

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

старший преподаватель
должность, уч. степень, звание

подпись, дата

Е.О. Шумова
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

Обработка исключительных ситуаций

по дисциплине: Объектно-ориентированное программирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

Z1431
номер группы

подпись, дата

М.Д. Быстров
инициалы, фамилия

Студенческий билет №

2021/3572

Санкт-Петербург 2023

Условие

Цель работы: изучить принципы построения консольных приложений, применив на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования. Закрепить знания по теме: классы, наследование классов, варианты доступа.

Общая часть задания:

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 варианту)
- функцию вывода исходного массива
- задействовать механизм обработки исключительных ситуаций (не менее 2-3 ситуаций)
- одну из ситуаций обработать с использованием собственного класса исключений

Индивидуальное задание (вариант 2):

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

- сумму положительных элементов массива;
- произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.
- упорядочить элементы массива по убыванию.

Полный текст (листинг) программы

1. Файл «main.cpp»

```
1. #include <iostream>
2. #include <string>
3. #include "MyArray.h"
4. #include "ArraySizeException.h"
5.
6. #define ARRAY_SIZE 12
7.
8. int main()
9. {
10.     cout << "LR#5 VAR 2" << endl << endl;
11.
12.     try
13.     {
14.         cout << "MyArray defined size: " << ARRAY_SIZE << ". Size of
creating array: " << 10 << endl;
15.
16.         MyArray* arr = new MyArray(new double[10] {1, 2, 3, 4, 5, 6, 7, 8,
9, 0}, 10);
17.
18.         delete arr;
19.     }
20.     catch (ArraySizeException e)
21.     {
22.         cout << "Caught error: " << e.what() << endl;
23.     }
24.
25.     cout << endl;
26.
27.     try
28.     {
29.         const int length = 12;
30.
31.         double* sourceArray = new double[12] {1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1};
32.
33.         cout << "Source array: ";
34.
35.         for (int i = 0; i < length; i++)
36.         {
37.             if (i > 0)
38.             {
39.                 cout << ", ";
40.             }
```

```

41.
42.         cout << sourceArray[i];
43.     }
44.
45.     cout << endl;
46.
47.     MyArray* arr = new MyArray(sourceArray, length);
48.
49.     arr->getBetweenMaxMinProduct();
50.
51.     delete arr;
52. }
53. catch (std::string e)
54. {
55.     cout << "Caught error: " << e << endl;
56. }
57.
58. cout << endl;
59.
60. try
61. {
62.     MyArray* arr = new MyArray(new double[12] {1, 2, 3, 4, 12, 5, 0.5,
63.     6, 11, -23.5, 9, 10}, 12);
64.
65.     cout << "Array: " << arr->getArrayStr() << endl;
66.     cout << "Positive elements sum: " << arr->getPositiveElementsSum()
67.     << endl;
68.     cout << "Product of elements situated between absolute max and min:
69.     " << arr->getBetweenMaxMinProduct() << endl;
70.
71.     delete arr;
72. }
73. catch (...)
74. {
75.     cout << "Caught error" << endl;
76. }

```

2. Файл «MyArray.h»

```

1. #include <string>
2.
3. #pragma once
4. class MyArray
5. {

```

```

6. private:
7.
8.     double* arr;
9.     int arraySize;
10.
11. public:
12.     MyArray(double* const source, int sourceSize);
13.     ~MyArray();
14.
15.     std::string getArrayStr() const;
16.
17.     double getPositiveElementsSum() const;
18.     double getBetweenMaxMinProduct() const;
19. };
20.
21.

```

3. Файл «MyArray.cpp»

```

1. #include <sstream>
2. #include <limits>
3. #include "MyArray.h"
4. #include "ArraySizeException.h"
5.
6. #define ARRAY_SIZE 12
7.
8. MyArray::MyArray(double* const source, int sourceSize)
9. {
10.     this->arraySize = ARRAY_SIZE;
11.     this->arr = new double[this->arraySize];
12.
13.     if (sourceSize != this->arraySize)
14.     {
15.         throw ArraySizeException();
16.     }
17.
18.     for (int i = 0; i < this->arraySize; i++)
19.     {
20.         this->arr[i] = source[i];
21.     }
22. }
23.
24. MyArray::~MyArray()
25. {
26.     delete[] this->arr;
27. }

```

```

28.
29. std::string MyArray::getArrayStr() const
30. {
31.     std::stringstream ss;
32.
33.     for (int i = 0; i < this->arraySize; i++)
34.     {
35.         if (i != 0)
36.         {
37.             ss << " ";
38.         }
39.
40.         ss << this->arr[i];
41.     }
42.
43.     ss << std::endl;
44.
45.     return ss.str();
46. }
47.
48. double MyArray::getPositiveElementsSum() const
49. {
50.     double sum = 0;
51.
52.     for (int i = 0; i < this->arraySize; i++)
53.     {
54.         if (this->arr[i] > 0)
55.         {
56.             sum += this->arr[i];
57.         }
58.     }
59.
60.     return sum;
61. }
62.
63. double MyArray::getBetweenMaxMinProduct() const
64. {
65.     int maxIdx = -1;
66.     int minIdx = -1;
67.
68.     double max = std::numeric_limits<double>::min();
69.     double min = std::numeric_limits<double>::max();
70.
71.     for (int i = 0; i < this->arraySize; i++)
72.     {
73.         if (abs(this->arr[i]) > max)
74.         {
75.             max = abs(this->arr[i]);
76.
77.             maxIdx = i;

