**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2**

**студента гр. И-1-22**

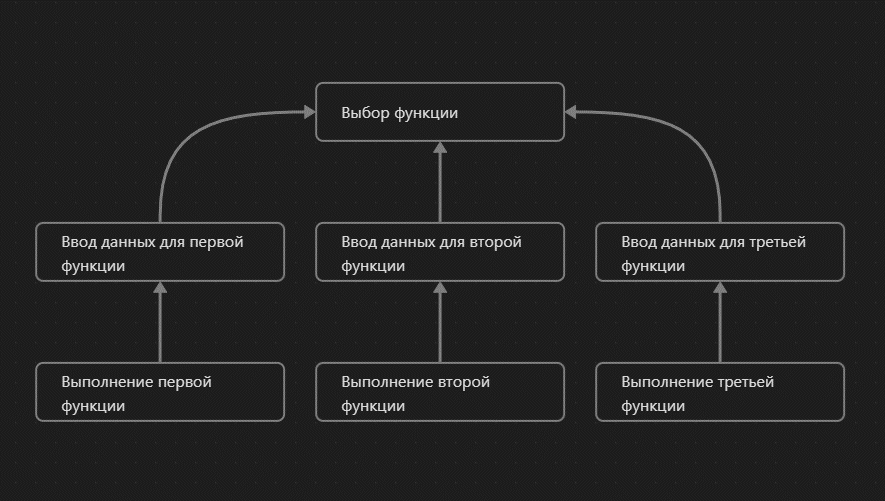
**Пайвин Кирилл**

По дисциплине: Программная инженерия

Тема работы: Модульное программирование. Метод восходящего программирования.

Цель работы: Получить практические навыки оценки разработки программ по методу восходящего программирования.

Структурная схема программы:



Тестовый драйвер осуществляет вызов 3 функции с уже подготовленными вводными данными.

Листинг тестового драйвера:

void testThreeFun**(){**

//Test 1

cout **<<** "m = 2, n = 3, a = 2"**<<** endl**;**

ThreeFunParams params1**;**

params1**.**m **=** 2**;** params1**.**n **=** 3**;** params1**.**a **=** 2**;**

ThreeFun**(&**params1**);**

cout **<<** "Result: " **<<** params1**.**result **<<** endl**;**

//Test 2

cout **<<** "m = 1, n = 1, a = 1"**<<** endl**;**

ThreeFunParams params2**;**

params2**.**m **=** 1**;** params2**.**n **=** 1**;** params2**.**a **=** 1**;**

ThreeFun**(&**params2**);**

cout **<<** "Result: " **<<** params2**.**result **<<** endl**;**

//Test 3

cout **<<** "m = 23, n = 12, a = 12.3"**<<** endl**;**

ThreeFunParams params3**;**

params3**.**m **=** 23**;** params3**.**n **=** 12**;** params3**.**a **=** 12.3**;**

ThreeFun**(&**params3**);**

cout **<<** "Result: " **<<** params3**.**result **<<** endl**;**

**}**

Листинг:

#include <iostream>

#include <cmath>

**using** **namespace** std**;**

// ----------------- 1

struct FirstFunParams**{**

double result **=** 0**;**

double a **=** 0**;**

**};**

void InOutFirstFun**(**double **\***a**){**

cout **<<** "(1/(a^1/2 + (a+1)^1/2) + 1/(a^1/2 - (a-1)^1/2))/(1+((a+1)/(a-1))^1/2)" **<<** endl**;**

cout **<<** "Input a (double):"**;**

cin **>>** **\***a**;**

**}**

void FirstFun**(**void**\*** params**){**

FirstFunParams**\*** p **=** **static\_cast<**FirstFunParams**\*>(**params**);**

double a **=** p**->**a**;**

**if(**a **<=** 1**){**

p**->**result **=** 0**;**

**}**

**else{**

p**->**result **=**

**((**1**/(**pow**(**a**,** 0.5**)** **+** pow**(**a**+**1**,** 0.5**)))+(**1**/(**pow**(**a**,** 0.5**)** **-** pow**(**a**-**1**,** 0.5**))))**

**/**

**(**1**+**pow**(**a**+**1 **/** a**-**1**,** 0.5**));**

**}**

**}**

// ----------------- 1 end

// ----------------- 2

struct SecondFunParams**{**

double result **=** 0**;**

float x **=** 0**;**

float a **=** 0**;**

**};**

void InOutSecondFun**(**float **\***x**,** float **\***a**){**

cout **<<** "((x \* (x^2-a^2)^-1/2 + 1)/(a\* (x-a)^-1/2 + (x-a)^-1/2))/(((a^2 \* (x+a)^1/2)/(x-(x^2-a^2)^1/2))+(1/(x^2-a\*x)))" **<<** endl**;**

cout **<<** "Input x (float):"**;**

cin **>>** **\***x**;**

cout **<<** "Input a (float):"**;**

cin **>>** **\***a**;**

**}**

void SecondFun**(**void**\*** params**){**

SecondFunParams**\*** p **=** **static\_cast<**SecondFunParams**\*>(**params**);**

