МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №2**

**з дисципліни «Основи програмної інженерії»**

**на тему: *«*Модульне програмування. Метод висхідного програмування.»**

Виконав: студент гр. ПЗ2011

Горбань М.О.

Прийняла: к.т.н., доцент кафедри КІТ: Горбова О.В.

Дніпро, 2021

програмування. Метод висхідного програмування.

Мета роботи. Отримати практичні навички розробки програм за методом висхідного програмування.

# Завдання

Розробити програму-калькулятор для обчислення трьох математичних виразів за індивідуальним завданням. Розробку програми провести за висхідним методом через створення тестового драйверу для заданої в індивідуальному завданні пари функцій введення даних і обчислення виразу. Для обчислення функцій використати типи даних, вказані у індивідуальному завданні.

Вимоги до програми:

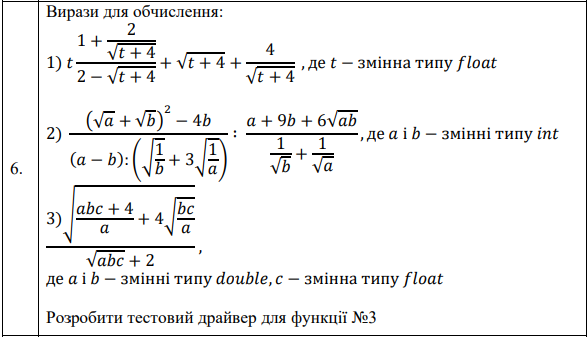
1) вибір виразу для обчислення здійснити за допомогою діалогу з користувачем;

2) введення даних і обчислення виразів реалізувати окремими функціями (всього 6 функцій);

3) для виклику функцій використати масиви вказівників на функції введення даних і обчислення виразів (функція обчислення виразу і відповідна їй функція введення даних повинні мати однаковий індекс в масивах, де вони зберігаються);

4) данні для обчислення виразів ввести з клавіатури, за потреби здійснити їх перевірку;

Варіант 6



**1. Стуктурна схема**

****

**2. Текст програми**

func.h

#pragma once

struct Exam

{

double A, B;

float C;

};

void\* input1();

void\* input2();

void\* input3();

double example1(void\* ptr);

double example2(void\* ptr);

double example3(void\* ptr);

Source.cpp

#include <iostream>

#include <Windows.h>

#include "func.h"

using namespace std;

//Тестовий драйвер для 3-го виразу

void\* inputTest()

{

static Exam m;

m.A = 5;

m.B = 10;

m.C = 15;

return &m;

}

double testDriver(void\* ptr)

{

Exam\* ex = static\_cast<Exam\*>(ptr);

return ex->A + ex->B + ex->C; // = 30

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

//==============тестовий драйвер=====================

void\* par = inputTest();

cout << "A + B + C = " << testDriver(par) << endl;

//===================================================

double(\*ptrExample[])(void\*) { example1, example2, example3 };

void\* (\*ptrInput[])() { input1, input2, input3 };

int index;

cout << "Введите необходимую цифру(1, 2, 3) для подсчёта выражения: "; cin >> index;

while (index > 3)

{

cout << "Выражения с таким номером не существует, введите номер еще раз." << endl;

cin >> index;

}

cout<< "Результат вычисления = "<<ptrExample[index-1](ptrInput[index-1]());

return 0;

}

func.cpp

#include "func.h"

#include<iostream>

#include<cmath>

#define a arr[0]

#define b arr[1]

using namespace std;

/// <summary>

/// Ввод параметров 1-го выражения

/// </summary>

/// <returns>параметр для 1-го выражения</returns>

void\* input1()

{

static float t;

cout << "Цей вираз приймає один десятковий параметр (t). t! = 0 " << endl;

cout << "Введіть значення t = "; cin >> t;

while (t < 0)

{

cout << "'t' не може бути менше 0, введіть параметр ще раз! " << endl;

cout << "Введіть значення t = "; cin >> t;

}

return &t;

}

/// <summary>

/// Ввод параметров 2-го выражения

/// </summary>

/// <returns>параметр для 2-го выражения</returns>

void\* input2()

{

static int arr[2];

cout << "Цей вираз приймає 2 цілочисельних параметра (a, b) " << endl;

cout << "Введіть значення a = "; cin >> a;

cout << "Введіть значення b = "; cin >> b;

while (a <= 0)

{

cout << "Помилка! Введіть число більше 0. Введіть параметр ще раз! " << endl;

cout << "Введіть значення a = "; cin >> a;

}

while (b <= 0)

