МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

Старший преподаватель Е.О.Шумова должность, уч. степень, звание подпись, дата инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

№4

**«Обработка исключительных ситуаций»**

по дисциплине: Объектно-ориентированное программирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. Z7431 20.01.2020 М.Д.Семочкин

подпись, дата инициалы, фамилия

Студ. билет 2014/1054

Санкт-Петербург

2020

**Цель работы**

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

Закрепить знания по теме: Обработка исключительных ситуаций.

**Задание**

Создать класс Массив, содержащий:

* одномерный целочисленный массив
* размерность массива равна 12
* поле, для хранения длины массива
* конструктор, в котором осуществляется ввод данных в массив
* расчетные функции (согласно варианту)
* функцию вывода исходного массива
* задействовать механизм обработки исключительных ситуаций

Вариант 7:

В одномерном массиве, состоящем из n целочисленных элементов, вычислить:

* номер максимального элемента массива;
* произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине — элементы, стоявшие в четных позициях.

**Используемые инструменты разработки**

IDE – Microsoft Visual C++ 2010 Express

**Текст программы**

// Создать класс Массив, содержащий:

// - одномерный целочисленный массив

// - размер массива равен 12

// - поле для хранения длины массива

// - конструктор, в котором осуществляется ввод данных в массив

// - расчетные функции (согласно варианта)

// - функцию вывода исходного массива

// - задействовать механизм обработки исключительных ситуаций

//

// Вариант 7

//

// В одномерном массиве, состоящем из n целочисленных элементов, вычислить:

// - номер максимального элемента массива;

// - произведение элементов массива, расположенных между первым и вторым

// нулевыми элементами.

// Преобразовать массив таким образом, чтобы в первой его половине располагались

// элементы, стоявшие в нечетных позициях, а во второй половине — элементы,

// стоявшие в четных позициях.

//

#include <iostream>

#include <stdlib.h>

#include <math.h>

#include <string>

#include <exception>

using namespace std;

int const ARR\_LENGTH = 12;

class NoZeroError : public exception {

virtual const char\* what() const throw() {

return "В массиве нет нулей";

}

};

class NoNumbersBetweenZeros : public exception {

virtual const char\* what() const throw() {

return "Между нулями нет цифр";

}

};

class MyArray {

private:

int content[12];

int length;

public:

// Конструктор по умолчанию

MyArray();

// Конструктор

MyArray(int arr[]);

// Выводит массив

void print();

// Вычисляет номер максимального элемента массива

int getMaxElementIndex();

// Вычисляет произведение элементов массива, расположенных

// между первым и вторым нулевыми элементами

int calculateProductOfElementsBetweenZeros();

// Преобразует массив таким образом, чтобы в первой его половине

// располагались элементы, стоявшие в нечетных позициях, а во

// второй половине — элементы, стоявшие в четных позициях

void rearrangeArray();

};

MyArray::MyArray() {

// Если конструктор вызван без аргументов, заполнить массив нулями

length = ARR\_LENGTH;

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

content[i] = 0;

}

}

MyArray::MyArray(int arr[]) {

length = ARR\_LENGTH;

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

content[i] = arr[i];

}

}

void MyArray::print() {

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

cout << content[i] << " ";

}

cout << endl << endl;

}

int MyArray::getMaxElementIndex() {

int maxValue = content[0];

int maxValueIndex = 0;

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

if (content[i] > maxValue) {

maxValue = content[i];

maxValueIndex = i;

}

}

return maxValueIndex;

}

int MyArray::calculateProductOfElementsBetweenZeros() {

int firstZeroIndex = -1;

int secondZeroIndex = -1;

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

if (content[i] == 0) {

if (firstZeroIndex == -1) {

firstZeroIndex = i;

} else {

secondZeroIndex = i;

break;

}

}

}

if ( (firstZeroIndex == -1) || (secondZeroIndex == -1) ) {

throw NoZeroError();

