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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| «Обработка исключительных ситуаций» |
| по дисциплине: Объектно-ориентированное программирование |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | Z9431 |  |  |  | Андреев Д.И. |
|  | номер группы |  | подпись, дата |  | инициалы, фамилия |
| Студенческий билет № | 2019/3781 | |  |  |  |

Санкт-Петербург 2021

Оглавление

[1. Условие 3](#_Toc70910120)

[2. Листинг программы 3](#_Toc70910121)

[3. Результаты работы программы. 7](#_Toc70910122)

[4. Вывод 7](#_Toc70910123)

1. Условие

Вариант 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить:

• сумму отрицательных элементов массива;

• произведение элементов массива, расположенных между максимальным и минимальным элементами.

Упорядочить элементы массива по возрастанию.

1. Листинг программы

Файл Array.h

pragma once

#include <initializer\_list>

#include <iostream>

class Array

{

friend std::ostream& operator<< (std::ostream& os, const Array& arr);

public:

Array(std::initializer\_list<int> li);

int negativeSum() const;

int multRange(int min, int max) const;

private:

int m\_arr[12];

int m\_arrSize;

};

Файл Array.cpp

#include "Array.h"

#include "Out\_of\_range.h"

Array::Array(std::initializer\_list<int> li)

: m\_arr()

, m\_arrSize(12)

{

try {

if (li.size() > static\_cast<unsigned>(m\_arrSize))

throw Out\_of\_range("cannot initialize array. Out of range. Everything is 0\n");

for (size\_t i = 0; i < li.size(); ++i)

m\_arr[i] = \*(li.begin() + i);

}

catch (Out\_of\_range err) {

std::cerr << err.what();

}

}

int Array::negativeSum() const

{

int sum = 0;

for (int i = 0; i < m\_arrSize; ++i)

if (m\_arr[i] < 0)

sum += m\_arr[i];

return sum;

}

int Array::multRange(int min, int max) const

{

int mult = 0;

try {

if (min < 0)

throw Out\_of\_range("min limit out of range\n");

else if (max > m\_arrSize)

throw Out\_of\_range("max limit out of range\n");

else if (min > max)

throw Out\_of\_range("min limit cannot be less than max\n");

mult = 1;

for (int i = min; i < max; ++i)

mult \*= m\_arr[i];

}

catch (Out\_of\_range err) {

std::cerr << err.what();

}

return mult;

}

std::ostream& operator<< (std::ostream& os, const Array& arr)

{

for (int i = 0; i < arr.m\_arrSize; ++i)

os << arr.m\_arr[i] << ' ';

return os;

}

Файл Out\_of\_range.h

#pragma once

#include <string>

class Out\_of\_range

{

public:

Out\_of\_range(std::string msg);

std::string what() const;

private:

std::string m\_massage;

};

Файл Out\_of\_range.cpp

#include "Out\_of\_range.h"

Out\_of\_range::Out\_of\_range(std::string msg)

: m\_massage(msg)

{

}

std::string Out\_of\_range::what() const

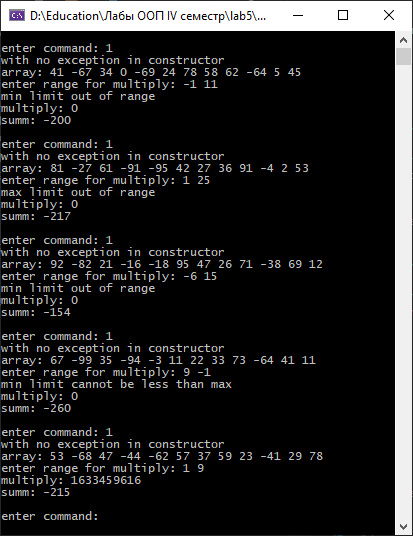
{

return m\_massage;

}

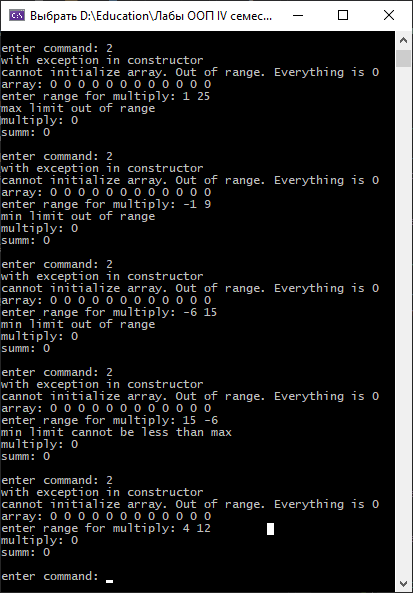
1. Результаты работы программы.

Результат обработки исключений при выходе за пределы при выполнении умножения:



Как можно увидеть, в функции multRange() обрабатывается 3 исключительных ситуации: нижняя граница за пределами массива (меньше 0), верхняя граница за пределами массива, верхняя граница меньше нижней. При появлении исключительной ситуации, производится обработка того исключения, которое было выброшено первым.

Так как в конструктор классу Array передается список инициализации std::initializer\_list<int>, необходимо обработать исключительную ситуацию, когда размер списка инициализации превышает размер массива. Есть 2 варианта обработки: выбросить исключение сразу при превышении размера списка инициализации над размером массива и в этом случае все элементы массива будут заполнены нулями, либо забрать первые значения списка инициализации, количество который будет равно размеру массива и выбросить исключение после этого. При решении задачи был выбран первый вариант:



При этом как можно увидеть все остальные функции выполняются также, как если бы массив был заполнен.

1. Вывод

В ходе выполнения лабораторной работы был разработан класс Array и Out\_of\_range, который является классом ошибки. Класс Array обладает возможностью обрабатывать исключения при неверной инициализации (при передаче списка инициализации в конструктор), а также при выполнении операции перемножения элементов (при выходе за пределы массива переданных в функцию границ).