*Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования*

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет***  ***имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ: Информатики и систем управления

КАФЕДРА: Компьютерные системы и сети

**Отчет**

**по усложнённой лабораторной работе №** 8

**Название лабораторной работы:** *Программирование с использованием классов С++. Создание контейнеров.*

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент гр. ИУ6-22 |  |  | | Боярских Никита |
|  | (Подпись, дата) | |  | (И. О. Фамилия) |
| Преподаватель |  | |  | Черноусова Т. Г. |
|  | (Подпись, дата) | |  | (И. О. Фамилия) |

**Вариант 1**

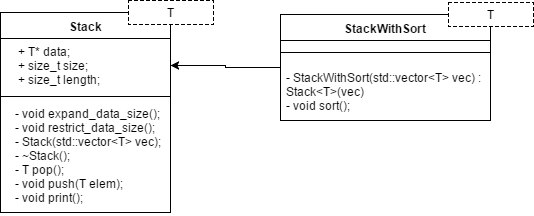
Москва, 2017

**Задание:**

Моделировать стек, в качестве элементов которого могут использоваться числа и символы. Операции: добавление элемента, удаление элемента, печать элементов стека. Создать класс - потомок, который добавляет процедуру сортировки элементов стека. Тестировать полученную модель. Примечание: сначала реализовать и отладить структуру данных как класс, после чего преобразовать в класс шаблон.

В отчете представить диаграмму классов и обосновать выбранную структуру представления данных.

Для поставленной задачи былиразработаны шаблон и классы, представленные на диаграмме (см. рис. 1)

**

*Рисунок 1 - диаграмма классов программы, моделирующей стек*

**Код программы, моделирующей стек**

#include <iostream>

#include <vector>

#include <algorithm>

#include <cstring>

template <class T>

class Stack {

protected:

T\* data;

size\_t size;

size\_t length;

void expand\_data\_size();

void restrict\_data\_size();

public:

Stack(std::vector<T> vec);

~Stack();

T pop();

void push(T elem);

void print();

};

(Продолжение кода на следующей странице)

(Продолжение кода)

template <class T>

class StackWithSort : public Stack<T> {

public:

StackWithSort(std::vector<T> vec) : Stack<T>(vec) {};

void sort();

};

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

{

std::vector<int> ivec = {3, -5, 2, 8, -1, 45};

StackWithSort<int> istack(ivec);

istack.pop();

istack.push(3);

std::cout <<"Int stack contains: ";

istack.print();

istack.sort();

std::cout <<"Int stack after sorting: ";

istack.print();

std::vector<char> cvec = {'C', 'H', 'E', 'B', 'U', 'R', 'E', 'B'};

StackWithSort<char> cstack(cvec);

cstack.pop();

cstack.push('K');

std::cout <<"Char stack contains: ";

cstack.print();

cstack.sort();

std::cout <<"Char stack after sorting: ";

cstack.print();

return 0;

}

template <class T>

Stack<T>::Stack(std::vector<T> vec)

{

size = vec.size();

length = size;

data = new T[vec.size()];

if(!data) {

throw std::runtime\_error("Bad allocation");

}

for( size\_t i = 0; i < vec.size(); i++ ) {

data[i] = vec[i];

}

}

template <class T>

void Stack<T>::push(T elem)

{

if( length + 1 > size ) expand\_data\_size();

data[length] = elem;

length++;

}

template <class T>

T Stack<T>::pop()

{

if( length - 1 < size/4 ) restrict\_data\_size();

length--;

return data[length];

}

template <class T>

void Stack<T>::print()

{

(Продолжение кода на следующей странице)

(Продолжение кода)

for( size\_t i = 0; i < length; i++ ) std::cout << data[i] << ' ';

std::cout << std::endl;

}

template <class T>

Stack<T>::~Stack()

{

if(data) delete[] data;

}

template <class T>

void Stack<T>::expand\_data\_size() {

T \*temp = new T[size \*= 2];

if(!temp) {

delete[] data;

throw std::runtime\_error("Bad allocation");

}

memmove(temp, data, size\*sizeof(data[0]));

delete[] data;

data = temp;

}

template <class T>

void Stack<T>::restrict\_data\_size() {

T \*temp = new T[size /= 2];

if(!temp) {

delete[] data;

throw std::runtime\_error("Bad allocation");

}

memmove(temp, data, size\*sizeof(data[0]));

delete[] data;

data = temp;

}

template <class T>

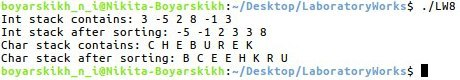
void StackWithSort<T>::sort()

{

std::sort(this->data, this->data + this->length);

}

Проведено тестирование программы (см рис. 2)



*Рисунок 2 - тестирование программы, моделирующей стек*

**Вывод:**

1) Разработаны шаблоны и классы для решения поставленной задачи. Составлена ихдиаграмма в среде [draw.io](http://draw.io) (рис. 1)

2) Создан код программы в среде Vim

4) Проведено тестирование (см. табл. 1)

5) Тестирование показало корректность работы при заданных исходных данных (см. табл. 1 и рис. 2 - 4)