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. ПЗ1911

Сіньков Г.О.

Прийняла: ас. каф. КІТ

Куроп'ятник О. С.

Дніпро, 2019

**Лабораторна робота №2**

Тема. Модульне програмування. Метод висхідного програмування.

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

**Постановка задачі**

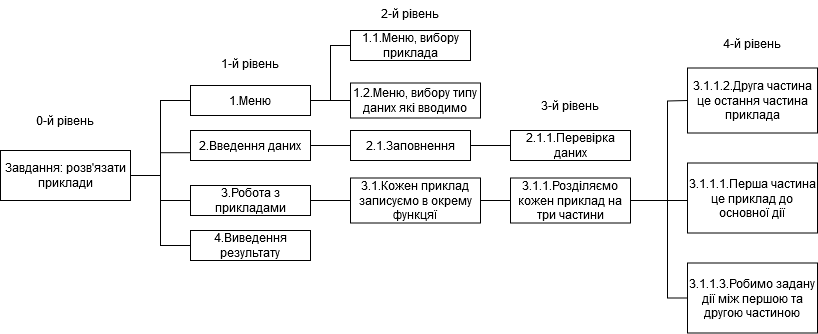
1.Загальна постановка: Розробити програму-калькулятор для обчислення трьох математичних виразів за індивідуальним завданням. Програма повинна мати текстове меню. В програмі використати вказівники на функції.

2.Індивідуальна постановка:

Рівень 3 (відмінно). Параметри для виклику функції записати в масив.

|  |  |
| --- | --- |
| Варіант | Завдання |
| 1. |  |

**Схема модульної структури програми**

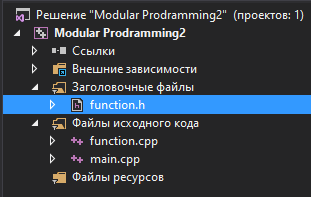


**Розробка програми за методом висхідного програмування**

Для початку розбиваємо функції(f1, f2, f3) для розв’язку заданих прикладів. В них ми розбиваємо приклад на три частини для того щоб зробити менше помилок. Також за допомогою switch розробляємо так зване меню, для того щоб користувач вибирав який тип даних він хоче ввести(ціле або дійсне число).

Наступним кроком буде розробка головної функції(main()), яка є інтерфейсом. Там реалізуємо меню для користувача, де він може вибрати який приклад йому потрібно вибрати і який ті він хоче ввести.

Файлова структура проекту:



**Текст програми**

Файл main.cpp

#include <iostream>

#include <cmath>

#include "Windows.h"

#include "function.h"

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int number1, number2;

do {

system("cls");

std::cout << "1.Перший приклад" << std::endl;

std::cout << "2.Другий приклад" << std::endl;

std::cout << "3.Третій приклад" << std::endl;

std::cout << "4.Вихід" << std::endl;

number1 = input1();

std::cin.get();//очистка буфера

switch (number1) {

case 1: {

do {

system("cls");

std::cout << "1.Цілі числа" << std::endl;

std::cout << "2.Дійсне числа" << std::endl;

std::cout << "3.Вихід в меню" << std::endl;

std::cout << "4.Вихід з програми" << std::endl;

number2 = input2();

std::cin.get();

switch (number2) {

case 1: {

system("cls");

int result;

int arr[3];

std::cout << "Введіть число a: ";

std::cin >> arr[0];

std::cout << "Введіть число b: ";

std::cin >> arr[1];

std::cout << "Введіть число c: ";

std::cin >> arr[2];

result = f1(&arr, INT\_);

std::cout << "Відповідь: " << result << std::endl;

system("pause");

break;

}

case 2: {

system("cls");

double result;

double arr[3];

std::cout << "Введіть число a: ";

std::cin >> arr[0];

std::cout << "Введіть число b: ";

std::cin >> arr[1];

std::cout << "Введіть число c: ";

std::cin >> arr[2];

result = f1(&arr, DOUBLE\_);

std::cout << "Відповідь: " << result << std::endl;

system("pause");

break;

}

case 4:

return NULL;

default:

break;

}

} while (number2 != 3);

break;

