TASK 1. IMPLEMENTATION OF A TEMPLATED CLASS

PURPOSE

The main purpose of this work is to learn how to work with templated classes in C++ programming language by implementing own version of basic STL containers.

TASK

The task is to implement your own templated class. For each variant the STL container and its methods to implement are given. It is strictly forbidden to use STL containers in your source code.

Variant	Container	Methods	Modifications
1	Stack, Queue	Stack.push(…) Stack.pop() Stack.top() Stack.size() Stack.empty() Queue.push(…) Queue.pop() Queue.back() Queue.front() Queue.size() Queue.empty()	Stack.clear() – for removing all elements from the stack; Queue.clear() – for removing all elements from the queue.
2	Vector	Brackets [] Vector.push_back(…) Vector.pop_back() Vector.back() Vector.size() Vector.erase() Vector.clear() Vector.resize(…) Vector.assign(…, …)	Vector.sort() – for sorting elements of vector in increasing order.
3	Stack, Queue	*Same as for variant 1*	Stack.reverse() – to reverse an order of elements in stack; Queue.reverse() – to reverse an order of elements in queue.
4	Vector	*Same as for variant 2*	Vector.find_maximum() – returns the value of the maximum element; Vector.find_minimum() – returns the value of the minimum element.
5	Stack, Queue	*Same as for variant 1*	Stack.count(…) – for the given

			value returns the number of occurrences of this value in stack; Queue.count(…) – for the given value returns the number of occurrences of this value in queue.
6	Vector	*Same as for variant 2*	Vector.is_sorted() – returns true if elements are sorted in increasing order, returns false otherwise; Vector.is_unsorted() – returns false if elements are sorted in increasing order, returns true otherwise.
7	Stack, Queue	*Same as for variant 1*	Stack.unique_count() – returns the number of unique elements in stack.
8	Vector	*Same as for variant 2*	Vector.random_shuffle() – randomly rearranges elements of the vector.
9	Stack, Queue	*Same as for variant 1*	Stack.fill(Queue of the same type) – fill the stack with elements of queue in the same order.
10	Vector	*Same as for variant 2*	Vector.find(…) – returns an index of the element with a given value.
11	Stack, Queue	*Same as for variant 1*	Queue.fill(Stack of the same type) – fill the queue with elements of stack in the same order.
12	Vector	*Same as for variant 2*	Vector.remove_dublicates() – removes all non-unique values from the vector.