}

if (secondZeroIndex - firstZeroIndex == 1) {

throw NoNumbersBetweenZeros();

}

int product = content[firstZeroIndex + 1];

for (int i = firstZeroIndex + 2; i < secondZeroIndex; i++) {

product \*= content[i];

}

return product;

}

void MyArray::rearrangeArray() {

int tmp;

for (int i = 0; i < (length / 2); i += 2) {

tmp = content[i + 1 + (length / 2)];

content[i + 1 + (length / 2)] = content[i];

content[i] = tmp;

}

}

int main(int argc, char \*\*argv) {

setlocale(LC\_ALL, "russian");

cout << "Создаем объект MyArray конструктором по умолчанию:" << endl;

MyArray arr0;

arr0.print();

cout << "Создаем объект MyArray, передав значения в конструктор:" << endl;

int array[] = {8, 800, 555, 35, 35, 0, 3, 5, 3, 0, 8, 0};

MyArray arr1(array);

arr1.print();

cout << "Вычисляем номер максимального элемента массива:" << endl;

cout << arr1.getMaxElementIndex() << endl << endl;

cout << "Вычисляем произведение элементов массива, расположенных между"

<< " первым и вторым нулевыми элементами:" << endl;

cout << arr1.calculateProductOfElementsBetweenZeros() << endl << endl;

cout << "Создаем массив, где нет нулевых элементов, пытаемся сделать то же"

<< " самое, и ловим ошибку:" << endl;

int array2[] = {8, 800, 555, 35, 35, 5, 3, 5, 3, 6, 8, 7};

MyArray arr2(array2);

arr2.print();

try {

cout << arr2.calculateProductOfElementsBetweenZeros() << endl << endl;

} catch (exception& err) {

cout << "Во время выполнения функции произошла ошибка:" << endl;

cout << err.what() << endl;

}

cout << "Создаем массив, где два нулевых элемента идут подряд:" << endl;

int array23[] = {8, 800, 555, 35, 35, 0, 0, 5, 3, 6, 8, 7};

MyArray arr23(array23);

arr23.print();

try {

cout << arr23.calculateProductOfElementsBetweenZeros() << endl << endl;

} catch (exception& err) {

cout << "Во время выполнения функции произошла ошибка:" << endl;

cout << err.what() << endl;

}

cout << "Преобразуем массив:" << endl;

cout << "Было:" << endl;

int array3[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11};

MyArray arr3(array3);

arr3.print();

cout << "Стало:" << endl;

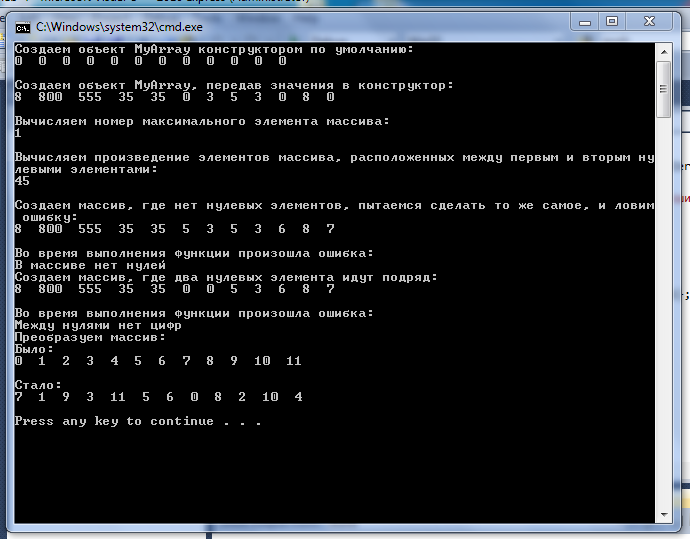
arr3.rearrangeArray();

arr3.print();

return 0;

}

**Результат работы программы**

****

**Вывод**

Во время выполнения лабораторной работы были изучены принципы построения консольных приложений, применены на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования, закреплены знания по теме «Обработка исключительных ситуаций».