Содержание

[Задание: 2](#_Toc4323753)

[Исходный код программы на языке С++: 2](#_Toc4323754)

[Граф вызовов программы и количественные характеристики программы 4](#_Toc4323755)

Задание:

В соответствии с представленным исходным кодом построить граф вызовов программы, составить список путей вызовов и рассчитать количественные характеристики качества программы.

Исходный код программы на языке С++:

#include "stdafx.h"

#include "conio.h"

#include <iostream>

using namespace std;

Int main()

{

int a=0, b=0, c=0, d=0, e=0, f=0, g=0, h=0, m=0, j=0, k=0, l=0; //М1, инициализация переменных

cout <<"input a"<<endl;

cin >> a; //М1, ввод переменной а

if (a<=100) //М1, проверка условия перехода, переход к М2

goto M2;

if (a>100 && a<1000) //М1, проверка условия перехода, переход к М3

goto M3;

if (a>=1000) //М1, проверка условия перехода, переход к М4

goto M4;

M2: //M2

b=a+10; //М2, выполнение действий с переменной а

if (b<=100) //М2, проверка условия перехода, переход к М5

goto M5;

if (b>100)

goto M6;

M3: //M3

c=a+20; //М3, выполнение действий с переменной а

goto M7; //М3, переход к М7

M4: //M4

d=a+100; //М4, выполнение действий с переменной а

if (d<=1100) //М4, проверка условия перехода, переход к М8

goto M8;

if (d>1100) //М4, проверка условия перехода, переход к М9

goto M9;

M5: //М5

e=b\*2; //М5, выполнение действий с переменной b, переход к М10

goto M10;

M6: //М6

f=b\*3; //М6, выполнение действий с переменной b, переход к М11

goto M11;

M7: //М7

g=c\*4; //М7, выполнение действий с переменной c, переход к М10

goto M10;

M8: //М8

h=d\*5; //М8, выполнение действий с переменной d, вывод конечного значения

cout<<"M8 result- "<<h<<endl;

M9: //М9

m=d\*6; //М9, выполнение действий с переменной d

f=b\*4; //М9, выполнение действий с переменной b

if (m<=1200) //М9, проверка условия перехода, переход к М11

goto M11;

if (m>1200) //М9, проверка условия перехода, переход к М12

goto M12;

M10: //М10

cout<<"M10 result for M7- "<<g<<endl;//М10, вывод значения переменной g из модуля 7

j=e\*10; //М10, выполнение действий с переменной e, вывод конечного значения

cout<<"M10 result- "<<j<<endl;

M11: //М11

k=f\*20; //М11, выполнение действий с переменной f, вывод конечного значения

cout<<"M11 result- "<<k<<endl;

M12: //М12

l=m\*30; //М12, выполнение действий с переменной m, вывод конечного значения

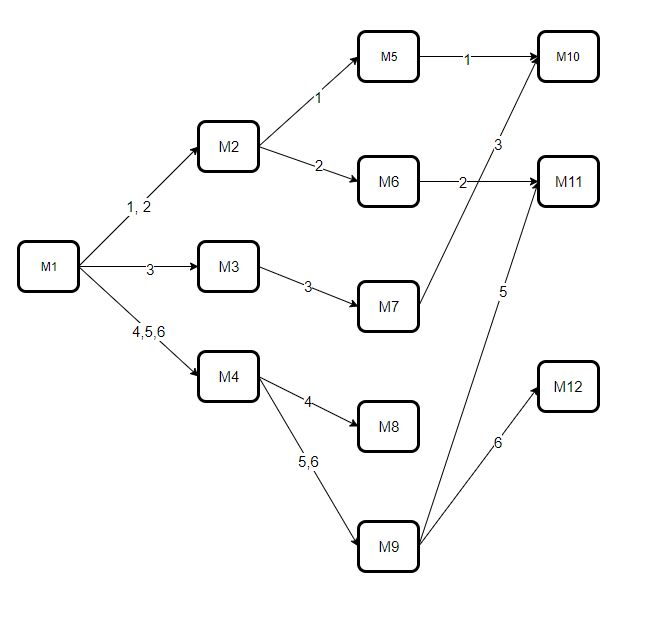
cout<<"M12 result- "<<l<<endl;

getch(); //пауза

}

Граф вызовов программы и количественные характеристики программы

**Граф вызовов программы:**



**Уровней вызова**: L = 4

**Вершин в графе**: N =12

**Расчет иерархической сложности:**

Иерархическая сложность графа рассчитывается по следующей формуле:

**I=N/L,**

где **N** – количество вершин в графе вызовов модулей;

**L** – количество уровней.

I=N/L=12/4=3.

**Расчет структурной сложности:**

Структурная сложность графа рассчитывается по следующей формуле:

**S=D/N,**

где **D** – количество ребер (стрелок) в графе вызовов модулей;

D=13

**N** – количество вершин.

S=D/N=13/12

**C (Мx)** – количество всех модулей, которые вызывает Мx;

С(М1)=3

С(М2)=2

С(М3)=1

С(М4)=3

С(М5)=1

С(М6)=1

С(М7)=1

С(М8)=0

С(М9)=2

С(М10)=0

С(М11)=0

С(М12)=0

**А (Мj)** – доступность модуля Мj, принадлежащего пути Рi,.

**А*(*Мj*)=*∑ А (Мx)/ C (Мx),**

где **А (Мx)** – доступность x-го модуля, вызывающего Мj;

А (М)=1, если модуль М является самым верхним (головным) модулем.

A(М2)=А(М1)/С(М1)=1/3=1/3

А(М3)=1/3

А(М4)=1/3

А(М5)=А(М2)/С(М2)=(1/3)/(1/2)=1/6

А(М6)=1/6

А(М7)=1/3

А(М8)=1/6

А(М9)=1/6

А(М10)=1/2

А(М11)=1/4

А(М12)=1/12

**Pi**– тестируемость *i*-го пути вызовов.

**Pi=[∑1/A( Mj )]-1,**

где **j** – индексы модулей в пути вызовов;

**x** – индексы модулей, которые вызывают Mj;

**Расчет тестируемости**

Оценка тестируемости ПС– (T) может быть проведена с использованием формул:

**Т=[(1/Nв) · (∑1/Pi)]-1, i=1..Nв**

где **NВ** – количество путей вызовов в графе вызовов модулей;