}

case 2: {

do {

system("cls");

std::cout << "1.Цілі числа" << std::endl;

std::cout << "2.Дійсне числа" << std::endl;

std::cout << "3.Вихід в меню" << std::endl;

std::cout << "4.Вихід з програми" << std::endl;

number2 = input2();

std::cin.get();

switch (number2) {

case 1: {

system("cls");

int result;

int arr[3];

std::cout << "Введіть число y: ";

std::cin >> arr[0];

std::cout << "Введіть число n: ";

std::cin >> arr[1];

std::cout << "Введіть число m: ";

std::cin >> arr[2];

result = f2(&arr, INT\_);

std::cout << "Відповідь: " << result << std::endl;

system("pause");

break;

}

case 2: {

system("cls");

double result;

double arr[3];

std::cout << "Введіть число y: ";

std::cin >> arr[0];

std::cout << "Введіть число n: ";

std::cin >> arr[1];

std::cout << "Введіть число m: ";

std::cin >> arr[2];

result = f2(&arr, DOUBLE\_);

std::cout << "Відповідь: " << result << std::endl;

system("pause");

break;

}

case 4:

return NULL;

default:

break;

}

} while (number1 != 3);

break;

}

case 3: {

do {

system("cls");

std::cout << "1.Цілі числа" << std::endl;

std::cout << "2.Дійсне числа" << std::endl;

std::cout << "3.Вихід в меню" << std::endl;

std::cout << "4.Вихід з програми" << std::endl;

number2 = input2();

std::cin.get();

switch (number2) {

case 1: {

system("cls");

int result;

int arr[3];

std::cout << "Введіть число z: ";

std::cin >> arr[0];

std::cout << "Введіть число p: ";

std::cin >> arr[1];

std::cout << "Введіть число q: ";

std::cin >> arr[2];

result = f3(&arr, INT\_);

std::cout << "Відповідь: " << result << std::endl;

system("pause");

break;

}

case 2: {

system("cls");

double result;

double arr[3];

std::cout << "Введіть число z: ";

std::cin >> arr[0];

std::cout << "Введіть число p: ";

std::cin >> arr[1];

std::cout << "Введіть число q: ";

std::cin >> arr[2];

result = f3(&arr, DOUBLE\_);

std::cout << "Відповідь: " << result << std::endl;

system("pause");

break;

}

case 4:

return NULL;

default:

break;

}

} while (number1 != 3);

break;

}

case 4:

return NULL;

default:

break;

}

} while (number1 != 4);

}

Файл prototypes.h

#ifndef \_\_FUNCTION\_H\_\_

#define \_\_FUNCTION\_H\_\_

enum Type { INT\_, DOUBLE\_ };

double f1(void \*args, Type type);

double f2(void\* args, Type type);

double f3(void\* args, Type type);

int input1();

int input2();

#endif

Файл matrix.h

#include <iostream>

#include <cmath>

#include "function.h"

int input1() {

int number1;

while (!(std::cin >> number1) || (std::cin.peek() != '\n')) {

std::system("cls");

std::cin.clear();

while (std::cin.get() != '\n');

std::cout << "1.Перший приклад" << std::endl;

std::cout << "2.Другий приклад" << std::endl;

std::cout << "3.Третій приклад" << std::endl;

std::cout << "4.Вихід" << std::endl;

}

return number1;

}

int input2() {

int number2;

while (!(std::cin >> number2) || (std::cin.peek() != '\n')) {

std::system("cls");

std::cin.clear();

while (std::cin.get() != '\n');

std::cout << "1.Цілі числа" << std::endl;

std::cout << "2.Дійсне числа" << std::endl;

std::cout << "3.Вихід в меню" << std::endl;

std::cout << "4.Вихід з програми" << std::endl;

}

return number2;

}

double f1(void \*args,Type type) {

switch (type) {

case INT\_: {

int result, result1, result2;

result1 = pow(pow((static\_cast<int\*>(args)[0] + static\_cast<int\*>(args)[1]), -(1 / 4)) \* pow(static\_cast<int\*>(args)[2], 1 / 2) / (static\_cast<int\*>(args)[0] \* pow(static\_cast<int\*>(args)[1], -(3 / 4))), 4 / 3);

result2 = pow((pow(static\_cast<int\*>(args)[1], 3) \* pow(static\_cast<int\*>(args)[2], 4) / (pow((static\_cast<int\*>(args)[0] + static\_cast<int\*>(args)[1]), 2) \* pow(static\_cast<int\*>(args)[0], 8))), 1 / 6);

result = result1 / result2;

return result;

break;