{

cout << "Помилка! Введіть число більше 0. Введіть параметр ще раз!";

cout << "Введіть значення b = "; cin >> b;

}

return arr;

}

/// <summary>

/// Ввод параметров 3-го выражения

/// </summary>

/// <returns>параметр для 3-го выражения</returns>

void\* input3()

{

static Exam exam;

cout << "Цей вираз приймає три десяткові параметри: " << endl;

cout << "Введите значение a = "; cin >> exam.A;

while (exam.A <= 0)

{

cout << "Помилка! Введіть число більше 0. Введіть параметр ще раз! " << endl;

cout << "Введіть значення a = "; cin >> exam.A;

}

cout << "Введите значение b = "; cin >> exam.B;

while (exam.B <= 0)

{

cout << "Помилка! Введіть число більше 0. Введіть параметр ще раз! " << endl;

cout << "Введіть значення a = "; cin >> exam.B;

}

cout << "Введите значение c = "; cin >> exam.C;

while (exam.C <= 0)

{

cout << "Помилка! Введіть число більше 0. Введіть параметр ще раз! " << endl;

cout << "Введіть значення a = "; cin >> exam.C;

}

return &exam;

}

/// <summary>

/// Вычисление 1-го примера

/// </summary>

/// <param name="ptr"></param>

/// <returns>результат подсчёта</returns>

double example1(void\* ptr)

{

float\* t = static\_cast<float\*>(ptr);

double step1 = sqrt(\*t + 4);

double step2 = 1 + (2 / step1);

double step3 = 2 - step1;

double step4 = \*t \* (step2 / step3) + step1 + (4 / step1);

return step4;

}

/// <summary>

/// Вычисление 2-го примера

/// </summary>

/// <param name="ptr"></param>

/// <returns>результат подсчёта</returns>

double example2(void\* ptr)

{

int\* arr = static\_cast<int\*>(ptr);

double step1 = pow(sqrt(a) + sqrt(b), 2) - (4. \* b);

double step2 = (a - b) / (sqrt(1. / b) + (3. \* sqrt(1. / a)));

double step3 = a + 9. \* b + 6. \* sqrt(a \* b);

double step4 = (1. / sqrt(b)) + (1. / sqrt(a));

double step5 = (step1 / step2) / (step3 / step4);

return step5;

}

/// <summary>

/// Вычисление 3-го примера

/// </summary>

/// <param name="ptr"></param>

/// <returns>результат подсчёта</returns>

double example3(void\* ptr)

{

Exam\* ex = static\_cast<Exam\*>(ptr);

double step1 = (ex->A \* ex->B \* ex->C + 4) / ex->A;

double step2 = 4 \* sqrt((ex->B \* ex->C) / ex->A);

double step3 = sqrt(step1 + step2);

double step4 = sqrt(ex->A \* ex->B \* ex->C) + 2;

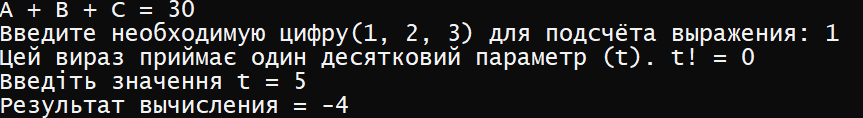
double step5 = step3 / step4;

return step5;

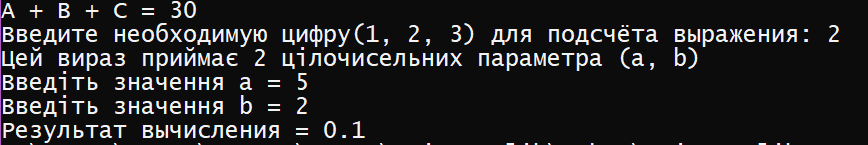
}

**3. Результати виконання**

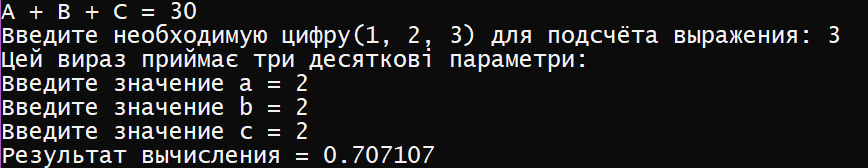
1-й вираз



2-й вираз



3-й вираз



**4. Висновок**

При виконанні лабораторної роботи познайомився з методом висхідного програмування. В цьому методі в першу чергу проектуються модулі найнижчих рівнів, і так поступово йде перехід від модулів нижчого рівня до модулів вищого рівня. Під час тестування створюються тестові драйвери, які імітують модулі вижчих рівнів.