Prof. Ralf Schenkel
Tobias Zeimetz
Trier University

Wintersemester 21/22

Exercises for the Class

Elements of Computer Science: Programming

Live Assignment 06

Submission of solutions until 6:00 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!

Exercise 1 (Evaluation: predefined `main` method) (15 Points)

Implement an abstract class `Person` that can store two pieces of information: The first and last name of a person (each as an `String`). Additionally, implement a suitable constructor.

The class consists of a method **`public`** `String toString()` and shall return the first and last name of a person.

The class should also contain two abstract methods:

- **`void`** `learn(String newWord)`
- **`int`** `getNumberOfWords()`

Derive from the `Person` class the two subclasses `PersonA` and `PersonB`. Implement appropriate constructors as used in the `Evaluation` class.

Since `PersonA` and `PersonB` in this example learn words of a new language in different ways, the method `learn(...)` must be implemented for both classes. It is important that in the class `PersonA` the learned words are stored in an array (initialized with 100 fields). In the class `PersonB` all learned words are stored in a `List<String>`.

Since `PersonA` and `PersonB` use different data structures to store the words, the method `getNumberOfWords()` must also be implemented separately for each class.

Implement appropriate constructors for the classes `PersonA` and `PersonB`

Exercise 2 (Evaluation: predefined test class `Test`) (15 Points)

Given is the class `PhoneBookEntry`, which stores a name (`String name`) and a phone number (`String number`). Your taks is to create a `PhoneBook` class that contains only **`static`** class members:

- A `PhoneBook` stores a `List` contianing objects of type `PhoneBookEntry`.
- Implement a method **`void`** `change(name,number)`, which can change an entry in the phone book (if already contained by the book). That means in case that the name is not known to the phonebook (in this example names are unique) create a new entry containing the name and the phone number. But if the name is known to the phone book you have to replace the old number with the new one.
- Implement a method `String searchNumber(name)` that returns the phone number for a name. If the search method searches for a name that is still unknown, the string `unknown` should be returned.

In order to keep the solution simple, you can assume that no **`null`** references occur as parameters. **You can change the test class but for the evaluation we will use the original version.**

Exercise 3 (Evaluation: predefined `main` method) (10 Points)

First a string is read in the specified program. Then an attempt is made to convert it into an **`double`** number and then output. During the conversion, however, depending on the string, conversion errors may occur.

Modify the program by using exceptions in such a way that the input is repeated so often (without producing further output), until for the first time a string could be converted into a number without errors.

Note: Use exception handling, e.g. **`try`**–**`catch`** statements.

Exercise 4 (Evaluation: predefined `main` method) (10 Points)

In the given files an interface `myDataInterface` is defined, in which there are two methods: One method `init(int array)` is to copy(!) the data of an **`int`** array to instances of the interface, a second method `at(int n)` is to read and return the value at the position `n`.

Write an appropriate class `myData` (with matching constructor) that implements this interface. You must decide yourself how to store the data transferred during initialization.