}

case DOUBLE\_: {

double result, result1, result2;

result1 = pow(pow((static\_cast<double\*>(args)[0] + static\_cast<double\*>(args)[1]), -(1 / 4)) \* pow(static\_cast<double\*>(args)[2], 1 / 2) / (static\_cast<double\*>(args)[0] \* pow(static\_cast<double\*>(args)[1], -(3 / 4))), 4 / 3);

result2 = pow((pow(static\_cast<double\*>(args)[1], 3) \* pow(static\_cast<double\*>(args)[2], 4) / (pow((static\_cast<double\*>(args)[0] + static\_cast<double\*>(args)[1]), 2) \* pow(static\_cast<double\*>(args)[0], 8))), 1 / 6);

result = result1 / result2;

return result;

break;

}

default:

std::cout << "Error" << std::endl;

}

}

double f2(void\* args, Type type) {

switch (type) {

case INT\_: {

int result, result1, result2;

result1 = (2 \* static\_cast<int\*>(args)[1]) / (pow(static\_cast<int\*>(args)[0], static\_cast<int\*>(args)[2] - static\_cast<int\*>(args)[1]));

result2 = (pow(static\_cast<int\*>(args)[2] - static\_cast<int\*>(args)[1], 2) + 4 \* static\_cast<int\*>(args)[2] \* static\_cast<int\*>(args)[1]) / (pow(static\_cast<int\*>(args)[2], 2) - pow(static\_cast<int\*>(args)[1], 2));

result = pow(result1, 1 / static\_cast<int\*>(args)[1]) / pow(pow(static\_cast<int\*>(args)[0], result2), 1 / static\_cast<int\*>(args)[2]);

return result;

break;

}

case DOUBLE\_: {

double result, result1, result2;

result1 = (2 \* static\_cast<double\*>(args)[1]) / (pow(static\_cast<double\*>(args)[0], static\_cast<double\*>(args)[2] - static\_cast<double\*>(args)[1]));

result2 = (pow(static\_cast<double\*>(args)[2] - static\_cast<double\*>(args)[1], 2) + 4 \* static\_cast<double\*>(args)[2] \* static\_cast<double\*>(args)[1]) / (pow(static\_cast<double\*>(args)[2], 2) - pow(static\_cast<double\*>(args)[1], 2));

result = pow(result1, 1 / static\_cast<double\*>(args)[1]) / pow(pow(static\_cast<double\*>(args)[0], result2), 1 / static\_cast<double\*>(args)[2]);

return result;

break;

}

default:

std::cout << "Error" << std::endl;

}

}

double f3(void\* args, Type type) {

switch (type) {

case INT\_: {

int result, result1, result2;

result1 = pow(pow(static\_cast<int\*>(args)[0], 2 / static\_cast<int\*>(args)[1]) + pow(static\_cast<int\*>(args)[0], 2 / static\_cast<int\*>(args)[2]), 2) - 4 \* pow(static\_cast<int\*>(args)[0], ((2 / static\_cast<int\*>(args)[1]) + 2 / static\_cast<int\*>(args)[2]));

result2 = pow(pow(static\_cast<int\*>(args)[0], 1 / static\_cast<int\*>(args)[1]) - pow(static\_cast<int\*>(args)[0], 1 / static\_cast<int\*>(args)[2]), 2) + 4 \* pow(static\_cast<int\*>(args)[0], ((1 / static\_cast<int\*>(args)[1]) + 1 / static\_cast<int\*>(args)[2]));

result = pow(result1/ result2, 1 / 2);

return result;

break;

