15) **{4 т.}** Напишете какъв резултат извежда следващата програма:

#include <iostream.h>

class Point{

public:

Point(float=0.0, float=0.0);

~Point();

protected:

float x, y;

};

Point::Point(float a, float b)

{ x = a;

y = b;

cout << "Конструктор на Point: "

<< '[' << x << ", " << y << ']' << endl;

}

Point::~Point()

{ cout << "Деструктор на Point: "

<< '[' << x << ", " << y << ']' << endl;

}

class Circle:public Point{

public:

Circle(float r=0.0, float x=0.0, float y=0.0);

~Circle();

private:

float radius;

};

Circle::Circle(float r, float a, float b):Point(a, b)

{ radius = r;

cout << " Конструктор на Circle: " << radius

<< '[' << a << ", " << b << ']' << endl;

}

Circle::~Circle ()

{ cout << "Деструктор на Circle: " << radius

<< '[' << x << ", " << y << ']' << endl;

}

void main()

{

{Point p(1.1, 2.2);}

Circle circle1(4.5, 7.2, 2.9);

Circle circle2(10, 5, 5);

}

**23) {2 т.} Функцията search търси елемент по ключ в свързан списък и го актуализира. Попълнете**

**липсващите оператори на местата на двете многоточия в следния фрагмент от програма:**

**struct student**

**{ long nomer; // факултетен номер**

**float sr\_uspeh; // среден успех**

**student \*next;**

**};**

**// first – указател към началото на свързания списък**

**// key – ключ. факултетният номер се разглежда като ключ**

**// value – стойност, която ще актуализира средния успех**

**void search(student \*first, long key, float value)**

**{ student \*ptr = first;**

**while (ptr)**

**if (ptr->nomer == key)**

**.........................................................................................................**

**else**

**.........................................................................................................**

**if(ptr)**

**ptr->sr\_uspeh = value;**

**else**

**cout << "\nНяма елемент с указания ключ";**

**}**

**27. Функцията push добавя елемент, а функцията pop изключва елемент от последователен стек.**

**Многоточето да се замени с липсващия оператор в следните фрагменти:**

**#define M 100**

**struct stack**

**{ int t; // номер на елемента във върха на стека**

**float stack\_array[M];**

**};**

**a) void push(const stack \*s, float x)**

**{ . . .++s->t;**

**if(s->t > M)**

**{ cout << "\nПРЕПЪЛВАНЕ\n";**

**exit(1); }**

**s->stack\_array[s->t - 1] = x;**

**}**

**b) float pop(const stack \*s)**

**{ . . .if (s->t=NULL)**

**s->t--;**

**return s->stack\_array[s->t];**

**}**

**28. Функцията push добавя елемент в свързан стек, а главната функция я използва.**

**Многоточето да се замени с липсващия оператор в следните фрагменти:**

**struct stack\_el**

**{ int info;**

**stack\_el \*link;**

**};**

**a) void push(stack\_el \*\*t, int x)**

**{ stack\_el \*p;**

**if ((p=new stack\_el) == NULL)**

**{ cout << "\nНяма свободна памет\n";**

**exit(1);**

**}**

**p->info = x;**

**. . .p->link=\*t**

**\*t = p;**

**}**

**b) void main()**

**{ . . .stack\_el\*stack;**

**push(&stack, 1);**

**push(&stack, 2);**

**}**

**26) {2 т.} Функцията add добавя нов елемент в свързан списък след k-тия елемент на списъка.**

**Попълнете липсващия израз и липсващия оператор на местата на двете многоточия в следния**

**фрагмент от програма:**

**фак. № стр.12/17**

**struct student**

**{ long nomer; // факултетен номер**

**float sr\_uspeh; // среден успех**

**student \*next;**

**};**

**// first – указател към началото на свързания списък**

**void add(student \*first, int k)**

**{ student \*ptr = first, \*ptr1;**

**while (.....................................ptr&&i<k)**

**{ ptr=ptr->next; i++;}**

**if (ptr == NULL)**

**{ cout << "\nНяма " << k << "елемента в списъка\n";**

**return; }**

**if ((ptr1 = new student) == NULL)**

**{ cout << "\nНяма свободна памет.\n";**

**exit(1); }**

**ptr1->next = ptr->next;**

**..........................................ptr->next=ptr1;**

**cout << "\nНомер:";**

**cin >> ptr1->nomer;**

**cout << "\nСреден успех:";**

**cin >> ptr1->sr\_uspeh;**

**}**

**Задача 23 {6 т.}** Какъв е резултатът от изпълнението на програмата?

#include <iostream.h>

class base

{public:

base(int x = 1)

{b = x;}

base& operator=(const base &x)

{if (this!=&x) b = x.b + 1;

return \*this;

}

void Print() const

{cout << "base: " << b << endl;

}

private: int b;

};

class der1 : public base

{public:

der1(int x = 1) {d = x;}

der1& operator=(const der1& x)

{if (this!=&x)

Име: Фак. № - 12 - / 16

{d = x.d + 2;

base::operator=(x);

} return \*this;

}

void Print() const

{cout << "der1: " << d << endl;

base::Print();

}

private: int d;

};

class der2 : public base

{public:

der2(int x = 2) : base(x+3)

{d = x;

}

der2(const der2& p)

{d = p.d + 2;

}

der2& operator=(const der2& x)

{if(this !=&x)

{d = x.d + 3;

base::operator=(x);

} return \*this;

}

void Print() const

{cout << "der2: " << d << endl;

base::Print();

}

private: int d;

};

class der3 : public der2

{public:

der3(int x = 3)

