Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе 19**

Тема: Стеки через классы

Выполнил работу

студент группы РИС-20-1б

Томилов В.А.

Проверила

Доцент кафедры ИТАС

к.т.н ПоляковаО.А.

Пермь, 2021

Постановка задачи

Анализ задачи

* 1. Для решения задачи необходимо
  2. Добавить в проект файл stack.h, содержащий описание класса.

class Stack {

private:

struct MyStack {

int Number;

MyStack\* Next;

};

int amountOfElements;

MyStack\* StackForIntNumbers;

public:

Stack()

~Stack()

Stack(const int size)

void make\_stack(const int size)

void push(int element)

void push(int element)

void pop()

void clear()

void delete\_even\_elements()

bool isEmpty()

int top()

};

* 1. Добавим в проект файл stack.cpp, содержащий описание методов класса stack.

#include<iostream>

#include ”Stack.h”

using namespace std;

Stack() {

amountOfElements = 0;

}

Stack(const int size) {

amountOfElements = size;

}

void make\_stack(const int size) {

int number;

amountOfElements = size;

StackForIntNumbers = new MyStack;

StackForIntNumbers->Next = NULL;

cout << "Введите элементы:\n";

if (amountOfElements > 0) {

cout << "1) ";

cin >> number;

StackForIntNumbers->Number = number;

}

for (int i = 2; i <= amountOfElements; i++) {

cout << i << ") ";

cin >> number;

push(number);

}

}

void push(int element) {

MyStack\* ptr\_on\_new\_element = new MyStack;

ptr\_on\_new\_element->Next = NULL;

ptr\_on\_new\_element->Number = element;

ptr\_on\_new\_element->Next = StackForIntNumbers;

StackForIntNumbers = ptr\_on\_new\_element;

}

void print\_stack() {

MyStack\* pointer = StackForIntNumbers;

if (StackForIntNumbers == NULL)

cout << "Стек пуст\n";

else

while (pointer != NULL) {

cout << pointer->Number << endl;

pointer = pointer->Next;

}

}

void pop() {

MyStack\* ptr\_on\_deleting\_element = StackForIntNumbers;

StackForIntNumbers = StackForIntNumbers->Next;

delete ptr\_on\_deleting\_element;

}

void clear() {

while (StackForIntNumbers != NULL)

pop();

}

void delete\_even\_elements() {

Stack temp\_stack;

for (int i = 0; i < amountOfElements; i++) {

if ((StackForIntNumbers->Number % 2) == 0)

pop();

else {

temp\_stack.push(StackForIntNumbers->Number);

pop();

}

}

while (!temp\_stack.isEmpty()) {

push(temp\_stack.top());

temp\_stack.pop();

}

}

bool isEmpty() {

return (StackForIntNumbers == NULL);

}

int top() {

return (StackForIntNumbers->Number);

}

~Stack() {

clear();

}

};

* 1. Добавить в проект файл Lab1\_main.cpp, содержащий основную программу.

int main()

{

setlocale(LC\_ALL, "");

int size;

cout << "Введите количество элементов: ";

do {

cin >> size;

if (size < 1)

cout << "Число элементов должно быть больше 0\n";

} while (size < 1);

Stack my\_stack;

my\_stack.make\_stack(size);

cout << "Содержимое стека:\n";

my\_stack.print\_stack();

my\_stack.delete\_even\_elements();

cout << "Содержимое стека после удаления четных элементов:\n";

my\_stack.print\_stack();

cout << "Содержимое стека после его опустошения:\n";

my\_stack.clear();

my\_stack.print\_stack();

system("pause");

return 0;

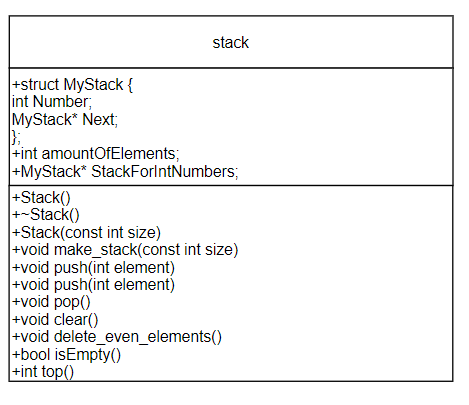
}

* 1. В ходе работы были использованы следующие типы данных:

2.1 (struct) структура стек, в которой есть поле – число (int), а также указатель на следующий элемент (MyStack\*)

2.2 (int) amountOfElements – количество элементов в стеке

2.3 (MyStack\*) StackForIntNumbers – вспомогательный объект класса.

UML – диаграмма

Код программы

Main.cpp

int main()

{

setlocale(LC\_ALL, "");

int size;

cout << "Введите количество элементов: ";

do {

cin >> size;

if (size < 1)

cout << "Число элементов должно быть больше 0\n";

} while (size < 1);

Stack my\_stack;

my\_stack.make\_stack(size);

cout << "Содержимое стека:\n";

my\_stack.print\_stack();

my\_stack.delete\_even\_elements();

cout << "Содержимое стека после удаления четных элементов:\n";

my\_stack.print\_stack();

cout << "Содержимое стека после его опустошения:\n";

my\_stack.clear();

my\_stack.print\_stack();

system("pause");

return 0;

}

Stack.h

class Stack {

private:

struct MyStack {

int Number;

MyStack\* Next;

};

int amountOfElements;

MyStack\* StackForIntNumbers;

public:

Stack()

~Stack()

Stack(const int size)

void make\_stack(const int size)

void push(int element)

void push(int element)

void pop()

void clear()

void delete\_even\_elements()

bool isEmpty()

int top()

};

Stack.cpp

#include<iostream>

#include ”Stack.h”

using namespace std;

Stack() {

amountOfElements = 0;

}

Stack(const int size) {

amountOfElements = size;

}

void make\_stack(const int size) {

int number;

amountOfElements = size;

StackForIntNumbers = new MyStack;

StackForIntNumbers->Next = NULL;

cout << "Введите элементы:\n";

if (amountOfElements > 0) {

cout << "1) ";

cin >> number;

StackForIntNumbers->Number = number;

}

for (int i = 2; i <= amountOfElements; i++) {

cout << i << ") ";

cin >> number;

push(number);

}

}

void push(int element) {

MyStack\* ptr\_on\_new\_element = new MyStack;

ptr\_on\_new\_element->Next = NULL;

ptr\_on\_new\_element->Number = element;

ptr\_on\_new\_element->Next = StackForIntNumbers;

StackForIntNumbers = ptr\_on\_new\_element;

}

void print\_stack() {

MyStack\* pointer = StackForIntNumbers;

if (StackForIntNumbers == NULL)

cout << "Стек пуст\n";

else

while (pointer != NULL) {

cout << pointer->Number << endl;

pointer = pointer->Next;

}

}

void pop() {

MyStack\* ptr\_on\_deleting\_element = StackForIntNumbers;

StackForIntNumbers = StackForIntNumbers->Next;

delete ptr\_on\_deleting\_element;

}

void clear() {

while (StackForIntNumbers != NULL)

pop();

}

void delete\_even\_elements() {

Stack temp\_stack;

for (int i = 0; i < amountOfElements; i++) {

if ((StackForIntNumbers->Number % 2) == 0)

pop();

else {

temp\_stack.push(StackForIntNumbers->Number);

pop();

}

}

while (!temp\_stack.isEmpty()) {

push(temp\_stack.top());

temp\_stack.pop();

}

}

bool isEmpty() {

return (StackForIntNumbers == NULL);

}

int top() {

return (StackForIntNumbers->Number);

}

~Stack() {

clear();

}

};