}

case DOUBLE\_: {

double result, result1, result2;

result1 = pow(pow(static\_cast<double\*>(args)[0], 2 / static\_cast<double\*>(args)[1]) + pow(static\_cast<double\*>(args)[0], 2 / static\_cast<double\*>(args)[2]), 2) - 4 \* pow(static\_cast<double\*>(args)[0], ((2 / static\_cast<double\*>(args)[1]) + 2 / static\_cast<double\*>(args)[2]));

result2 = pow(pow(static\_cast<double\*>(args)[0], 1 / static\_cast<double\*>(args)[1]) - pow(static\_cast<double\*>(args)[0], 1 / static\_cast<double\*>(args)[2]), 2) + 4 \* pow(static\_cast<double\*>(args)[0], ((1 / static\_cast<double\*>(args)[1]) + 1 / static\_cast<double\*>(args)[2]));

result = pow(result1 / result2, 1 / 2);

return result;

break;

}

default:

std::cout << "Error" << std::endl;

}

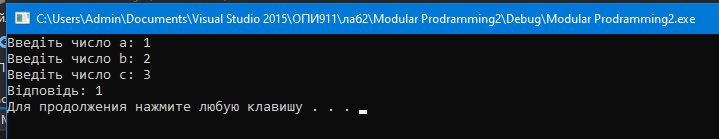
}

**Результати виповнення програми**

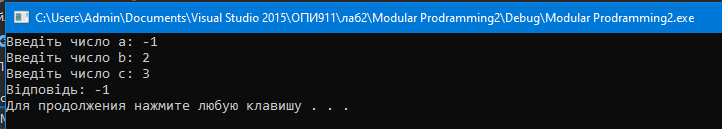
1.**Цілі числа**

а) Перший приклад

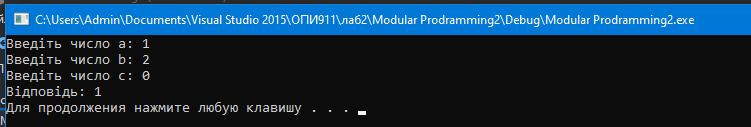
-додатні: a=1; b=2; c=3;



-від’ємні: a=-1; b=2; c=3;

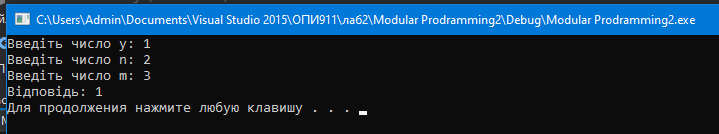


-з нульовою зміною: a=1; b=2; c=0;

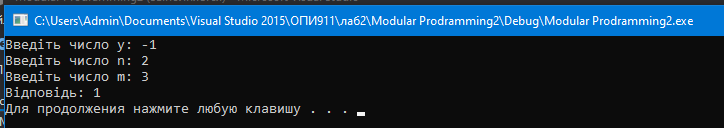


б)Другий приклад

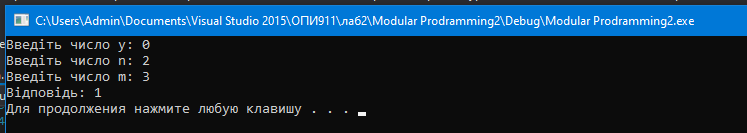
-додатні: y=1; n=2; m=3;



-від’ємні: y=-1; n=2; m=3;

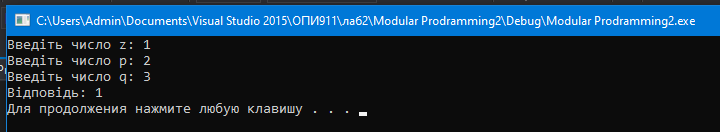


-з нульовою зміною: y=0; n=2; m=3;

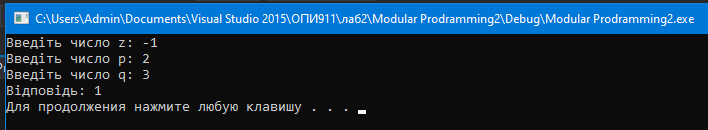


в)Третій приклад

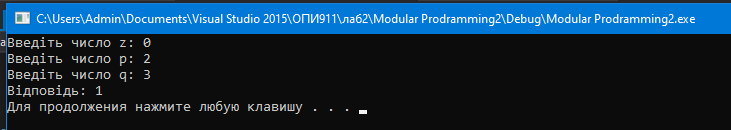
-додатні: z=1; p=2; q=3;



-від’ємні: z=-1; p=2; q=3;



