Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования "Уфимский государственный авиационный технический университет"

Кафедра Высокопроизводительных вычислительных технологий и систем

Дисциплина: Программирование

Отчет по лабораторной работе №10

Тема: «Шаблоны и STL»

Группа ПМ-153	Фамилия И.О.	Подпись	Дата	Оценка
Студент	Шамаев И.Р			
Принял	Гайнетдинова А.А.			

Уфа 2020

Цель: изучить механизмы создания шаблонов функций и классов, а также ознакомиться с библиотекой STL.

Теоретический материал

Ответы на контрольные вопросы

1) Что такое шаблон функции? Шаблоны функций -– это обобщенное описание поведения функций, которые могут вызываться для объектов разных типов.
2) Как создать и использовать шаблонную функцию? Обобщённая функция создаётся с помощью ключевого слова template. Общий формат определения шаблонной функции имеет следующий вид:

```
template < class Ttype> тип имя_функции (список_параметров)
{
    //тело функции
}
```

3) Что такое шаблон класса и когда он нужен? Шаблоны классов -– обобщенное описание пользовательского типа, в котором могут быть параметризованы атрибуты и операции типа. Представляют собой конструкции, по которым могут быть сгенерированы действительные классы путём подстановки вместо параметров конкретных аргументов.
4) Что такое STL? Библиотека STL (Standard Template Library) — это набор шаблонных классов и функций общего назначения.
5) Перечислите основные компоненты STL. Ядро стандартной библиотеки шаблонов включает три основных элемента: контейнеры, алгоритмы и итераторы.
   - Контейнеры — это объекты, которые содержат другие объекты.
   - Алгоритмы обрабатывают содержимое контейнеров.
   - Итераторы подобны указателям.
6) Какие контейнеры относятся к последовательным? Какие – к ассоциативным? Последовательные: vector, list, deque. Ассоциативные: set, multiset, map, multimap.
7) Какие особенности использованных в Вашей лабораторной работе контейнеров Вы знаете? В отличие от других контейнеров для типа list не определена операция обращения по индексу или функция at(), которая выполняет похожую задачу. Контейнер map, очень похож на остальные контейнеры, такие как vector, list, deque, но с небольшим отличием. В этот контейнер можно помещать сразу два значения.

8) Что такое итератор? Итератор — интерфейс, предоставляющий доступ к элементам коллекции (массива или контейнера) и навигацию по ним.

Индивидуальное задание №1

Задание:

Создайте шаблонную функцию, позволяющую обрабатывать объект класса `list` (двусвязный список) с элементами заранее не определенного типа. Функция должна осуществлять поиск номера элемента с минимальным значением.

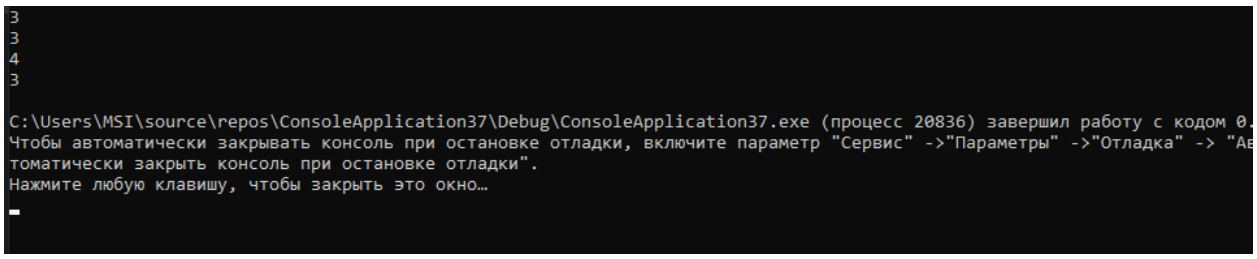
Напишите демонстрационное приложение, показывающее работу функции для трех различных типов данных (например, целые и вещественные числа различной точности, символы, строки).

Исходный код программы

```
#include <iostream>

using namespace std;

template <typename list>
void Min(const list* a, int count)
{
    list min_num = 100;
    int min_i = 100;
    for (int i = 0; i < count - 1; i++)
    {

        if (a[i] < min_num)
        {
            min_num = a[i];
            min_i = i;

        }
        /*
        if (static_cast<list> (a[i]) < min_num) {
            min_num = static_cast<list>(a[i]);
            min_i = i;
        }
      */
    }
    //cout << min_num << endl;
    cout << min_i << endl;

}


int main()
{
    const int A_Size = 6, B_Size = 5, C_Size = 6, D_Size = 5;
    int A_arr[A_Size] = { 1, 2, 4, 0, 6 };
    double B_arr[B_Size] = { 1.023, 4.525, 0.996 , 0.01 };
    float C_arr[C_Size] = { 2.2  , 3.3, 4.4, 5.5, 0.1 };
    char D_arr[D_Size] = { 'L', 'P', 'H', 'A', '\0' };

    Min(A_arr, A_Size);
    Min(B_arr, B_Size);
    Min(C_arr, C_Size);
    Min(D_arr, D_Size);


    return 0;
```

```
}
```

