Балтийский государственный технический университет «ВОЕНМЕХ»

им. Д. Ф. Устинова

Кафедра И5

«Информационные системы и программная инженерия»

Практическое задание № 3

по дисциплине «Программирование на ЯВУ»

на тему «Шаблоны»

Выполнил:

Студент Дубровский В.И.

Группа И582

Преподаватель:

Спирин Д.О.

Санкт-Петербург

2019 г.

**Вариант 8**

Цель работы – научиться создавать шаблоны функций и шаблоны классов для работы с любыми типами данных без переписывания кода программы.

Задание 1.

Типы аргументов float и char.

Перестановка элементов в массиве следующим образом: сначала записать неотрицательные элементы в том же порядке следования, затем отрицательные в том же порядке.

Текст программы:

main.cpp

#include <iostream>

#include <conio.h>

#include <iterator>

using namespace std;

template<class T>

void array\_changer(T \*first, int size)

{

int flag=-1;

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

{

if (first[i] < 0 && flag < 0) {

flag = i;

}

else if (flag >= 0 && first[i] >= 0)

{

T temp;

for (int j = i; j > flag; j--)

{

temp = first[j - 1];

first[j - 1] = first[j];

first[j] = temp;

}

i-=flag+1;

flag = -1;

}

}

}

int main()

{

float \*f;

char \*c;

int s1, s2;

cout << "Enter size of array float: ";

while (!(cin >> s1) || s1<0)

{

cout << "Wrong input" << endl;

cout << "Enter size of array float: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

f = new float[s1];

for (int i = 0; i < s1; i++)

{

cout << "f[" << i+1 << "] = ";

while (!(cin >> f[i]))

{

cout << "Wrong input" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << "f[" << i+1 << "] = ";

}

}

array\_changer(f, s1);

for (int i = 0; i < s1; i++)

{

cout << f[i] << '\t';

}

cout << '\n';

cout << "Enter size of array char: ";

while (!(cin >> s2) || s2<0)

{

cout << "Wrong input" << endl;

cout << "Enter size of array char: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

c = new char[s2];

for (int i = 0; i < s2; i++)

{

int temp;

cout << "c[" << i+1 << "] = ";

while (!(cin >> temp))

{

cout << "Wrong input" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << "c[" << i+1 << "] = ";

}

c[i] = temp;

}

array\_changer(c, s2);

for (int i = 0; i < s2; i++)

{

cout << static\_cast<int>(c[i]) << '\t';

}

\_getch();

return 0;

}  
Результат работы программы:

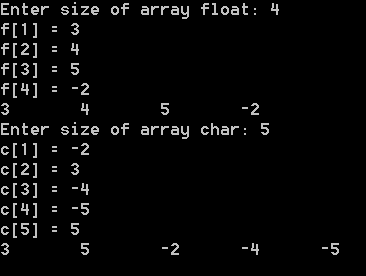


Рисунок 1.

Задание 2.

АТД Стек. Структура хранения векторная. Сложность - повышенный. Создать требуемый АТД с помощью двух структур хранения: векторной и списковой,– реализацию оформить в виде шаблонов классов с единым интерфейсом.

Текст программы:

*Векторная СХ.*

mystack\_vector.h

#pragma once

template <class T>

class mystack\_v

{

private:

T \*data;

int maxlen;

int top;

public:

mystack\_v(int n=100);

~mystack\_v(void);

const bool push(T);

T pop(void);

T read(void);

const bool Empty(void);

};

template<class T>

inline mystack\_v<T>::mystack\_v(int n) : maxlen(n), top(-1)

{

data = new T[maxlen];

}

template<class T>

inline mystack\_v<T>::~mystack\_v(void)

{

delete[] data;

}

template<class T>

const bool mystack\_v<T>::push(T x)

{

if (top == maxlen - 1) return false;

data[++top] = x;

return true;

}

template<class T>

inline T mystack\_v<T>::pop(void)

{

return data[top--];

}

template<class T>

inline T mystack\_v<T>::read(void)

{

return data[top];

}

template<class T>

inline const bool mystack\_v<T>::Empty(void)

{

return top<0;

}

main.cpp

#include <iostream>

#include <conio.h>

#include "mystack\_vector.h"

#include <string>

using namespace std;

template <class T>

void menu\_func(string type)

{

mystack\_v<T> \*ptr = new mystack\_v<T>;

int c = -1;

do {

system("cls");

cout << "\tMenu stack+<" << type << '>' << endl

<< "1 - push" << endl

<< "2 - pop" << endl

<< "3 - read" << endl

<< "0 - exit" << endl;

while ((c<'0' || c>'4') && (c = \_getch()));

if (c == '0') break;

switch (c)

{

case '1': {

T temp;

cout << "Enter data: ";

while (!(cin >> temp))

{

cout << "Incorrect input!" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << "Enter data: ";

