

Extra Lab Sheet 3: Methods and Strings

Make sure that you save all of the today's work under `ecs717/lab03` folder structure.

The source-code of all exercises will be available to you after the end of the today's lab.

Getting started

Remember the "Hello World!" example from last week? Here are three alternative versions that use methods:

```
public class Hello4 {  
    public static void main (String args[]) {  
        printMessage();  
    }  
  
    public static void printMessage() {  
        System.out.println("Hello World!");  
    }  
}
```

In this example, we call the method `printMessage` that prints the message "Hello World!" in the terminal screen. It doesn't return anything (`void`) and doesn't have any parameters (nothing inside the `()`).

```
public class Hello5 {  
    public static void main (String args[]) {  
        printMessage("Hello World!");  
    }  
  
    public static void printMessage(String message) {  
        System.out.println(message);  
    }  
}
```

In this example, we call the same method `printMessage`, but this time we pass the string "Hello World!" as a `String` parameter. The method is still `void` as it doesn't return anything again.

```

public class Hello6 {
    public static void main (String args[]) {
        System.out.println(message());
    }

    public static String message() {
        return "Hello World!";
    }
}

```

In the final example, we declare the method `message` with a return type `String`. It returns the message "Hello World!" that we print in the main method.

Exercise 1

Write the method `long pow(int base, int exponent)` that calculates the value of the first argument raised to the power of the second argument (e.g. $\text{pow}(2, 3) = 2^3 = 8$). Calculate the result of the operation $4^4 + 2^3 + 5^{10}$ in the main method. *What's the reason of using long instead of int as a return value?*

Now make the appropriate changes in order to support decimal (type: `double`) numbers in the base parameter (but not in the `exponent`). Calculate the result of the operation $5.4^2 + 2^3 + 5.5^3$.

Exercise 2

Another way to provide user input to a java program is with the use of the `Scanner` object. The following program is very similar to Exercise 2 of Lab Sheet 1, but prompts for user input during the program execution, using the `Scanner` object.

```

import java.util.Scanner;

public class Hello3c {

    public static void main (String args[]) {

        // Create a Scanner object
        Scanner reader = new Scanner(System.in);

        // Read First name
        System.out.print("Enter your first name: ");
        String first = reader.next();

        // Read Last name
        System.out.print("Enter your last name: ");
        String last = reader.next();

        // Read e-mail
        System.out.print("Enter your e-mail: ");
        String email = reader.next();

        System.out.println("Hi " + first + "!");
        System.out.println("Your e-mail address is: " + email + ".");

    }
}

```

Add the following restrictions to the previous example:

1. The length of the input Strings should always be greater than three characters.
2. The e-mail address should always contain the characters '@' and '.'.

When the user input does not comply with the above restrictions, the program should warn the user with a message and then terminate. It is a good idea to check the strings against the two restrictions using two separate methods (*e.g.* `Boolean checkLength(String str)` *and* `Boolean checkEmail(String str)`).

Exercise 3

Write a program that asks for the username and password of a user and prints "Access granted!" when the combination of username/password is correct (*e.g.* `user42/pass42`) or "Wrong username or password." when the combination is incorrect. Make sure you use a separate method for checking the user credentials.

Hint: Search for how to compare two Strings in Java!

Exercise 4

In this exercise you need to develop a basic command line calculator. The program should receive simple operations of integer numbers with two digits and calculate the result. You need to support the following operations: +, -, *, and ^ (*power*). You may use the method `power` that you developed in Exercise 1.

```
Enter your operation: 12+12
12 + 12 = 24
Enter your operation: 12^02
12 ^ 2 = 144
```