МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ:

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

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

«Обработка исключительных ситуаций

ПО ДИСЦИПЛИНЕ «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. | 4831 |  |  |  | К.А. Корнющенков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2020

Задание:

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)

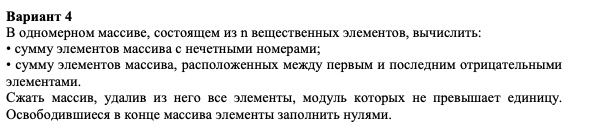
- поле, для хранения длины массива

- 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.



Листинг программы:

#include <iostream>

#include <string>

#include <vector>

using namespace std;

class My\_Error {

private:

string message;

public:

My\_Error();

My\_Error(string message);

string getMesage();

};

My\_Error::My\_Error(){

this->message = "Ошибка";

}

My\_Error::My\_Error(string message){

this->message = message;

}

string My\_Error::getMesage(){

return message;

}

class MyVector{

private:

vector<int> data;

const int dataSize = 12;

public:

MyVector();

void cinData();

void print();

int oddIndex(); //сумму элементов массива с нечетными номерами

int sumFLElement(); //сумму элементов массива, расположенных между первым и последним отрицательными элементами

void compression(); //Сжать массив, удалив из него все элементы, модуль которых не превышает единицу. Освободившиеся в конце массива элементы заполнить нулями

};

MyVector::MyVector(){}

void MyVector::cinData(){

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

int j;

cout << "введите " << i << " элемент массива" << endl;

cin >> j;

data.push\_back(j);

}

}

void MyVector::compression(){

vector<int> removeIndex;

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

if (data[i] >= -1 && data[i] <= 1){

removeIndex.push\_back(i);

}

}

for (int i=0;i<removeIndex.size();i++){

data.erase(data.begin()+removeIndex[i]);

}

for (int i=0;i<removeIndex.size();i++){

data.push\_back(0);

}

}

int MyVector::sumFLElement(){

int firstElIndex = -1;

int lastElIndex = -1;

int sum = 0;

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

if (data[i] < 0 && firstElIndex == -1){

firstElIndex = i;

}else if (data[i] < 0 && firstElIndex > -1){

lastElIndex = i;

}

}

if (lastElIndex == -1) {

return sum;

}else{

for (int i=firstElIndex;i<lastElIndex;i++){

sum += data[i];

}

return sum;

}

}

int MyVector::oddIndex(){

int sum = 0;

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

if (i % 2 == 0){

sum += data[i];

}

}

return sum;

}

void MyVector::print(){

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

cout << data[i] << " ";

}

cout << endl;

}

int main() {

MyVector data;

try {

data.cinData();

} catch (My\_Error error){

error = My\_Error("Ошибка ввода данных");

cout << error.getMesage() << endl;

}

data.print();

try {

cout << "сумма элементов массива с нечетными номерами " << data.oddIndex() << endl;

} catch (My\_Error error) {

error = My\_Error("Ошибка функции нахождения суммы элементов массива с нечетными номерами ");

cout << error.getMesage() << endl;

}

try {

cout << "сумма элементов массива, расположенных между первым и последним отрицательными элементами " << data.sumFLElement() << endl;

} catch (My\_Error error) {

error = My\_Error("Ошибка функции нахождения суммы элементов массива с нечетными номерами");

cout << error.getMesage() << endl;

}

try {

data.compression();

} catch (My\_Error error) {

error = My\_Error("Ошибка функции Сжать массив, удалив из него все элементы, модуль которых не превышает единицу. Освободившиеся в конце массива элементы заполнить нулями");

cout << error.getMesage() << endl;

}

try {

data.print();

} catch (My\_Error error) {

error = My\_Error("Ошибка");

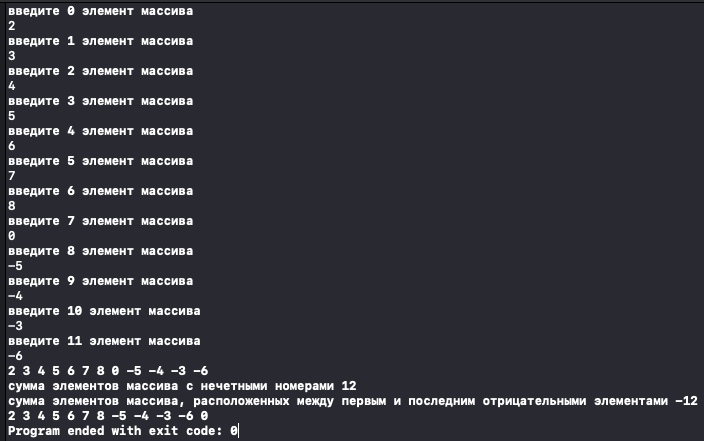
cout << error.getMesage() << endl;

}

return 0;

}

Скриншоты с результатами:



Вывод:

В ходе выполнения лабораторной работы получили практические навыки обработки исключительных ситуация в языке С++ .