

Балтийский государственный технический университет  
«ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра И5 «Информационные системы и программная инженерия»

**Лабораторная работа №4**  
по дисциплине «Программирование на языке высокого уровня»  
по теме «STL»

Выполнил:  
Студент Альков В. С.  
Группа И407Б

Преподаватель:  
Кимсанбаев К. А.

Санкт-Петербург  
2021 г.

Задача: Написать шаблон функции, выполняющей указанные в вариативной части задания действия.

Написать программу тестирования шаблонных функций, созданных на основе этого шаблона, с аргументами указанных типов. Разработать шаблон класса, описывающий указанный в вариативной части задания абстрактный тип данных, и написать программу тестирования объектов двух шаблонных классов. Выбор тестируемого метода должен осуществляться с помощью меню.

Типы аргументов int и float.

1. Поиск максимального отрицательного элемента в массиве.
2. Очередь.

## Задание 1

```
#include <vector>
#include <iostream>
/*шаблон функции*/
template<class T>
T maxNegative(std::vector<T> &arr)
{
    T max;
    int i;
    for(i=0; i<arr.size(); i++)
        if(arr[i]<0)
        {
            max = arr[i++];
            break;
        }
    for(; i<arr.size(); i++)
        if(arr[i]>max && arr[i]<0)
            max = arr[i];
    return max;
}

int main()
{
    int choice, count;
    std::vector<int> arr1;
    std::vector<float> arr2;
    setlocale(LC_ALL, "rus");
    std::cout << "1. Тип int\n";
    std::cout << "2. Тип float\n";
    std::cin >> choice;
    std::cout << "Введите кол-во элементов массива: ";
    std::cin >> count;
    /*проверяем ввод*/
    if(choice!=1 && choice != 2 || count<0)
    {
        std::cout<<"Неправильный ввод\n";
        return 0;
    };
    if(choice == 1)
    {
        int num;
        std::cout<<"Введите "<<count<<" элементов: ";
        /*заполняем вектор*/
        for(int i = 0; i<count; i++)
```

```

        {
            std::cin>>num;
            arr1.push_back(num);
        };
        std::cout<<"Если 0, значит нет отрицательных\nНаибольший из отрицательных: " << maxNegative(arr1);
        /*очищаем вектор*/
        std::vector<int> empty;
        std::swap(arr1, empty);
    }
    else
    {
        float num;
        std::cout<<"Введите "<<count<<" элементов: ";
        /*заполняем вектор*/
        for(int i = 0; i<count; i++)
        {
            std::cin>>num;
            arr2.push_back(num);
        };
        std::cout<<"Если 0, значит нет отрицательных\nНаибольший из отрицательных: " << maxNegative(arr2);
        /*очищаем вектор*/
        std::vector<float> empty;
        std::swap(arr2, empty);
    };
    return 0;
}

```

### Результат работы программы

|   |  |
|---|--|
| <pre> 1. Тип int 2. Тип float 1 Введите кол-во элементов массива: 5 Введите 5 элементов: 0 1 2 3 4 Если 0, значит нет отрицательных Наибольший из отрицательных: 0 Process returned 0 (0x0)   execution time: 0.000 s Press any key to continue. </pre> | <pre> 1. Тип int 2. Тип float 1 Введите кол-во элементов массива: 5 Введите 5 элементов: -5 -3 -4 -2 0 Если 0, значит нет отрицательных Наибольший из отрицательных: -2 Process returned 0 (0x0)   execution time: 0.000 s Press any key to continue. </pre> |
|---|--|

|   |  |
|---|--|
| <pre> 1. Тип int 2. Тип float 2 Введите кол-во элементов массива: 5 Введите 5 элементов: 1.1 5.1 6.1 7.1 4.4 Если 0, значит нет отрицательных Наибольший из отрицательных: 0 Process returned 0 (0x0)   execution time: 0.000 s Press any key to continue. </pre> | <pre> 1. Тип int 2. Тип float 2 Введите кол-во элементов массива: 5 Введите 5 элементов: 0.1 -1.5 6 -2.2 6.3 Если 0, значит нет отрицательных Наибольший из отрицательных: -1.5 Process returned 0 (0x0)   execution time: 0.000 s Press any key to continue. </pre> |
|---|--|

