МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Белорусский государственный университет транспорта»

Кафедра «Информационно-управляющие системы и технологии»

Отчет  
по лабораторным работам

по дисциплине «Объектно-ориентированное программирование»

Выполнил Проверил

студент группы ГИ-21 ст. преп. каф. «ИУСиТ»

Зайцев И. С. Голдобина Т. А.

Гомель, 2020

Оглавление

[Лабораторная работа № 8 Тема: «Обработка исключений» 3](#_Toc59086742)

[1.1 Задания 3](#_Toc59086743)

[1.2 Контрольные вопросы 7](#_Toc59086744)

Лабораторная работа № 8  
Тема: «Обработка исключений»

### **Цель**

Ознакомиться с понятием исключительных ситуаций и способами их обработки, изучить методы создания классов исключений и их объектов, способов формирования исключений в программе.

## Задания

### Задание 1.1

### Условие

Для каждого варианта необходимо создать три массива a, b и с размерами соответственно n1, n2 и n3 (n1≠n2≠n3). В массив a занести значения функции f(x) согласно варианту (при возникновении исключения заносить нули). Массив b заполнить случайными числами (среди них должны быть и отрицательные числа, и нули). Массив с формируется согласно варианту. Предусмотреть и обработать возникающие при этом исключительные ситуации (деление на ноль, корень из отрицательного числа, арифметическое переполнение, выход за пределы диапазона индексов массива и т.п.). Варианты заданий приведены в таблице.



Программный код

#include <iostream>

#include <cmath>

#include <ctime>

using namespace std;

//класс исключения "деление на ноль"

class ZeroDivideException

{

};

//класс исключения "корень из отрицательного числа"

class SqrtException

{

};

//класс исключения "выход за пределы индексов массива"

class IndexException

{

};

//класс исключения "логарифм от отрицательного чилса"

class LogException

{};

int setN(int i)

{

int n;

cout << "Enter n" << i << ": ";

cin >> n;

return n;

}

int main()

{

int n1, n2, n3, i=0, MAX=10, MIN=-10;

double x=-3;

srand(time(NULL));

do

{

n1 = setN(1);

n2 = setN(2);

n3 = setN(3);

}while(n1 == n2 || n2 == n3 || n1 == n3 || n1<=0 || n2<=0 || n3<=0);

double \*a = new double[n1],

\*b = new double[n2],

\*c = new double[n3];

for(x=-3; x<=7; x+=0.5)

{

try

{

if(i >= n1)

{

throw IndexException();

}

if(x==-1)

{

throw ZeroDivideException();

}

double temp = (x-1)/(x+1);

if(temp<=0)

{

throw LogException();

}

a[i] = log(temp);

cout << "a[" << i << "] = " << a[i] << endl;

i++;

}

catch(IndexException)

{

cout << "Caught IndexException\n";

break;

}

catch(ZeroDivideException)

{

a[i] = 0;

cout << "a[" << i << "] = " << a[i] <<" - Caught ZeroDivideException"<<endl;

i++;

}

catch(LogException)

{

a[i] = 0;

cout << "a[" << i << "] = " << a[i] <<" - Caught LogException"<<endl;

i++;

}

}

for(i; i<n1; i++)

{

a[i] = 0;

cout << "a[" << i << "] = " << a[i] << endl;

}

cout << endl;

for(i=0; i<n2; i++)

{

b[i] = (float)rand()/RAND\_MAX \* (MAX-MIN)+(MIN);

if(b[i]>-1 && b[i]<1)

{

b[i] = 0;

}

cout << "b[" << i <<"] = " << b[i] << endl;

}

cout << endl;

for(i=0; i<n3; i++)

{

try

{

if(i>=n1)

{

throw IndexException();

}

if(i>=n2)

{

throw IndexException();

}

double temp = a[i]+b[i];

if(temp<0)

{

throw SqrtException();

}

c[i] = sqrt(temp);

cout << "c[" << i <<"] = " << c[i] <<endl;

