

ГУАП  
Кафедра №

ОТЧЁТ  
ЗАЩИЩЁН С ОЦЕНКОЙ \_\_\_\_\_

ПРЕПОДАВАТЕЛЬ

\_\_\_\_\_  
должность, уч.степень, звание

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

## ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

Лабораторная работа №6 «Обработка исключительных ситуаций»

по курсу:

ПРЕДМЕТ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр №

\_\_\_\_\_  
M412

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
В.Д. Панков  
инициалы, фамилия

## 1 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.

## 2 Задание

Вариант 5: В одномерном массиве, состоящем из  $n$  целочисленных элементов, вычислить номер максимального элемента массива и 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. Размерность массива определяется как сумма двух последних цифр шифра студента (2024/1624), что равно  $2+4=6$ , но поскольку сумма меньше 12, размерность массива равна 12.

## 3 Код программы

### 3.1 Исключения

```
#pragma once

#include <exception>

class NoZerosException : public std::exception {
public:
    const char* what() const noexcept override {
        return "Массив не содержит двух нулей для вычисления произведения.";
    }
};

class InvalidSizeException : public std::exception {
public:
    const char* what() const noexcept override {
        return "Неверный размер массива.";
    }
};
```

## 3.2 Класс Array

```
#pragma once

#include <vector>
#include "Exceptions.hpp"

class Array {
private:
    static const int SIZE = 12;
    int data[SIZE];
public:
    Array(const std::vector<int>& input);
    int getMaxIndex() const;
    long long getProductBetweenZeros() const;
    void print() const;
    void transform();
};

#include "Array.hpp"
#include <iostream>

Array::Array(const std::vector<int>& input) {
    if (input.size() != SIZE) {
        throw InvalidSizeException();
    }
    for (int i = 0; i < SIZE; ++i) {
        data[i] = input[i];
    }
}

int Array::getMaxIndex() const {
    int maxIdx = 0;
    for (int i = 1; i < SIZE; ++i) {
        if (data[i] > data[maxIdx]) {
            maxIdx = i;
        }
    }
    return maxIdx;
}

long long Array::getProductBetweenZeros() const {
    int firstZero = -1;
    int secondZero = -1;
```

```

    for (int i = 0; i < SIZE; ++i) {
        if (data[i] == 0) {
            if (firstZero == -1) {
                firstZero = i;
            } else {
                secondZero = i;
                break;
            }
        }
    }
    if (firstZero == -1 || secondZero == -1) {
        throw NoZerosException();
    }
    long long product = 1;
    for (int i = firstZero + 1; i < secondZero; ++i) {
        product *= data[i];
    }
    return product;
}

```

```

void Array::print() const {
    for (int i = 0; i < SIZE; ++i) {
        std::cout << data[i] << " ";
    }
    std::cout << std::endl;
}

```

```

void Array::transform() {
    int odd[6];
    int even[6];
    int oddIdx = 0, evenIdx = 0;
    for (int i = 0; i < SIZE; ++i) {
        if (i % 2 == 1) {
            odd[oddIdx++] = data[i];
        } else {
            even[evenIdx++] = data[i];
        }
    }
    for (int i = 0; i < 6; ++i) {
        data[i] = odd[i];
        data[i + 6] = even[i];
    }
}

```

```
}  
}
```

### 3.3 Основная программа

```
#include "Array.hpp"  
#include <vector>  
#include <iostream>  
  
int main() {  
    // Нормальный случай  
    std::vector<int> data = {1, 2, 0, 4, 5, 0, 7, 8, 9, 10, 11, 12};  
    try {  
        Array arr(data);  
        std::cout << "Оригинальный массив: ";  
        arr.print();  
        std::cout << "Индекс максимального элемента: " << arr.getMaxIndex()  
        << std::endl;  
        std::cout << "Произведение между нулями: " <<  
        arr.getProductBetweenZeros() << std::endl;  
        arr.transform();  
        std::cout << "После преобразования: ";  
        arr.print();  
    } catch (const std::exception& e) {  
        std::cout << "Ошибка: " << e.what() << std::endl;  
    }  
  
    // Неверный размер  
    std::vector<int> badData = {1, 2, 3};  
    try {  
        Array arr2(badData);  
    } catch (const std::exception& e) {  
        std::cout << "Ошибка размера: " << e.what() << std::endl;  
    }  
  
    // Нет нулей  
    std::vector<int> noZeroData(12, 1);  
    try {  
        Array arr3(noZeroData);  
        arr3.getProductBetweenZeros();  
    } catch (const std::exception& e) {  
        std::cout << "Ошибка нулей: " << e.what() << std::endl;  
    }  
}
```

```
    return 0;  
}
```

#### 4 Скриншоты выполнения программы

Оригинальный массив: 1 2 0 4 5 0 7 8 9 10 11 12

Индекс максимального элемента: 11

Произведение между нулями: 20

После преобразования: 2 4 0 8 10 12 1 0 5 7 9 11

Ошибка размера: Неверный размер массива.

Ошибка нулей: Массив не содержит двух нулей для вычисления произведения.

Рисунок 1 — Вывод программы, демонстрирующий нормальную работу программы, вычисление индекса максимального элемента, произведения между нулями, преобразование массива, а также обработку исключений для неверного размера и отсутствия нулей

#### 5 Вывод

В результате выполнения лабораторной работы был реализован класс Array для работы с одномерным целочисленным массивом, выполняющий задания варианта 5: вычисление индекса максимального элемента, произведения между нулями и преобразование массива. Программа успешно компилируется и выполняется, демонстрируя корректную работу всех функций и обработку исключительных ситуаций.