

КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

Обработка исключительных ситуаций

по курсу: Объектно-ориентированное программирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2022

Условие

6 вариант

Вариант 6

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

- минимальный элемент массива;
- сумму элементов массива, расположенных между первым и последним положительными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом — все остальные.

Цель работы

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

Создать класс Массив, содержащий:

- одномерный целочисленный массив
- размерность массива определяется суммой двух последних цифр шифра студента (если

сумма получается меньше 12, то размерность массива будет равна 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 (не менее 2-3 ситуаций)

- одну из ситуаций обработать с использованием собственного класса исключений

(класс исключений не должен быть пустым!!!)

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

- минимальный элемент массива;
- сумму элементов массива, расположенных между первым и последним положительными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные

нулю, а потом — все остальные.

Листинг программы

```
main.cpp
#include <iostream>
using namespace std;
```

```

#include <cmath>

#include "exception.h"
#include "array.h"

int main() {
    // смена кодировки
    system("chcp 65001"); // для VS заменить на setlocale(LC_ALL,
    "Russian");

    // размер массива (2 + 6)
    Array array(8); // вместо 8 можно указать любой размер для
массива
    array.show();

    try {
        cout << "Минимальный элемент массива: " << array.get_arr()[ar-
ray.get_min()] << endl;
    } catch (MyException &ex) {
        // cout << "Мы поймали " << ex.what() << endl;
    }

    cout << "Сумма между первым положительным и последним: " << ar-
ray.get_sum() << endl;

    array.my_sort();
    array.show();

    return 0;
}

```

array.h

```

#include <iostream>
using namespace std;

#include <iomanip>

#define rand_min -10
#define rand_max 10

/*-----*/
-----*/

// генерирует случайное число в диапазоне от А до В
int random_int(int a, int b) {
    return a + (rand() % ( b - a + 1 ) );
}

// Ввод целого числа с проверкой
int read_int(){
    int x;
    while ( (scanf("%d",&x) ) != 1 ) {
        printf("Неверное введенное значение, попробуйте еще: ");
        while(getchar() != '\n');
    }
    return x;
}

```

```
/*-----*/
-----*/
```

```
class Array {
public:
    Array(int);
    ~Array();

    void generate(bool);
    void show(const char* separator);
    int* get_arr();
    int get_min();
    int get_sum();
    void my_sort();
private:
    int size;
    int *arr;
};

// конструктор
Array::Array(int Size) {
    size = Size;
    arr = (int*) malloc(size * sizeof(int));
    generate(false);
}

// деструктор
Array::~~Array() {
    free(arr);
}

// заполнение массива
void Array::generate(bool random = false) {
    if (random) {
        for (int i = 0; i < size; i++) {
            arr[i] = random_int(rand_min, rand_max);
        }
    } else {
        for (int i = 0; i < size; i++) {
            cout << "Array[" << i << "] = ";
            arr[i] = read_int();

            try {
                exception_func(arr[i]);
            }
            catch (MyException &ex) {

                cout << "число записалось в массив! " << ex.what() << endl;
                cout << "число, из-за которого произошла ошибка = " <<
ex.GetData() << endl;
            }
        }
    }
}

// вывод массива на экран
void Array::show(const char* separator = " ") {
```

```

    for (int i = 0; i < size; i++) {
        cout << setw(4) << arr[i] << separator;
    }
    cout << endl;
}

// возврат массива
int* Array::get_arr() {
    return arr;
}

// нахождение минимального элемента массива
int Array::get_min() {
    int min_id = 0;

    for (int i = 1; i < size; i++) {
        if (arr[i] < arr[min_id])
            min_id = i;
    }

    return min_id;
}

// сумма элементов массива, расположенных между первым и последним
положительными элементами
int Array::get_sum() {
    bool ok = false;
    for (int i = 0; i < size; i++) {
        if (arr[i] > 0)
            ok = true;
    }

    if (!ok)
        throw runtime_error ("В массиве нет положительных
элементов.");

    int sum = 0;

    int perv_id = -1;
    int posl_id = -1;

    int i;
    for (i = 0; i < size; i++)
        if (arr[i] > 0) {
            perv_id = i;
            break;
        }

    for (i = size - 1; i > -1; i--)
        if (arr[i] > 0) {
            posl_id = i;
            break;
        }

    if (perv_id == posl_id)
        throw runtime_error ("В массиве только 1 положительный
элемент.");

    for (i = perv_id + 1; i < posl_id; i++)
        sum += arr[i];
}

```

```

    return sum;
}

// сортировка (сначала 0)
void Array::my_sort() {
    int* new_arr = (int*) malloc(size * sizeof(int));
    int j = 0;

    for (int i = 0; i < size; i++) {
        if (arr[i] == 0)
            new_arr[j++] = arr[i];
    }

    for (int i = 0; i < size; i++) {
        if (arr[i] != 0)
            new_arr[j++] = arr[i];
    }

    arr = new_arr;
}

```

exception.h

```

#include <iostream>
using namespace std;

class MyException : public runtime_error {
    int DataState;
public:
    MyException(char*, int);
    // ~MyException() {}
    int GetData();
};

MyException::MyException(char* msg, int DataState):runtime_er-
ror(msg) {
    this->DataState = DataState;
}

int MyException::GetData() { return DataState; }

void exception_func(int a) {
    if (a==0) {
        throw MyException((char*)"Вызвано исключение из класса!",a);

        exit(EXIT_FAILURE);
        abort();
    }
}

```

Скриншоты

Вывод

Мы изучили исключения и научились работать с ними.