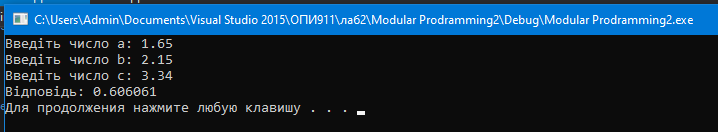
-з нульовою зміною: z=0; p=2; q=3;



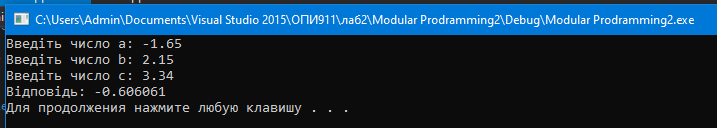
2.**Дійсні числа**

а) Перший приклад

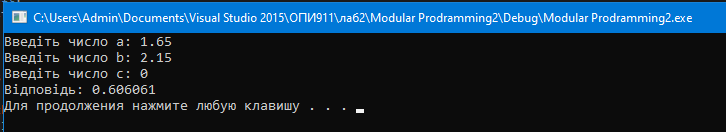
-додатні: a=1,65; b=2,15; c=3,34;



-від’ємні: a=-1,65; b=2,15; c=3,34;

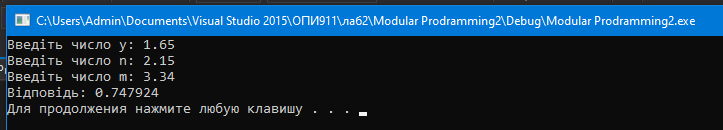


-з нульовою зміною: a=1,65; b=2,15; c=0;

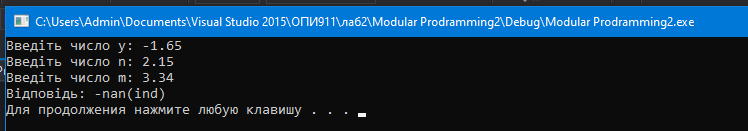


б)Другий приклад

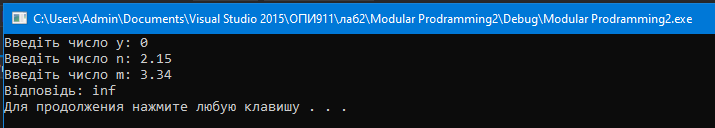
-додатні: y=1,65; n=2,15; m=3,34;



-від’ємні: y=-1,65; n=2,15; m=3,34;

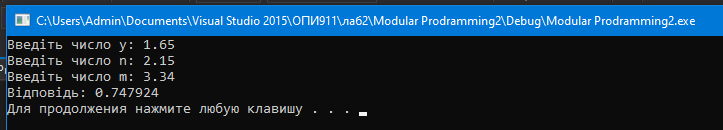


-з нульовою зміною: y=0; n=2.15; m=3,34;

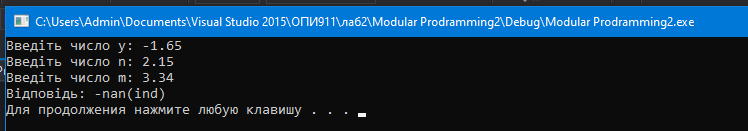


в)Третій приклад

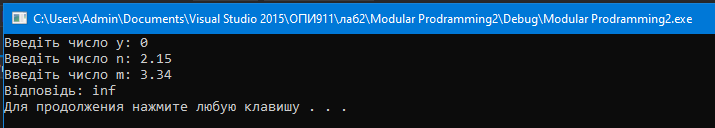
-додатні: z=1,65; p=2,15; q=3,34;



-від’ємні: z=-1,65; p=2,15; q=3,34;



-з нульовою зміною: z=0; p=2,15; q=3,34;



**Висновок**: після перевірки набором тестів програма не відобразила ніяких недоліків, умова індивідуального завдання було виконано. Для розробки програми використовувалися метод висхідного. Використання методів для розробки мені було значно легше так, як я кожний модуль розбивав на багато маленьких частин, де потім збирав в одне ціле. Цей метод займає більше часу бо кожний модуль розробляється досить детально і доводиться до ідеалу.