Lab 1

October 28th, 2022

Throughout the course we will use the Eclipse IDE for Java programming. Please refer to the tutorial in GitLab for a description of how to setup the IDE including a Hello World tutorial. The following tasks are for refreshing your knowledge of basic (procedural) programming. Depending on how much you struggle with these tasks, you may want to consult additional online resources. Otherwise, you will have difficulty completing the lab.

Task 1.1: BMI calculation & SimpleIO

Write a program that calculates the body mass index (BMI) of a human. The BMI is a measure for a human's body shape (i.e. underweight, normal weight, overweight) based on his or her weight and height. It is calculated with the formula:

$$\text{BMI} = \frac{w}{h^2}\,, \quad w := \text{weight in kg}, \quad h := \text{height in m} \tag{1}$$

When the program is started, it queries weight and height from the user. Then it calculates the BMI and displays it to the user. Complete the following steps:

1. Create a new Eclipse project and add a class named `BmiCalculation` with a method `main()`, as in the *HelloWorld* example from the Eclipse tutorial (there is no need to create a `module-info.java`)
2. Declare the needed variables given in Eq. (1) and choose appropriate data types.
3. Write statements that query weight and height from the user and assign them to the corresponding variables (see below how to import the class `SimpleIO`).
4. Calculate the BMI and display the value to the user.
5. Test your program for several inputs.

SimpleIO

For this lab we have provided the class (= module) `SimpleIO`, which simplifies handling of console inputs and outputs in Java. First you need to import the package:

- Create a new package (*File→New→Package*) with the name `de.tuhh.diss.io`. You will see it in the package explorer on the left.
- Download the file *SimpleIO.java* from the public repository and drag and drop it into the eclipse project explorer into the package just created.
- To import `SimpleIO` to your Java class, add the following as first line of your program:

```
import de.tuhh.diss.io.SimpleIO;
```

Then you can use its methods `SimpleIO.print()` and `SimpleIO.println()` to display data to the user. Methods such as `SimpleIO.readInteger()` and `SimpleIO.readDouble()` can be used to query data from the user, e.g.

```
myNewInteger = SimpleIO.readInteger();
```

queries an integer value from the user and assigns it to the variable `myNewInteger`. Note that you have to declare `myNewInteger` first.

Task 1.2 : Check inputs using loops

Since your BMI calculation might produce illogical output data if unexpected values of weight or height are entered, the program should be modified. Use a loop to ensure the user enters neither a negative weight and height nor a height exceeding 2.72 m (height of Robert Wadlow, the tallest men ever recorded). The user should be prompted again and again until a legal value is entered. What kind of loop is useful? Proceed with the calculation and print the correct result to the console.

Why are if-statements *alone* no good idea for this task?

Task 1.3 : Classification of results using if-statements

Based on the BMI, you can rate if the entered person has underweight, normal weight or overweight:

Condition	Result
BMI < 18.5	underweight
BMI ≥ 25.0	overweight
$18.5 \leq$ BMI < 25.0	normal weight

Add two double constants to your program and protect them from modification within the program:

- `BMI_UPPER = 25.0`
- `BMI_LOWER = 18.5`

Use if-statements, the given table and the constants to perform classification. Display the results to the user. How many if-statements do you need?

Task 1.4 : Storing values in an one-dimensional array

For later retrieval, we want to store the BMI for each person separately. For that purpose, replace the double variable `bmi` by a suitable array to store the value for each person.

Create a dialog so that the user is able to input a new user or to print a specific entry of the array. Keep in mind that you cannot store more users than you defined your array for, i.e. prevent out-of-boundary access.

Task 1.5 : Calculation within a for-loop

For statistical purposes, it is interesting to know the average BMI of different persons. Use a `for-loop` to access all elements in the BMI array and cumulate them in a temporal variable. Only treat non-zero elements. Divide the sum by the number of BMI entries in the array.

Print the average to the console along with the entries itself.

Note: a similar task has been part of the exam in Winterterm 18/19.

Task 1.6 : Java Methods

In training theory, the maximal oxygen consumption (VO_2 max) is an important factor to assess your aerobic fitness. Research has shown that VO_2 max, maximum heart rate (HR_{max}) and resting heart rate (HR_{rest}) are related by the following equation:

$$VO_2 \text{ max} \approx 15 \times \frac{HR_{max}}{HR_{rest}} \quad [\mathrm{mL}/(\mathrm{kg\,min})] \tag{2}$$

HR_{max} and HR_{rest} may be estimated by the age of the person, via equation

$$HR_{max} \approx 206.9 - (0.67 \times \text{age}) \quad [1/\mathrm{min}]\,, \tag{3}$$

and by the given (rough) classification table:

age	18-35	36-55	56+
HR_{rest}	71	73	76

Write a class `MaxOxygen` that calculates VO_2 max based on the age of a person. Your program contains three methods for:

- VO_2 max calculation,
- HR_{max} calculation,
- HR_{rest} decision.

To ensure all methods are accessible by each other, declare them as `public static`[1] and use a suitable return type. Write a method `main()`, which queries the age of the user and print its VO_2 max to the console.

[1]The purpose of this will be explained in the following lectures