Programming 1

Tutorial 10

Activity 1

The text-based Java programs that we write in this course are called console programs. One way to enable a user to interact with a console program is through a menu. Here is an example:

```
[1] Add item
[2] Item list
[3] Edit an item
[4] Remove an item
[5] Quit
Choose an option: 1

Adding a new item...

[1] Add item
[2] Item list
[3] Edit an item
[4] Remove an item
[5] Quit
Choose an option: 2

Listing all items...

[1] Add item
[2] Item list
[3] Edit an item
[4] Remove an item
[5] Quit
Choose an option: 5

Goodbye!
```

Let's analyze this program. First of all, the menu text is printed out on the screen. The menu text is `blue` in the example. Then, the user chooses an option by entering an integer. The user's input is `orange` in the example. Next, according to the user's choice, the program takes an action (resulting in `green` texts in the example).

This is a way to allow a user to choose one of many actions to carry out in the program. After one action has been performed, the program prints out the menu again and let the user choose the next action unless the last choice is `5` (Quit). The way this program works is like a `while` loop which stops when the user's input is `5`. So how to write a `while` loop like that? Suppose the user's input is stored in variable `n`, then:

```
while (n != 5) {
    // loop content
}
```

Now, we have to determine the loop's content. To do this, answer the question: *What things are repeated?* They are: printing menu text, getting user's option, performing actions based on user's input. Isn't that everything? Nope, there are also declaring variable `n` and initializing `Scanner`. But yeah, almost everything gets repeated. So, our program technically contains some variable declaration and a big while loop which stops when `n` is `5`.

How to choose the action based on the user's choice (variable `n`)? Well, we have learned the `if…else` statement and we have also learned the `switch` statement. Any of them should do the trick.

In this activity, you'll practice writing such menu-driven program.

(*) In the menu-driven program, each user's option should execute one method. This kind of program structure is called a *modular* program.

(*) Make sure to pay attention to the newline (`'\n'`) character in the input buffer when working with `Scanner`.

Activity 2

Create an `ArrayList` of Strings. Add 3 Strings to it. Use `ObjectOutputStream` to write the `ArrayList` object to a file called `data.bin`.

Activity 3

Use `ObjectInputStream` to load an `ArrayList` of Strings from the file `data.bin` produced by the last activity and display the Strings on the screen.