Міністерство освіти і науки України

Національний технічний університет України «КПІ ім. Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра Автоматизованих Систем Обробки Інформації та Управління

Звіт

з лабораторної роботи №7 з дисципліни

«Програмування - 2. Структури даних та алгоритми»

**«ДОСЛІДЖЕННЯ МЕХАНІЗМУ ОБРОБКИ ВИНЯТКОВИХ СИТУАЦІЙ»**

Виконав студент

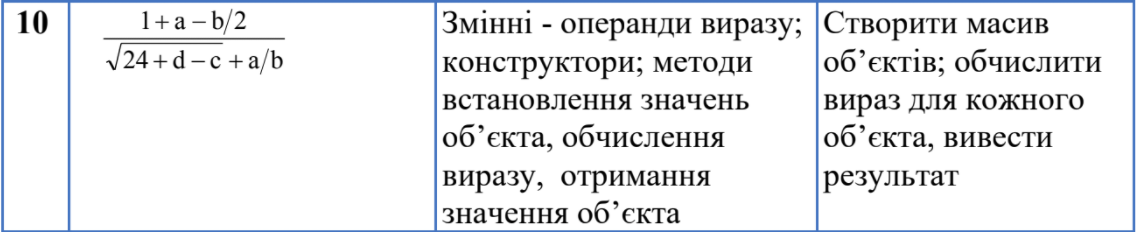
**ІС-01 Кубай Дмитро**

Перевірив викладач

**Проскура С. Л.**

**Київ – 2021**

1. **Завдання:**



1. **Код програми:**

**C#:**

using System;

using FormulaLibrary;

namespace Lab07sharp

{

class Program

{

static void Main(string[] args)

{

Formula[] arr = new Formula[3];

for(int i = 0; i < arr.Length; i++)

{

arr[i] = new Formula(20, 5, 5, 5);

}

arr[0] = new Formula(5, 5, 5, 5);

arr[1] = new Formula(5, 0, 5, 5);

arr[2] = new Formula(5, 5, 100, 5);

for (int i = 0; i < arr.Length; i ++)

{

try

{

Console.WriteLine($"{i}. The result is: {String.Format("{0:0.0000}", arr[i].FindValue())}");

}

catch(Exception ex)

{

throw ex;

}

finally

{

Console.WriteLine("\nThe value was found successfully\n");

}

}

}

}

}

using System;

namespace FormulaLibrary

{

public class Formula

{

private int \_a, \_b, \_c, \_d;

public Formula(int a, int b, int c, int d)

{

\_a = a;

\_b = b;

\_c = c;

\_d = d;

}

public double FindValue()

{

double val;

try

{

if(24 + \_d - \_c < 0)

throw new Exception("You cant get a square root from negative number");

val = 1 + \_a - \_b / 2;

val = val / (Math.Sqrt(24 + \_d - \_c) + \_a / \_b);

}

catch(DivideByZeroException)

{

throw new Exception("You cant divide by null!");

}

catch (Exception ex)

{

throw ex;

}

return val;

}

}

}

**C++:**

#include "head.h"

int main()

{

Formula\* arr = new Formula[3];

for (int i = 0; i < sizeof(arr); i++)

{

arr[i] = Formula(20, 5, 5, 5);

}

arr[0] = Formula(5, 5, 5, 5);

//arr[1] = Formula(5, 0, 5, 5);

arr[2] = Formula(5, 5, 100, 5);

cout.precision(4);

for (int i = 0; i < sizeof(arr); i++)

{

try

{

cout << i << ". The result is:" << fixed << arr[i].FindValue() << endl;

}

catch (string messeage)

{

throw messeage;

}

}

}

#pragma once

#include <iostream>

#include <math.h>

using namespace std;

class Formula

{

public:

Formula()

{

\_a = 0;

\_b = 0;

\_c = 0;

\_d = 0;

}

Formula(int a, int b, int c, int d)

{

\_a = a;

\_b = b;

\_c = c;

\_d = d;

}

double FindValue()

{

double val;

try

{

if (24 + \_d - \_c < 0)

throw ("You cant get a square root from negative number");

val = 1 + \_a - \_b / 2;

val = val / (sqrt(24 + \_d - \_c) + \_a / \_b);

}

catch (string\* messeage)

{

cout << messeage << "\n";

}

return val;

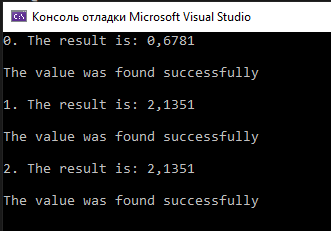
}

private:

int \_a, \_b, \_c, \_d;

};

1. **Виконання програми**

****

1. **Висновок**

У даній роботі – дослідити застосування обробки виняткових ситуацій у мовах програмування С++ та С# і написали програму, яка реалізує:

* клас «Вираз», що представляє собою математичний вираз, у якого змінні a, b, c є операндами виразу. Також задано конструктор, метод встановлення значень об’єкта, та методи обчислення виразу і отримання значення коефіцієнтів a, b, c.

У масив об’єктів класу «Вираз», у який додано декілька екземплярів класу. У разі введення, значень a, b, c, що не задовольняють область допустимих значень виразу, викликається виключення – об’єкт класу Exception, яке після цього обробляється. Механізм обробки виняткової ситуації базуються на використанні блоків try та catch.

Якщо введені значення a, b, c допустимі, то проходячи по створеному масиву обчислюємо значення виразу для кожного об’єкта та виводимо результат.