Object Orientation – Spring 2018 **Assignment 1**

Learning objectives

After having completed this assignment, you should be able to

- write, compile and execute a Java program in the NetBeans IDE;
- define classes with attributes and methods;
- distinguish static and dynamic methods;
- define objects as instances of such a class;
- use Java arrays;
- use rudimentary Java input/output;
- define and use a `toString()` method for user-defined classes;
- concentrate all input/output in a single class.

Programming environment

We advise you to use the **NetBeans** versions 8.2 for this practical course. This programming environment innately works with Java 8. In this course we will make use of specific features of this very version of Java on a few occasions. The programming environment NetBeans is installed on all student PCs at the Faculty of Science and can be downloaded for free from `netbeans.org` for all usual platforms. The standard edition Java SE is sufficient for this course.

For some of the exercises, we will provide pre-written code snippets for you to use. We always use said versions of NetBeans to test our pre-written code, but there is no specific reason for it not to work in different programming environments with Java 8. Some of the computers in the Spinoza building have the Java IDE Eclipse installed. Be aware that they use the outdated Java 7 which might lead to errors when running some of our provided code.

Since **NetBeans** will be available during the exam we encourage you to use it, even if you are currently used to a different IDE.

Javadoc

Every program you write for this object orientation practical course should contain a minimal amount of javadoc documentation:

- the name and student number of both authors in every file.
- a short description the purpose of the class if the name does not describes its purpose.
- a brief description of the purpose of a method and its arguments if the names are not self-explanatory. State preconditions explicitly if they are essential for a proper behaviour.

Have a look at the programming style instructions on Blackboard.

1 Assignment

To achieve the goals of this assignment you have to design and implement a Java program that creates a group of students and offers the user the possibility to change the name of these students.

Each student has a first_name and a second_name, both strings, and an integer for student number. These values are specified upon the construction of the student object. The student number is fixed, but the name can be changed by an appropriate setter method. The student with name Alice Liddells and number 42 should be shown as `Alice Liddells, s42`.

The behavior of the program to create and manipulate a group of students is summarized as:

1. Your program should ask the user how big the group of students is supposed to be and create a new group of the given size.
2. For each group member, your program should request a name and a student number and add this student to the group until it is full.
3. Afterwards the program should display the text `The group now contains:` and print all the students from the group.
4. In a loop, the program will then ask the user to input the student number (only the numerical part without the `s`) alongside with a new name: `Student number and new first/second name?` Every time a name is changed, the entire group should be shown again on screen with the message `The group now contains:`.
5. The program terminates as soon as the user enters a negative student index.

Follow the general guidelines to develop an object oriented program as outlined in the lecture and tutorial.

Reflection

Make it a good habit to check the learning objectives before you start and finish an assignment.

- Does your program reflect the learning objectives?
- Are you able to apply these skills in a similar new project?

Hand in

Hand in your code **before Sunday 11 februari, 23:59 on Blackboard**. Do not forget to put javadoc with your name, the name of your programming partner and your student numbers into **every** file. The automated administration system uses this information to file your results and send the feedback back to you - so take care that it's always in the file.

Please **only** hand in the `.java` files that you created or modified yourselves. If you create a project with NetBeans, all `.java` files from your project will be put into the `source` directory of that package. Be careful: this folder has the same name as the folder of your entire project – which you are *not* supposed to hand in.

Choose a discernible name for your main class.