Projects

- There are 30 projects. The list with the projects can be found at the course webpage.
- Projects are composed of an ADT and a representation for the given ADT. Your task is to:
  - Find a problem that can be solved with the given ADT.
  - Implement and test the given ADT (all operations from the ADT, even if they will not be used later).
  - Implement the solution of the problem using the implemented ADT.
- Realization of a project consists of the implementation of an application and the writing of a documentation.
  - The application has to be implemented in C++, and it does not have to be generic (you can implement the ADT directly with the elements that will be stored in it for solving the problem, no templates needed).
    - The application does not have to use layered architecture, it is enough to have a class for your container, one class for the iterator, one file (not necessarily a class) for tests and a file (not necessarily a class) for the application. This last file can contain both read/write operations and actual computations.
    - If you have an array based representation (dynamic array, linked list on array, heap, hash table) you do not have to implement resize. Just pick an initial capacity large enough so that your application will work (do not pick a capacity of 10 if you intend to add more than 10 elements).
    - **In the implementation you are not allowed to use vector, list or any other container from STL or other libraries.** You have to implement the container from scratch.
    - Every operation from the interface of the ADT has to be tested (tests with assert).
    - If the application is not working, the project grade will be 4.
  - The documentation has to contain the following elements:
    - ADT – specification and interface (independent of the representation). If the container can be iterated, then the interface of the iterator has to be written as well.
    - Representation of the container and implementation of the operations in PSEUDOCODE (iterator as well).
    - Tests for the container and iterator (can be copied from editor).
    - Complexity of the operations from the interface (can be written directly after the implementation)
    - The statement of the problem chosen to be solved using the given ADT. Describe why the given ADT is suitable for solving the selected problem.
    - The solution for the proposed problem: PSEUDOCODE for every function from the solution.
    - Complexity of the operations from the solution.

- Stage of the project is a **PDF document**, part of the documentation, consisting of the ADT Specification and interface, representation of the container and iterator (but no implementation) and the statement of the selected problem with the explanations why the ADT is suitable for solving the problem.
- Projects stage will have to be uploaded to Moodle

- Projects will be uploaded to Moodle

Project grading:

- 1p - Start
- 1p – ADT Specification and interface
- 3p – ADT Representation + Operation implementation in pseudocode
- 1p – ADT Tests
- 1p – ADT Complexities
- 1p – Problem statement
- 2p – Application implementation and complexities