**Курс «Объектно-ориентированное программирование на C++»**

**Встреча №29**

Тема: STL часть 1

Задания для самостоятельной работы:

**Задание №1.**

В контейнере типа vector хранится набор целых чисел. Создайте клас для решения следующих задач:

1. Поиск минимального значения
2. Поиск максимального значения
3. Сортировка данных по убыванию
4. Сортировка данных по возрастанию
5. Увеличение значений в контейнере на заданную константу
6. Уменьшение значений в контейнере на заданную константу
7. Удаление элементов из контейнера равных искомому значению
8. Поиск количества элементов равных заданному

**Задание №2.**

Создать и протестировать функцию void parseString(const string &src, vector &dst), которая сформирует массив множеств строк dst из строки src по заданному правилу.

Завдання 1

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include<string>

#include<time.h>

#include<vector>

using namespace std;

template <class T>

class MyVector

{

private:

vector<T> arr;

public:

void Add(T elem)

{

arr.push\_back(elem);

}

T &Minimal\_Elem()

{

T min = arr[0];

for (int i = 1; i < arr.size(); i++)

{

if (min > arr[i])

min = arr[i];

}

return min;

}

T& Maximum\_Elem()

{

T min = arr[0];

for (int i = 1; i < arr.size(); i++)

{

if (min < arr[i])

min = arr[i];

}

return min;

}

void Show()

{

for (int i = 0; i < arr.size(); i++)

{

cout << arr[i] << endl;

}

}

void Sort\_By\_Minimal()

{

bool b;

do {

b = false;

for (int i = 0; i < arr.size() - 1; i++)

{

if (arr[i] > arr[i + 1])

{

T elem;

elem = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = elem;

b = true;

}

}

} while (b);

cout << "Sort complete!" << endl;

}

void Sort\_By\_Maximum()

{

bool b;

do {

b = false;

for (int i = 0; i < arr.size() - 1; i++)

{

if (arr[i] < arr[i + 1])

{

T elem;

elem = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = elem;

b = true;

}

}

} while (b);

cout << "Sort complete!" << endl;

}

void Add\_Memory\_By\_Const\_Value(int value)

{

for (int i = 0; i < value; i++)

{

arr.push\_back(rand() % 100);

}

}

void Delete\_Memory\_By\_Const\_Value(int value)

{

for (int i = 0; i < value; i++)

{

arr.pop\_back();

}

}

void Delete\_Elements\_By\_Value(T elem)

{

for (int i = 0; i < arr.size(); i++)

{

if (elem == arr[i])

arr.erase(arr.begin() + i);

}

}

int Number\_Of\_Element\_By\_Value(T elem)

{

int k = 0;

for (int i = 0; i < arr.size(); i++)

{

if (elem == arr[i])

k++;

}

return k;

}

};

int main()

{

srand(time(0));

MyVector<int> v;

v.Add(-4);

v.Add(5);

v.Add(3);

v.Add(-4);

v.Add(0);

v.Show();

cout << "---------------\n";

cout << v.Number\_Of\_Element\_By\_Value(-4);

/\*cout << v.Minimal\_Elem()<<endl;

cout << v.Maximum\_Elem()<<endl;\*/

return 0;

}

Завдання 2

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include<string>

#include<time.h>

#include<vector>

using namespace std;

void parseString(const string& src, vector<string>& dst)

{

char\* copy = new char[src.size() + 1];

strcpy(copy, src.c\_str());

char separator[] = "\n ";

char\* pstr;

pstr = strtok(copy, separator);

while (pstr != NULL)

{

dst.push\_back(pstr);

pstr = strtok(NULL, separator);

}

delete[]copy;

}

int main()

{

srand(time(0));

vector<string> dst;

string txt;

getline(cin, txt);

parseString(txt, dst);

for (int i = 0; i < dst.size(); i++)

{

cout << dst[i] << endl;

}

return 0;

}