ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ­­

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | А. В. Устинов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4 |
| ШАБЛОНЫ КЛАССОВ |
| по дисциплине: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4631 |  |  |  | С.А. Гришин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2018

**1. Цель работы**

Изучить принципы построения консольных приложений, применив на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования.

**2. Описание работы**

В работе необходимо реализовать класс в соответствии с вариантом задания и создать приложение для вызова методов класса в функции main(). Реализуемый класс должен представлять из себя структуру данных и уметь работать с встроенными типами данных, таким как: int, double, float, char. И реализовывать функции вставки, удаления и в зависимости от варианта доступ по ключу или индексу.

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Название класса** | **Описание** |
| Вариант 8 | Stack | Адаптируется контейнеров обеспечить стек (LIFO структуры данных) |

**3. Листинг программы**

#include <iostream>

#include "Stack.h"

using namespace std;

int main() {

Stack<int> stcInt;

Stack<char> stcChar;

// Stack int

cout << "\t Int:" << endl;

stcInt.push(1);

stcInt.push(2);

stcInt.push(3);

// stc.push(4);

cout << "Pop:" << endl;

cout << stcInt.pop() << endl;

cout << stcInt.pop() << endl;

cout << stcInt.pop() << endl;

// cout << stc.pop() << endl;

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

// Stack char

cout << "\t Char:" << endl;

stcChar.push(76);

stcChar.push(123);

stcChar.push(88);

cout << "Pop:" << endl;

cout << stcChar.pop() << endl;

cout << stcChar.pop() << endl;

cout << stcChar.pop() << endl;

system("pause");

return 0;

}

#pragma once

#include <memory>

#include <stdexcept>

template <typename T> class Stack {

public:

Stack() { arr = std::make\_unique<T[]>(size); }

void push(T i) {

if (idx == size) {

throw std::runtime\_error("stack overflow");

}

arr[idx++] = i;

}

T pop() {

if (idx == 0) {

throw std::runtime\_error("stack empty");

}

return arr[--idx];

}

private:

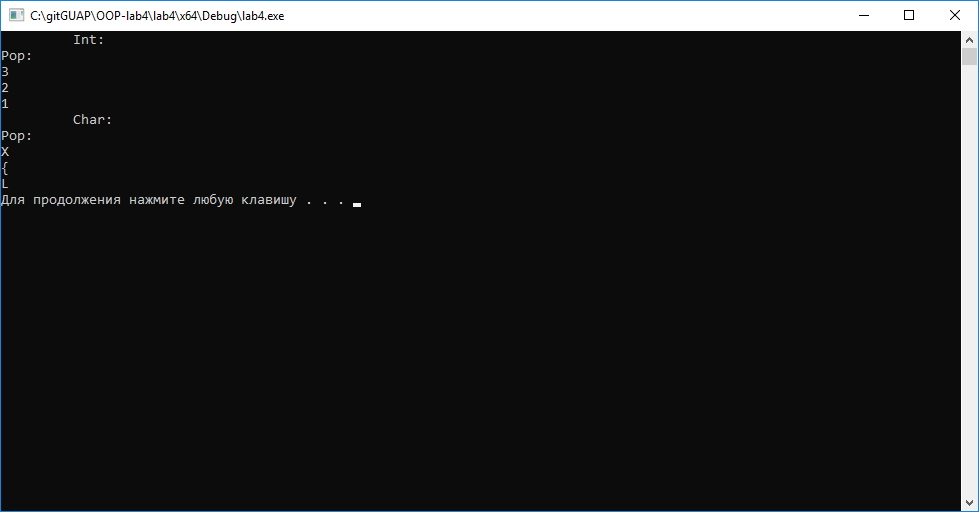
size\_t idx = 0;

size\_t size = 3;

std::unique\_ptr<T[]> arr;

};

**4. Пример работы**



**4. Вывод**

Изучил принципы построения консольных приложений, применив на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования.