Distributed Systems
Exercise Sheet 1
Klingemann, SS 2022

Voluntary Java-Exercise (without assessment)

Java-Exercise 1
Write a program that contains a function which calculates the factorial of a non-negative integral number. Call this function from your main-function.

Java-Exercise 2
Implement a class `Publication`, which contains the private attributes title, language and price. The class should contain exactly the following methods:

- A constructor with three parameters that is initializing the three attributes with the corresponding parameter values.
- A method `print` that is displaying the values of the three attributes on the screen.

Implement a class `Book`, which is derived from `Publication`. `Book` contains the additional attributes Author and ISBN.
The class should contain exactly the following methods:

- A constructor with five parameters that is initializing all five attributes (including the inherited ones) with the corresponding parameter values.
- A method `print` that is displaying the values of the five attributes on the screen.

Write an appropriate main-function to test your classes. Use for this purpose an array of type `Publication`. Assign objects of both classes to the components of the array. Iterate over the array and call on each object the method `print`.

Java-Exercise 3
Write a program that is taking all strings from the command line and is displaying them on the screen. However, in the output duplicates should be eliminated.
Hint: Use the class `java.util.HashSet` and insert all strings from the command line into this set. Afterwards, take all strings from this set by using the class `java.util.Iterator` and display them on the screen. (Note, that a set does not contain duplicates by definition.)

Java-Exercise 4
Write a class that is derived from the class `Thread`. Each object of your class should have a number as an attribute. After the start of the thread, this number is displayed five times on the screen. Between each pair of statements to display the number, there should be a pause of three seconds.
Write a second class, which is waiting continuously for a user input. For each line of user input, a new thread should be created and started. The threads should get numbers in increasing order.
When necessary consult the Java API-documentation for the class `Thread`.

Hint: You can use the following code-fragment to read a line from the keyboard:

```
Scanner sc = new Scanner(System.in);
String line = sc.nextLine();
```