}

catch(SqrtException)

{

c[i] = 0;

cout << "c[" << i <<"] = " << c[i] <<" - Caught SqrtException"<<endl;

}

catch(IndexException)

{

cout << "Caught IndexException" << endl;

break;

}

}

for(i; i<n3; i++)

{

c[i] = 0;

cout << "c[" << i <<"] = " << c[i] << endl;

}

delete []a;

delete []b;

delete []c;

return 0;

}

### Результат

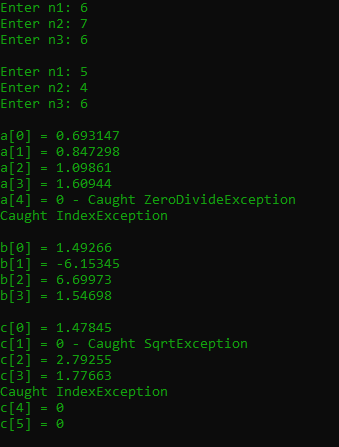


Рисунок 1 – Результат задания 1

## Контрольные вопросы

1. Что такое исключительная ситуация? Опишите ситуации, когда возникают исключения.

Исключительная ситуация – это возникновение непредвиденного или аварийного события, которое может порождаться некорректным использованием аппаратуры.

При выполнении операторов программы может возникнуть ошибочная ситуация (деление на ноль, корень из отрицательного числа, выход за пределы массива). Для обработки таких ситуаций можно предпринять одно из следующих действий: прервать выполнение программы; возвратить код ошибки; вывести сообщение об ошибке и вернуть программе некоторое значение, которое позволит ей продолжить работу. Для решения таких проблем и используются средства генерации и обработки исключений.

1. Как выделить в программе контролируемый блок?

Для выделения контролируемого блока в C++ используется служебная конструкция try{операторы}.

1. В каком месте кода программы может быть сгенерировано исключение?

Исключение может быть сгенерировано в любом месте программы, где возможно возникновение какого-либо аварийного или непредвиденного события, т. е. исключительной ситуации.

1. С помощью какого ключевого слова осуществляется генерация исключительных ситуаций? Укажите существующие формы этого оператора.

Генерация исключения осуществляется с помощью ключевого слова throw. Это слово может употребляться как с параметром, так и без него: throw [выражение].

1. Перечислите основные достоинства и недостатки блока catch(...) {}.

Этот блок предоставляет возможность обрабатывать все исключения, даже если тип переданного параметра неизвестен. Однако это же является и минусом использования этого блока: все исключения обрабатываются одинаково, и более детальная обработка этим блоком не предоставляется.

1. Что случится, если имеются несколько подходящих для этого типа обработчиков?

В этом случае исключение будет обработано в том обработчике, который перехватит его первым (т. е. в том, который расположен выше, ближе к блоку try{}, в котором рассматриваемое исключение было сгенерировано).

1. Каким образом осуществляется обработка исключений во вложенных контролируемых блоках?

Если в каком-либо контролируемом блоке было сгенерировано исключение, то поиск подходящего обработчика будет осуществляться среди тех catch-блоков, которые относятся непосредственно к этому контролируемому блоку. Если же подходящего обработчика не нашлось или исключение было обработано не полностью, то поиск продолжается среди обработчиков внешнего контролируемого блока, если таковой имеется.

1. Каким образом можно ограничить типы исключений, генерируемых функцией?

Это можно сделать с помощью спецификации исключений. В заголовке функции можно задать список типов исключений, которые она может прямо или косвенно порождать. Этот список приводится в конце заголовка и предваряется ключевым словом throw, например, void func(int a) throw (ex\_l, ex\_2). Такое объявление означает, что функция func может сгенерировать только исключения ex\_l, ex\_2 и исключения, являющиеся производными от этих типов.

### Вывод

Ознакомился с понятием исключительных ситуаций и способами их обработки, изучил методы создания классов исключений и их объектов, способов формирования исключений в программе.