float x **=** p**->**x**;**

float a **=** p**->**a**;**

**if(**x**+**a **<=** 0 **||** **(**a **==** 0 **&&** x **==** 0**)){**

p**->**result **=** 0**;**

**}**

**else{**

p**->**result **=**

**(**

**(**x**\***pow**((**pow**(**x**,** 2**)** **-** pow**(**a**,** 2**)),** **-**0.5**)** **+** 1**)**

**/**

**(**a**\***pow**(**x**-**a**,-**0.5**)+**pow**(**x**-**a**,** 0.5**))**

**)**

**/**

**(**

**(**

**(**pow**(**a**,** 2**)\***pow**(**x**+**a**,**0.5**))**

**/**

**(**x**-**pow**(**pow**(**x**,** 2**)-**pow**(**a**,** 2**),** 0.5**))**

**)**

**+**

**(**

1

**/**

**(**pow**(**x**,** 2**)** **-** a**\***x**)**

**)**

**);**

**}**

**}**

// ----------------- 2 end

// ----------------- 3

struct ThreeFunParams**{**

double result **=** 0**;**

int m **=** 0**;**

int n **=** 0**;**

double a **=** 0**;**

**};**

void InOutThreeFun**(**int **\***m**,** int **\***n**,** double **\***a**){**

cout **<<** "((a^1/m - a^1/n)^2 \* 4\*a^(m+n/m\*n))/( (a^2/m - a^2/n)\*((a^m+1 + (a^n+1)^1/n)^1/m))" **<<** endl**;**

cout **<<** "Input m (int):"**;**

cin **>>** **\***m**;**

cout **<<** "Input n (int):"**;**

cin **>>** **\***n**;**

cout **<<** "Input a (double):"**;**

cin **>>** **\***a**;**

**}**

void ThreeFun**(**void**\*** params**){**

ThreeFunParams**\*** p **=** **static\_cast<**ThreeFunParams**\*>(**params**);**

double a **=** p**->**a**;**

int n **=** p**->**n**;**

int m **=** p**->**m**;**

**if((**m **==** n**)** **||** **(**m **==** 0 **&&** n **==** 0**)** **||** a **<=** 1**){**

**return;**

**}**

**else{**

p**->**result **=**

**(**

pow**(**pow**(**a**,** **(**double**)**1 **/** **(**double**)**m**)** **-** pow**(**a**,** **(**double**)**1 **/** **(**double**)**n**),** 2**)** **\*** 4 **\*** pow**(**a**,** **(**double**)**m**+**n **/** **(**double**)**m**\***n**)**

**)**

**/**

**(**

**(**

pow**(**a**,** **(**double**)** 2 **/** **(**double**)** m**)** **-** pow**(**a**,** **(**double**)** 2 **/** **(**double**)** n**)**

**)**

**\***

**(**

pow**(**

pow**(**a**,** m**+**1**)**

**+**

pow**(**pow**(**a**,** n**+**1**),** **(**double**)** 1 **/** **(**double**)** n**),**

**(**double**)** 1 **/** **(**double**)** m

**)**

**)**

**);**

**}**

**}**

// ----------------- 3 end

// ----------------- Test 3

void testThreeFun**(){**

//Test 1

cout **<<** "m = 2, n = 3, a = 2"**<<** endl**;**

ThreeFunParams params1**;**

params1**.**m **=** 2**;** params1**.**n **=** 3**;** params1**.**a **=** 2**;**

ThreeFun**(&**params1**);**

cout **<<** "Result: " **<<** params1**.**result **<<** endl**;**

//Test 2

cout **<<** "m = 1, n = 1, a = 1"**<<** endl**;**

ThreeFunParams params2**;**

params2**.**m **=** 1**;** params2**.**n **=** 1**;** params2**.**a **=** 1**;**

ThreeFun**(&**params2**);**

cout **<<** "Result: " **<<** params2**.**result **<<** endl**;**

//Test 3

cout **<<** "m = 23, n = 12, a = 12.3"**<<** endl**;**

ThreeFunParams params3**;**

params3**.**m **=** 23**;** params3**.**n **=** 12**;** params3**.**a **=** 12.3**;**

ThreeFun**(&**params3**);**

cout **<<** "Result: " **<<** params3**.**result **<<** endl**;**

**}**

// ----------------- Test 3 end

int main**(){**

testThreeFun**();**

// void(\*ptrFun[3])(void\*) = {FirstFun, SecondFun, ThreeFun};

// int num;

// cout << "Select 0 or 1 or 2: ";

// cin >> num;

// switch (num)

// {

// case 0:

// {

// FirstFunParams params;

// InOutFirstFun(&params.a);

// ptrFun[num](&params);

// cout << "Result: " << params.result << endl;

// break;

// }

// case 1: {

// SecondFunParams params;

// InOutSecondFun(&params.x, &params.a);

// ptrFun[num](&params);

// cout << "Result: " << params.result << endl;

// break;

// }

// case 2:{

// ThreeFunParams params;

// InOutThreeFun(&params.m, &params.n, &params.a);

// ptrFun[num](&params);

// cout << "Result: " << params.result << endl;

// break;

// }

// default:

// {

// cout << "error";

// break;

// }

// }

system**(**"pause"**);**

**}**