

Exercises for the Class
Elements of Computer Science: Programming
Live Assignment 5

Submission of solutions for group 1 until 2:00 p.m. and for group 2 until 3:00 p.m.
at `moodle.uni-trier.de`

- Submission that can't be compiled are rated with **0** points!
- Please comment your solutions, otherwise you can lose points!
- If you try to cheat, you will lose your points and the classroom exercise will be over!

Exercise 1 (Evaluation: predefined main method)

(10 Points)

Given is the class `Product`, which stores the name (`String name`), a product category (`String category`) and the price (**`double price`**) of a product. All instance variables of the class are **`public`**, which is why getter methods are not necessary.

In the class `Evaluation`, implement the method

```
double categoryValue(String category)
```

which calculates and returns the value of all products with the category `category`.

Exercise 2 (Evaluation: predefined main method)

(10 Points)

Implement the class `Restaurant`. An object of this class stores the name (`String name`) and a rating (**`double rating`**) of a restaurant. All instance variables of the class should only be accessible to the class itself. Implement next a constructor of the following form:

```
public Restaurant(String name, double rating)
```

Make sure that, as usual, the rating is in the interval between 0 and 5 inclusive. If a value is less than 0, set the rating to 0 and if the rating is greater than 5, set the rating to 5. Furthermore, implement all required getter methods.

Next, implement the following method in the `RestaurantFinder` class:

```
public Restaurant findBest(Restaurant[] restaurants)
```

The aim of the method is to find the best restaurant from the restaurant array. The best restaurant is the restaurant with the highest rating.

Exercise 3 (Evaluation: predefined main method)

(10 Points)

Your task is to implement a class `ArrayDictionary` and `DictionaryElement`.

A `DictionaryElement` stores an `Object` `key` and an `Object` `value`. These variables should only be accessible by the class itself. Therefore, implement all necessary getter methods of the class.

In addition, you have to implement a constructor

```
public DictionaryElement(Object key, Object value)
```

that initialises both parameters accordingly.

The class `ArrayDictionary` stores an array `DictionaryElement[] dictionary`. The size of the array is specified via the constructor and the dictionary is initialised accordingly in the constructor. Additionally, implement the methods of the interface as follows:

- **boolean** `add(Object key, Object value):`
The method searches in the dictionary for the next free position and then stores a `DictionaryElement` there with the values `key` and `value`. If an entry with the `key` already exists, overwrite the saved value `value` with the new value.
If a `DictionaryElement` is stored successfully, **true** should be returned. If the array is already full, **false** should be returned.
- `Object get(Object key):`
The method searches in the dictionary array for a `DictionaryElement` with the same `key`. If no `DictionaryElement` with the searched `key` exists, return the **null** reference.

Exercise 4 (Evaluation: predefined main method)

(10 Points)

Given is the class `Product` which stores an ID (**int** `id`), the name (`String` `name`) and the price (**double** `price`) of a product.

Implement next in the class `ProductParser` a method

```
Product parse(String productString).
```

The method is passed a string containing information about the ID, the name and the price. Examples of such strings can be found in the following example:

```
12345;Smartphone;399.99  
12;Smartphone;399.99
```

```
; Smartphone; 399.99  
12345; ; 399.99  
; ; 399.99
```

The aim of the method is to process the given string and create and return an object of the class `Product`.

The individual pieces of information are always separated by a semicolon (;) and always in the same order. This means that the ID is given first, followed by the product name and finally the price. However, as you can see from the examples, it can happen that an ID is incorrect (i.e. shorter than 5 characters and simply does not exist). In this case, the ID should be set to -1. It can also happen that the product name is missing. In this case, you should set the product name to "Unknown".