Пример выполнения программы



Индивидуальное задание №2

Задание:

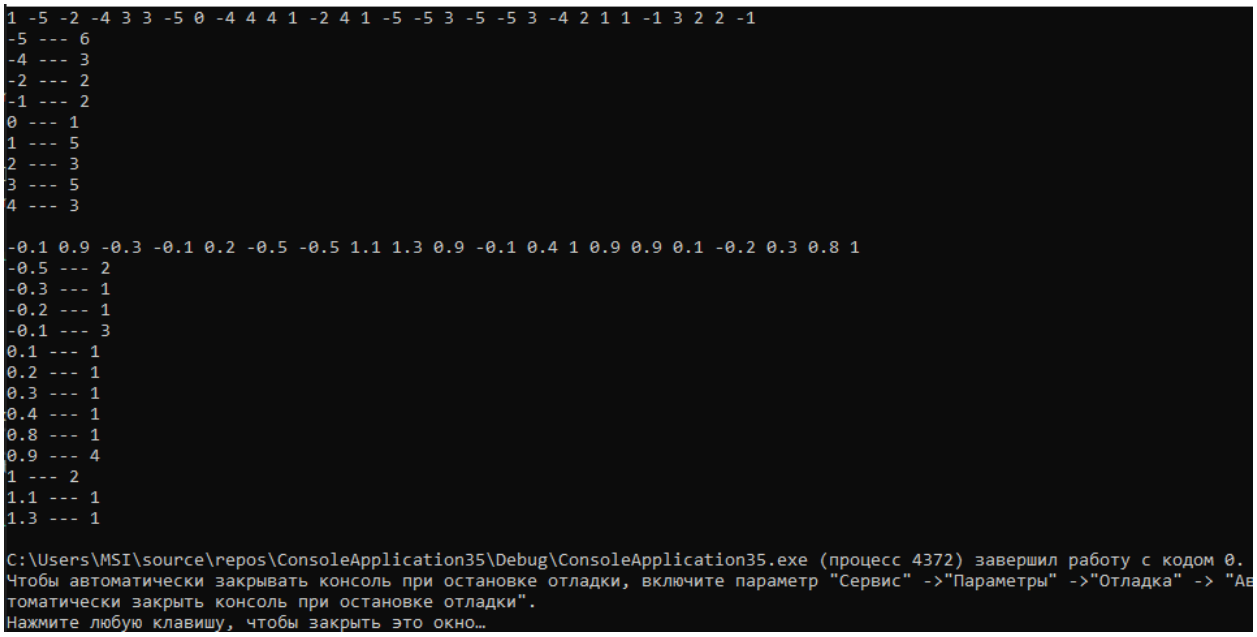
Создайте функцию, принимающую в качестве аргумента объект класса deque (двусторонняя очередь). Функция должна выполнять подсчет и печать на экран частоты встречаемости элементов объекта. При решении необходимо использовать подходящий ассоциативный контейнер.

Исходный код программы

```
#include <iostream>
#include <deque>
#include <map>
#include <time.h>

using namespace std;

template <typename Type>//создание шаблона
void Info(deque <Type> Deque)
{
	map <Type, int> Container;//создание контейнера map для хранения значений

	for (typename deque <Type>::iterator i = Deque.begin(); i != Deque.end();
i++)
	{
		cout << *i << " ";//распечатка массива
		for (typename deque <Type>::iterator j = Deque.begin(); j !=
Deque.end(); j++)
		{
			if (*i == *j)
			{
				Container[*i] += 1;//подсчет количества элементов
			}
		}
	}
	cout << endl;

	//typename map <Type, int>::iterator it = Container.begin();
	for (typename map <Type, int>::iterator it = Container.begin(); it !=
Container.end(); it++)
	{
		cout <<  it->first << " --- " << sqrt(it->second) <<
endl;//распечатка map'a
	}
}

int main()

{
	setlocale(LC_ALL, "Rus");
	srand(time(NULL));

	deque <int> TestInt(30);
	for (deque <int>::iterator i = TestInt.begin(); i != TestInt.end(); i+
+)//генерация элементов deque
	{
		*i = -5 + rand() % 10;
	}

	Info(TestInt);

	cout << endl;
```

```
	deque <double> TestD(20);

	for (deque <double>::iterator i = TestD.begin(); i != TestD.end(); i++)//генерация элементов deque
	{
		*i = (double)(-5 + rand() % 20) / 10;
	}

	Info(TestD);
	return 0;
}
```

Пример выполнения программы



Вывод: я изучил механизмы создания шаблонов функций и классов, а также ознакомился с библиотекой STL.