial/essential/io/formatting.html

Primitive Data Types

The information in your program must be stored in the memory. The computer only has bits (0 or 1), but this bits can be organized in groups to represent different data. For the most common types of data, there are a set of optimized and standardized data types to represent them. In statically typed programming languages like Java, it is the task of the programmer to inform in advance (via a definition) about what data type is each information.

Write a program that does the following:

Integers:

- Define an `int` variable with name myint.
- Find out the maximum value that this variable can take.
- Try to set the value of this variable over that limit. Compile your code and see which error you get.
- Which other type should the variable have in order to hold the last point's large number?

Floats and Doubles:

- Define a `float` variable to hold the number π with four decimals. Print its value.
- Add then a float variable with value 10^{-10} to the previous variable. Print the result again.
- Did the printed value changed? Is there a better data type for this operation? Repeat the calculation with that data type.

Booleans, Characters:

- Define a variable of type `boolean` and another one of type `char`. Assign them the values `True`, `1`, `c`, `hello`. Notice assignment errors.

Arrays

Arrays are essentially like a mathematical vector. They are structures that store a number of primitive elements, for instance numbers.

- Define an array of 10 integers. Assign the values 1,2,3... to the elements of the array. Print then the values of the array.
- Now try to assign the values 'a', 'b', 'c'... to the same array? What happens? Define an array that can actually store those new values.
- Try to change now to 0 the value of the last element of the integers array. Then try to assign an element of index bigger than the size of the array. Notice what happens. Think of a solution for that case.