# Lab 5 – Class templates

## Exercise 1. Pair - Class template

In the file ex1_Pair.cpp the class template Pair<T,S> is defined.

Outside this class in the "implementation section"

- define static member numberOfPairs
- implement all the methods of class template Pair<T,S>, as described in comments
- implement stream insertion operator << as non-member friend (you need also to add forward declarations before class declaration).

## Exercise 2. Static Vector - Class template

Implement a class template Vector<T,N> that has two template parameters:

- T - type of vector elements,
- N - number of elements.

Coefficients of a vector should be stored in a static array.
Implement:

- constructors:
  - default (set coordinates to 0),
  - copy constructor (can be defaulted?),
  - from initialization list,
  - private constructor (do not set coordinates to 0, use artificial parameter),
- access operators to the coefficients by a[index] (both for read and write),
- internal type value_type equal to T and method size() that returns N,
- addition operator. Operations on vectors with incompatible dimensions should be detected during compilation.

Class should be implemented in file staticVector.h.

Question: Is it reasonable to implement move semantics?

## Exercise 3. Dynamic Vector - Partial specialization

Implement a partial specialization of template Vector for N equal to 0 ( i.e. `Vector<T, 0>`). It should store vector elements in a dynamical array (use smart pointers). The number of elements should be provided in a constructor e.g. `Vector<T,0> v(5);`

Make interfaces of the primary template and the specialization the same or at least very similar (if needed, extend also the primary template).

Add method `resize(newSize)` that resizes Vector keeping elements values and adding tailing zeroes if needed. If it is reasonable implement move semantics.

Operator+ should throw an exception if sizes do not match. The exception should be of a user defined type `VectorException` that is derived from runtime_error.

## Exercise 4. Mixed operators

Add explicit conversions between a static Vector and a dynamic Vector and vice versa.

Add explicit conversion from Vector<T,N> to Vector<S,M>.

Implement addition between static and dynamic vectors:

- sum of static and dynamic vector should be a static vector.