## Задание 2

```

#include <iostream>
#include <queue>
/*шаблон класса, на основе std::queue*/
template<class T> class Queue
{
    std::queue<T> object;
public:
    /*конструктор*/
    Queue(){};
    /*деструктор*/
    ~Queue(){std::queue<T> empty; std::swap(object, empty);};

```

```

/*проверка на пустоту*/
bool isEmpty(){return object.empty();};
/*получение первого элемента*/
T Front(){return object.front();};
/*добавление элемента*/
void EnQueue(T x){object.push(x);};
/*извлечение элемента*/
T DeQueue(){T num = Front(); object.pop(); return num;};
};
/*шаблон ф-ии меню*/
template<class T>
void menu(Queue<T> *object)
{
    setlocale(LC_ALL, "rus");
    int menu;
    T num;
    do
    {
        system("cls");
        std::cout<<"1. Добавить в очередь\n";
        std::cout<<"2. Неразрушающее чтение\n";
        std::cout<<"3. Убрать из очереди\n";
        std::cout<<"4. Проверить на пустоту\n";
        std::cout<<"5. Выйти\n";
        std::cin>>menu;
        getchar();
        switch(menu)
        {
            case 1: std::cin>>num;
                    object->EnQueue(num);
                    break;
            case 2: if(object->isEmpty())
                    std::cout<<"Очередь пуста\n";
                    else
                    std::cout<<object->Front();
                    break;
            case 3: if(object->isEmpty())
                    std::cout<<"Очередь пуста\n";
                    else
                    std::cout<<object->DeQueue();
                    break;
            case 4: std::cout<<object->isEmpty();
                    break;
            case 5: break;
            default: std::cout<<"Неправильный ввод\n";
                    break;
        };
        getchar();
    }
    while(menu!=5);
};

int main()
{
    Queue<int> a;
    Queue<float> b;
    int choice;
    setlocale(LC_ALL, "rus");
    std::cout << "1. Тип int\n";
    std::cout << "2. Тип float\n";
    std::cin >> choice;
    /*проверка ввода*/
    if(choice != 1 && choice != 2)
    {

```

```

        std::cout << "Неправильный ввод";
        return 0;
    };
    if (choice == 1)
        menu(&a);
    else
        menu(&b);
    return 0;
}

```

## Результат работы программы

### Int очередь

```

1. Тип int
2. Тип float
1

```

### 4. Проверить на пустоту

```

1. Добавить в очередь
2. Неразрушающее чтение
3. Убрать из очереди
4. Проверить на пустоту
5. Выйти
4
1

```

### 1. Добавить в очередь

```

1. Добавить в очередь
2. Неразрушающее чтение
3. Убрать из очереди
4. Проверить на пустоту
5. Выйти
1
10

```

```

1. Добавить в очередь
2. Неразрушающее чтение
3. Убрать из очереди
4. Проверить на пустоту
5. Выйти
1
12

```

```

1. Добавить в очередь
2. Неразрушающее чтение
3. Убрать из очереди
4. Проверить на пустоту
5. Выйти
1
22

```

### 4. Проверить на пустоту

```

1. Добавить в очередь
2. Неразрушающее чтение
3. Убрать из очереди
4. Проверить на пустоту
5. Выйти
4
0

```

### 3. Убрать из очереди

```

1. Добавить в очередь
2. Неразрушающее чтение
3. Убрать из очереди
4. Проверить на пустоту
5. Выйти
3
10

```

```

1. Добавить в очередь
2. Неразрушающее чтение
3. Убрать из очереди
4. Проверить на пустоту
5. Выйти
3
12

```

```

1. Добавить в очередь
2. Неразрушающее чтение
3. Убрать из очереди
4. Проверить на пустоту
5. Выйти
3
22

```

```

1. Добавить в очередь
2. Неразрушающее чтение
3. Убрать из очереди
4. Проверить на пустоту
5. Выйти
3
Очередь пуста

```

### 5. Выйти

```

1. Добавить в очередь
2. Неразрушающее чтение
3. Убрать из очереди
4. Проверить на пустоту
5. Выйти
5

```

```

Process returned 0 (0x0)
Press any key to continue.

```