}

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

if (!ptr->push(temp)) cout << "Stack is full";

break;

}

case '2':

if (ptr->Empty()) {

cout << "Empty" << endl;

break;

}

cout << "pop:" << ptr->pop() << endl;

break;

case '3':

if (ptr->Empty()) {

cout << "Empty" << endl;

break;

}

cout << "read:" << ptr->read() << endl;

break;

}

while (\_getch() != 13);

c = -1;

} while (true);

delete ptr;

}

int main()

{

int c = -1;

while (true)

{

cout << "List of datatypes:" << endl

<< "1 - float" << endl

<< "2 - double" << endl

<< "3 - int" << endl

<< "4 - unsigned int" << endl

<< "5 - char" << endl

<< "6 - string" << endl

<< "0 - exit" << endl;

cout << "Press num of datatype";

while ((c<'0' || c>'6') && (c = \_getch()));

if (c == '0') break;

switch (c)

{

case '1':

menu\_func<float>("float");

break;

case '2':

menu\_func<double>("double");

break;

case '3':

menu\_func<int>("int");

break;

case '4':

menu\_func<unsigned int>("unsigned int");

break;

case '5':

menu\_func<char>("char");

break;

case '6':

menu\_func<string>("string");

}

c = -1;

system("cls");

}

return 0;

}

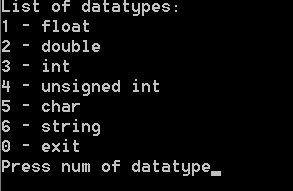


Рисунок 1.

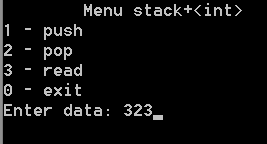


Рисунок 2.

*Линейная СХ.*

mystack.h

#pragma once

template<class Temp>

class mystack

{

struct list{

Temp data;

struct list \*next;

}\*end;

public:

mystack();

~mystack();

int push(Temp);

Temp pop(void);

Temp read(void);

const bool Empty(void);

};

///////////////////

//////////

///////////////////

template<class Temp>

mystack<Temp>::mystack()

{

end = nullptr;

}

template<class Temp>

mystack<Temp>::~mystack()

{

struct list \*temp;

while (end)

{

temp = end;

end = end->next;

delete temp;

}

}

template<class Temp>

int mystack<Temp>::push(Temp x)

{

struct list \*help = new (std::nothrow) struct list;

if (help == nullptr)

return 0;

help->next = end;

help->data = x;

end = help;

return 1;

}

template<class Temp>

Temp mystack<Temp>::pop(void)

{

struct list \*help = end;

Temp data = help->data;

end = end->next;

delete help;

return data;

}

template<class Temp>

Temp mystack<Temp>::read(void)

{

return end->data;

}

template<class Temp>

const bool mystack<Temp>::Empty(void)

{

return end == nullptr;

}

main.cpp

#include <iostream>

#include <conio.h>

#include "mystack.h"

#include <string>

using namespace std;

template <class T>

void menu\_func(string type)

{

mystack<T> \*ptr = new mystack<T>;

int c=-1;

do {

system("cls");

cout << "\tMenu stack+<" << type << '>' << endl

<< "1 - push" << endl

<< "2 - pop" << endl

<< "3 - read" << endl

<< "0 - exit" << endl;

while ((c<'0' || c>'4') && (c = \_getch()));

if (c == '0') break;

switch (c)

{

case '1': {

T temp;

cout << "Enter data: ";

while (!(cin >> temp))

{

cout << "Incorrect input!" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << "Enter data: ";

}

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

if(!ptr->push(temp)) cout << "Stack is full";

break;

}

case '2':

if (ptr->Empty()) {

cout << "Empty" << endl;

break;

}

cout << "pop:" << ptr->pop() << endl;

break;

case '3':

if (ptr->Empty()) {

cout << "Empty" << endl;

break;

}

cout << "read:" << ptr->read() << endl;

break;

}

while (\_getch() != 13);

c = -1;

} while (true);

delete ptr;

}

int main()

{

int c=-1;

while (true)

{

cout << "List of datatypes:" << endl

<< "1 - float" << endl

<< "2 - double" << endl

<< "3 - int" << endl

<< "4 - unsigned int" << endl

<< "5 - char" << endl

<< "6 - string" << endl

<< "0 - exit" << endl;

cout << "Press num of datatype";

while ((c<'0' || c>'6') && (c = \_getch()));

if (c == '0') break;

switch (c)

{

case '1':

menu\_func<float>("float");

break;

case '2':

menu\_func<double>("double");

break;

case '3':

menu\_func<int>("int");

break;

case '4':

menu\_func<unsigned int>("unsigned int");

break;

case '5':

menu\_func<char>("char");

break;

case '6':

menu\_func<string>("string");

}

c = -1;

system("cls");

}

return 0;

}