Step 4 – User-input, ArrayList and Debugger

In the previous step, we learned how to create objects. We created book objects. But the data for the books we made came from the code. What if we don't know that data? We might want to have a user provide the relevant information.

If anyone needs this in Norwegian, a video exists about this topic from last year, curtesy of Per.

The goals for this step:

- I can use Scanner to receive user input.
- I can use an ArrayList to
    - manipulate objects
    - go through objects
- I can create a user menu using loops
- I have become a little familiar with the debugger in IntelliJ

Chapter 6 in the book deals with both array and ArrayList (and includes some use of Scanner).

Here is the feedback questionnaire for step 4 where you can tell how it went.

Task 1

Create a new project. Add a class that has a main method.

Task 2

Create a class «Program». The program class must have a public method «runProgram» which does not take any parameters, but which only prints (System.out.println - hereinafter referred to as SOUT) «Program starting».

Task 3

In your main method; create an object of the Program class and call the runProgram method.

Run your main method. Hopefully your program prints "Program starting" before it ends. Then you are done with the preparations.

Task 4

In the Program class, create a method «task4». The method should create a local variable of type Scanner, and it should refer to a new Scanner object that can read from System.in. This is explained in the intro video, but you can also read more here.

Use your Scanner object to load three strings. But where are we going to place them? Since they are of the same type (String), we can place them in an ArrayList! Hurray! Therefore, create an ArrayList that can store strings, and place the strings there. You can read more about ArrayList here.

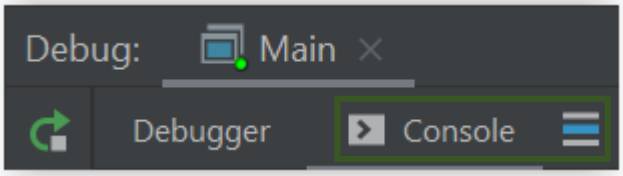
But we have to get some training in going through the elements of an ArrayList. There are many ways to do this (check the previous link), but choose the way you like best. Go through all the items in your array list and print the value (SOUT).

But how do we check that this works? This is something you need to think about. And to make sure you think so it cracks (and maybe finds out), I'll help you with white text on a white background below. If you give up, copy the text and paste in a place (e.g. Notepad) to get the help:

Task 5

We need to have some training in using debugger. It is completely raw. In your main method, enter (at the very beginning of the method) a SOUT: "In the start of main ()". Put a break-point on this code line (check lecture, or read more here). Start your program in debug mode. Follow your program step by step. "Step into" the methods you have created (runProgram and task4). Along the way, see what variables are created and their value. Follow the program until it ends.

You will find that when you step into a step that requires user input, you must go to the Console tab to enter your value:



Do not forget that the debugger is gold if you are struggling to find a bug in your program. Put a break point in the area you assume your error is, run the program in debug mode, and you can get lots of valuable help in finding it. (This only applies to programs that can run, i.e. do not have compilation errors.)

Task 6

Create a method task6 (in the Program class). Use a loop (you choose the type of loop) where you read in numbers from the user. Keep the numbers that the user enters. The loop should run until the user enters a negative number. When the loop is complete, print (SOUT) the sum of the numbers entered. The negative number should not be included in the sum. If you think this was more than difficult, then I have a little more wonderful if text on a white background to you below 😊. But try to solve the problem without help first.

Task 7

Today's most difficult (before the extra task). Here you just have to steel yourself.

Create a method task7. The method should present a menu to the user:

1. Add single word

2. Display all words
3. End

This week we will not take into account that the user does something wrong (it will come later), so you can assume that the user will always be able to enter the number 1, 2 or 3.

When the user selects 1, you will receive a word from the user and put it in an array list.

When the user selects 2, print (SOUT) all the values in the array list.

When the user selects 3, you should print (SOUT) a farewell greeting.

Test that the method works satisfactorily by entering some texts, and then select option 2 before exiting (3).

Extra task

Go back to the extra tasks for step 3. Now that we have been introduced to ArrayList, we can replace the array we used in step 3 with an ArrayList. Then it is also much easier for us to be able to remove books from the book register.

Expand your solution (which you created in step 3) with the option for the user to add books. Create a menu where the user can:

- Add books
- Get an overview of all books
- Get an overview of all books by a specific author
- Get an overview of all books in a genre
- Get an overview of all books with a maximum reading time. If you (like me) think that reading time per page should be located on Book and not Chapter, then move the field to Book.
- Remove books (maybe add an extra field ISBN in Book to be able to remove based on id?)

To avoid starting with an empty register every time you run the program, you can enter some books via code at the beginning of the program.