Assignment 4: Reuse of classes, co-operation of objects, Java API classes (ArrayList, StringBuilder)

Exercise 4.1

In Assignment 1, we had a class that represent an order line. Use that class in this exercise (you can add a suitable constructor to your class, if needed).

Write a class that represents the features of an order: An order has variable for the order ID, the name of the customer, the order date (object of your own date class of Assignment 2), and an ArrayList collection for order lines. Implement following methods:

- a constructor that gets the order ID, the name of the customer, and the order date as parameters
- a method that adds a new order line to the order
- a method that calculates the total price of the order
- toString-method that returns the data of the order as a string. Use here a StringBuilder object to build the return value.

Write a short main method to test your class. You do **not** need to ask the values from the user.

```
OrderId:    1056
Customer:   Charlie Brown
Order date: 18.11.2018
Product              Price     Quantity    Sum
Pencil                2,50        100     250,00
Notebook              5,00         20     100,00
Pencil case          14,95         10     149,50
Total price: 499,50 €
```

Exercise 4.2

As an attachment of this assignment, there is a readymade class that represents a (statistical) distribution so that you can use an object of this class to calculate the frequencies of integer values min, min+1, min+2, ..., max and the average. There are methods:

- a constructor that takes integer values min and max as parameters and creates an array to store the frequencies
- a method that inserts a new value to the distribution
- a method that returns the frequency of a specified value
- a method that returns the average of the distribution
- a method that returns how many values are inserted to the distribution

a) Write a new class the object of which knows how to print a distribution. It has a reference to a distribution object as its instance variable and the distribution to be printed is passed as an argument to the constructor. It has a method that prints the distribution in a neat form (it prints out frequencies of the values and the average).

b) Write a program that calculates the grade distribution of a course. The possible grades are values 0, 1, ..., 5. The program asks the grades form the user. Use a distribution object to do the calculations, and a "distribution printer" object to do the printing.

c) Write a program that rolls a dice as many times as the user wants. The program calculates the distribution of the scores. Use the dice class from Homework 1, use a distribution object for calculations and use a "distribution printer" object to print the distribution.

Use the Debugger of Eclipse to explore how your program is running.