```

```

78.         }
79.
80.         if (abs(this->arr[i]) < min)
81.         {
82.             min = abs(this->arr[i]);
83.
84.             minIdx = i;
85.         }
86.     }
87.
88.     if (maxIdx == minIdx)
89.     {
90.         throw std::string("Absolute max/min not found");
91.     }
92.
93.     int direction = maxIdx > minIdx ? 1 : -1;
94.
95.     double product = 1;
96.
97.     for (int i = minIdx + direction; i != maxIdx; i += direction)
98.     {
99.         product *= this->arr[i];
100.    }
101.
102.    return product;
103. }
104.

```

4. Файл «ArraySizeException.h»

```

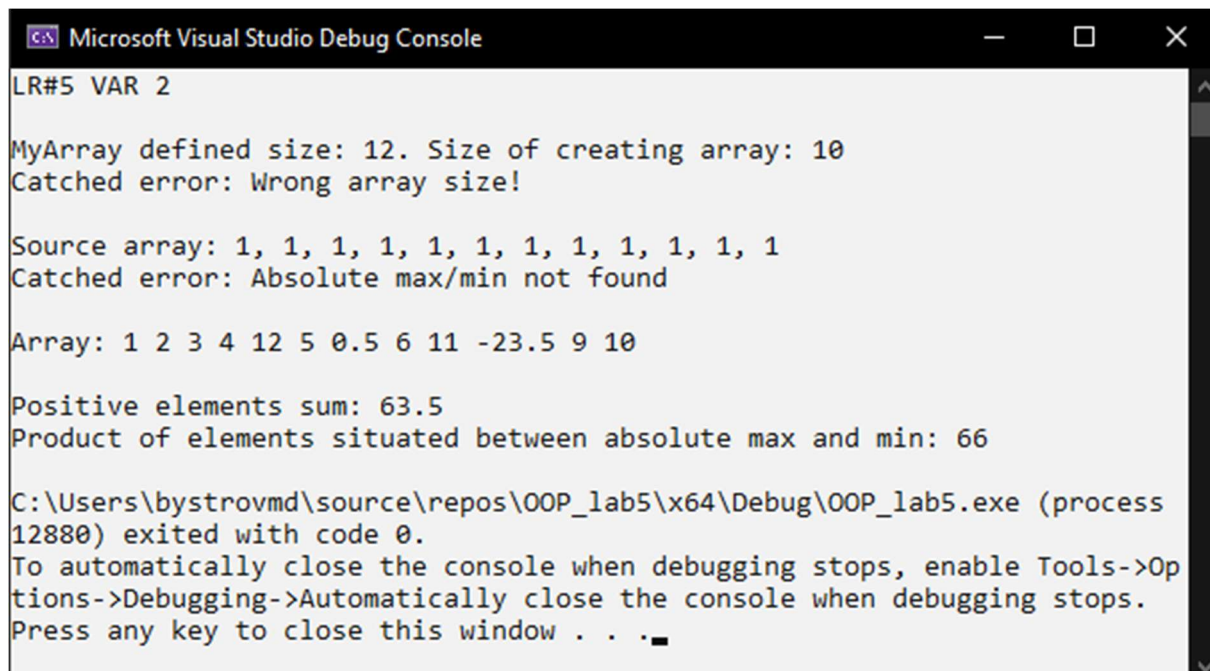
1. #include <iostream>
2. using namespace std;
3.
4. #pragma once
5. class ArraySizeException : public std::exception
6. {
7. public:
8.     const char* what();
9.
10. };

```

5. Файл «ArraySizeException.cpp»

```
1. #include "ArraySizeException.h"
2.
3. const char* ArraySizeException::what()
4. {
5.     return "Wrong array size!";
6. }
7.
```


Работа программы



```
Microsoft Visual Studio Debug Console

LR#5 VAR 2

MyArray defined size: 12. Size of creating array: 10
Caught error: Wrong array size!

Source array: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
Caught error: Absolute max/min not found

Array: 1 2 3 4 12 5 0.5 6 11 -23.5 9 10

Positive elements sum: 63.5
Product of elements situated between absolute max and min: 66

C:\Users\bystrovmd\source\repos\OOP_lab5\x64\Debug\OOP_lab5.exe (process
12880) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Op
tions->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Рисунок 1 Работа программы

Выводы

В ходе выполнения лабораторной работы №5 получены навыки работы с исключениями в языке C++.

Создан класс исключения. При работе программы предусмотрены ситуации, при которых происходит выброс исключения. Продемонстрирован перехват ошибок различных типов, в том числе созданных самостоятельно.

Написанное приложение показывает, каким образом можно организовать обработку ошибок в ходе выполнения программы при помощи перехвата исключений.