

## 1 Simple Counting ( template )

[STL/01\_prepare/task01\_count.cpp]

Разработайте класс, содержащий несколько статических функций, позволяющих подсчитывать «частотность» элемента в массивах различных типов. Каждой функции в качестве аргументов передается массив (при необходимости, количество элементов) и искомое значение. Функция подсчитывает, сколько раз искомое значение встречается в массиве и возвращает результат подсчета.

Реализуйте функции, подсчитывающие частотность в массивах типа `int`, `double`, и функцию, подсчитывающую частотность символа в строке `string`.

Пример работы программы:

```
int grade[7] = { 2, 3, 2, 3, 2, 2, 5};  
cout << Счёт::чслсчет(grade, 7, 2); // выведет 4 (количество двоек в  
массиве grade из 7-ми элементов)  
cout << Счёт::стрсчет("Hello, world", 'l'); // выведет 3 (количество  
символов 'l' в строке)
```

## 2 Counting under the condition ( lambda )

[STL/01\_prepare/task02\_count\_condition.cpp]

Создайте структуру `Pupil`, содержащую поля: `string имя`, `int возраст`, `int успеваемость`: значения (2,3,4,5).

Создайте массив `Pupil[]` и заполните его. Разработаете две функции, которые получают на вход массив учеников и количество элементов в массиве.

- Первая, подсчитывает и возвращает количество двоечников
- Вторая, подсчитывает и возвращает количество хорошистов, включая отличников.

### 3 Such a different Sort ( comparator )

[STL/01\_prepare/task03\_sort.cpp]

Реализуйте три функции, которые производят сортировку учеников в массиве методом пузырька:

- По возрастанию имени
- По убыванию возраста
- По убыванию возраста. Если возраст одинаковый – по возрастанию имени.