

Exercise: Copy Constructor and Assignment Operators

Learning Objectives

- Deep copy versus shallow copy,
- Copy-Constructor
- Assignment Operator

Additional Objectives (if not already done)

- avoid global variables
- Using classes
- **Getter** and **Setter**

Exercise at a glance

- Replace the global variables, if not already implemented.
- Use classes instead of the structures, if not already implemented before
- Implement also a minimal standard interface for your classes Levenshtein class, which allocates memory on the heap
- Visualize that you have no memory leaks.

Detailed Exercise Descriptions

In this exercise we do not implement new functionality to reach the final aim of extraction of call signs from ATC transcriptions, but just learn how to implement Copy-Constructor and Assignment-Operator, although this functionality might not be very helpful for our final goal.

Re-implement your code from the previous exercise, so that copying of an instance of the Levenshtein distance class is possible. Do still not use the STL-template class vector in this class. For the rest of the code you may use the vector template class.

Exercise 3.1

If you are still using global variables, replace them by e.g. using a class. Your previous code might contain

Exercise 3.2

Replace all structs via classes, i.e., avoid attributes in the public area. All attributes should be in the private area and all be usable via Getter and Setter methods and other public methods.

Exercise 3.3

Implement a copy constructor and an assignment operator for the class `Levenshtein` and provide suitable tests.¹

Write tests, which show that your copy constructor and assignment operator work correctly. I will change the code of the copy constructor and assignment operator and then at least one of the tests needs to crash or result in false.²

Do not forget, that the minimal standard interface requires more than a copy constructor and an assignment operator. And also test these methods etc.

Exercise 3.4

Also implement a stress test, which shows that you have no memory leaks (Tests can only show the presence of error, but never/seldom their absence).

¹ Do not use a different class name., Ff you have chosen a different one in the previous exercises, please change now.

² Implement something similar as the `DoNothing` function in the lecture and more tests.