БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

ИИТ БГУИР

Программное обеспечение информационных технологий

Отчет

По лабораторной работе №5

«Шаблоны»

по курсу «ООТПиСП»

Выполнил студент группы 381064 Сорока А.А.

Проверил преподаватель: Лашкевич Е.М.

Минск 2015 г

**Цель работы** – Изучение правил объявления и определения шаблонов функций и классов в языке С++. Изучение способов переопределения шаблонов функций. Изучение методов альтернативного определения шаблонов классов и функций.

**Задание А4.** Имеется элемент очереди и очередь (дисциплина обслуживания FIFO):

template <class T>

struct QueueItem

{

T \* data; //указатель на данные

QueueItem<T> \* next; //указатель на следующий элемент

};

template <class T>

class Queue

{

QueueItem<T> \* queue;

...

};

**Код программы:**

Main.h:

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

template <typename T>

struct QueueItem

{

T\* data = NULL;

QueueItem<T> \* next = NULL;

};

template <typename T>

class Queue

{

public:

Queue();

~Queue();

void Push(T\* data);

T\* Pop();

void View();

private:

QueueItem<T> \*queue = new QueueItem<T>;

void AddItem(QueueItem<T>\* st, T\* data);

void DeleteItem();

void ViewQueue(QueueItem<T> \*st);

};

Main.cpp:

#include "main.h"

int main(int argc, char \*argv[])

{

//целый стек

Queue <int> q;

int \*a = new int;

\*a = 123;

q.Push(a);

a = new int;

\*a = 456;

q.Push(a);

//просматриваем элементы

cout << "Queue int:" << endl;

q.View();

cout << endl;

a = q.Pop();

//достанем один и посмотри на его данные

cout << "Pop " << \*a << endl;

//просматриваем элементы

cout << endl << "Queue int:" << endl;

q.View();

cout << endl << endl;

Queue <double> d;

double \*dd = new double;

\*dd = 4.5;

d.Push(dd);

dd = new double;

\*dd = 6.234;

d.Push(dd);

//просматриваем элементы

cout << "Queue double:" << endl;

d.View();

cout << endl;

dd = d.Pop();

//достанем один и посмотри на его данные

cout << "Pop " << \*dd << endl;

//просматриваем элементы

cout << endl << "Queue double:" << endl;

d.View();

system("pause");

return 0;

}

template <typename T>

Queue<T>::Queue()

{

}

template <typename T>

Queue<T>::~Queue()

{

if (queue->data != NULL)

{

while (queue->next != NULL)

{

DeleteItem();

}

}

delete queue;

}

template <typename T>

T\* Queue<T>::Pop()

{

T\* tmp;

if (queue->data == NULL)

{

return NULL;

}

tmp = queue->data;

DeleteItem();

return tmp;

}

template <typename T>

void Queue<T>::View()

{

ViewQueue(queue);

}

template <typename T>

void Queue<T>::Push(T\* data)

{

AddItem(queue, data);

}

template <typename T>

void Queue<T>::AddItem(QueueItem<T>\* st, T\* data)

{

if (st->data == NULL)

{

st->data = data;

}

else

{

if (st->next == NULL)

{

st->next = new QueueItem<T>;

}

AddItem(st->next, data);

}

}

template <typename T>

void Queue<T>::ViewQueue(QueueItem<T>\* st)

{

if (st->data != NULL)

{

cout << \*st->data << endl;

}

if (st->next != NULL)

{

ViewQueue(st->next);

}

}

template <typename T>

void Queue<T>::DeleteItem()

{

if (queue->data != NULL)

{

if (queue->next != NULL)

{

QueueItem<T> \*tmp;

tmp = queue->next;

delete queue;

queue = tmp;

}

else

{

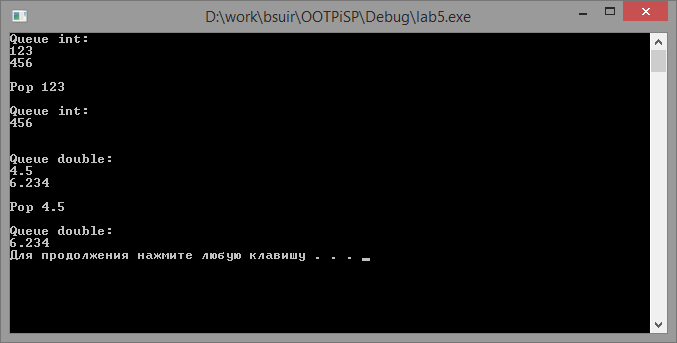
queue->data = NULL;

}

}

}

**Скриншот выполнения программы:**



**Вывод**: В ходе выполнения лабораторной работы я изучил правила объявления и определения шаблонов функций и классов в языке С++. Изучил способы переопределения шаблонов функций. Изучил методы альтернативного определения шаблонов классов и функций.