{d = x;

}

der3(const der3& p) : der2(p)

{d = p.d+3;

}

void Print()

{cout << "der3: " << d << endl;

der2::Print();

}

private: int d;

};

void main()

{der1 d11(2), d12=d11;

der2 d21(1), d22;

der3 d31(3), d32=d31;

cout << "d11: "; d11.Print();

cout << "d12: "; d12.Print();

d12 = d11; cout << "d12: "; d12.Print();

cout << "d21: "; d21.Print();

cout << "d22: "; d22.Print();

d22 = d21; cout << "d22: "; d22.Print();

cout << "d31: "; d31.Print();

cout << "d32: "; d32.Print();

d32 = d31; cout << "d32: "; d32.Print();

}

Задраскайте грешните линии на програмата. Какъв е резултатът от изпълнението на

програмата след отстраняване на грешките в нея? (Напишете резултата отстрани на програмата).

#include <iostream.h>

class Base

{public:

virtual void func1()

{ cout << "func1() \n";

}

void help()

{ cout << "help()\n";

func2();

func1();

func3();

}

private:

virtual void func2()

{ cout << "func2()\n";

}

protected:

virtual void func3()

{ cout << "func3()\n";

}

};

class Der : public Base

{virtual void func1()

{ cout << "Der-class\n";

}

protected:

virtual void func2()

{ cout << "Der-func2()\n";

}

public:

virtual void func3()

{ cout << "Der-func3()\n";

}

};

void main()

{Base b; Der d;

Base \*p = &b;

b.func1();

p->func1();

p->func2();

p->func3();

Base \*q = &d;

q->func1();

q->func2();

q->func3();

p->Base::func1();

Der \*r = new Der;

r->func2();

r->func1();

r->func3();

p->help();

q->help();

r->help();

}

Шаблонът на класа LList реализира свързан списък, представен чрез една връзка.

template <class T>

struct elem\_link1

{T inf;

elem\_link1<T> \*link;

};

template <class T>

class LList

{public:

LList();

~LList();

LList(LList const&);

LList& operator=(LList const &);

void IterStart(elem\_link1<T>\* = NULL);

elem\_link1<T>\* Iter();

void DeleteElem(elem\_link1<T> \*p, T & x); // изтрива сочения от p елемент и го записва в x

………….

Име: Фак. № - 10 - / 16

private:

elem\_link1<T> \*Start, // указател към началото

\*End, // указател към края

\*Current; // итератор

…

};

**Задача 30 {4 т}** Напишете какво извежда тази програма.

#include <iostream.h>

class Point {

public:

Point(float=0.0, float=0.0);

~Point();

protected:

float x, y;

};

Point::Point(float a, float b) {

x = a; y = b;

cout << "Конструктор на Point: " << '[' << x << ", " << y << ']' << endl;

}

Point::~Point() {

cout << "Деструктор на Point: " << '[' << x << ", " << y << ']' << endl;

}

class Circle:public Point {

public:

Circle(float r=0.0, float x=0.0, float y=0.0);

~Circle();

Име: Фак. № - 12 - / 16

private:

float radius;

};

Circle::Circle(float r, float a, float b):Point(a, b) {

radius = r;

cout << " Конструктор на Circle: " << radius << '[' << a << ", " << b << ']'

<< endl;

}

Circle::~Circle () {

cout << "Деструктор на Circle: " << radius << '[' << x << ", " << y << ']'

<< endl;

}

void main() {

{Point p(1.1, 2.2);}

Circle circle1(4.5, 7.2, 2.9);

Circle circle2(10, 5, 5);

}

#include <iostream.h>

class Base1

{public:

Base1(int a = 1)

{n = a;

x = 1.5;

cout << "Base1: " << n << ", " << x << endl;

}

Base1(const Base1& p)

{n = p.n;

x = p.x;

cout << "Base1.n: " << n << endl

<< "Base1.x: " << x << endl;

}

Base1& operator=(const Base1& p)

{if(this != &p)

{n = p.n + 5;

x = p.x + 1.5;

cout << "Base1.n: " << n << endl

<< "Base1.x: " << x << endl;

}

return \*this;

}

private:

int n;

double x;

};

class Base2

{public:

Base2(double b = 1)

{n = 1;

y = b;

cout << "Base2: " << n << ", " << y << endl;

}

private:

int n;

double y;

};

class Base3

{public:

Base3(double b = 1)

{n = 3;

x = b;

cout << "Base3: " << n << ", " << x << endl;

}

Base3(const Base3& p)

{n = p.n + 1;

x = p.x + 1.3;

cout << "Base3.n: " << n << endl

<< "Base3.x: " << x << endl;

}

Име: Фак. № - 8 - 17

private: Изпишете решението тук:

int n;

double x;

};

class Der : public Base2, protected Base1, Base3

{public:

Der(int x = 1, int y = 2, int z = 3) : Base1(x), Base2(y), Base3(z)

{n = z;

m = x\*y;

cout << "Der: " << n << ", " << m << endl;

}

Der& operator=(const Der& p)

{if(this != &p)

{Base1::operator =(p);

n = p.n;

m = p.m;

}

return \*this;

}

private:

int n, m;

};

void main()

{Der d1, d2(3, 4, 5), d3(6,7), d4;

Der d5 = d1;

Der d6(d3);

d4 = d2;

}

**Задача 13.** Отстранете грешните обръщения към функции в main( ) .

Какъв ще бъде резултатът от изпълнението на програмата

след отстраняване на грешките?

#include <iostream.h>

class B

{public:

virtual void p( ) const

{cout << "B::p( )\n";

}

private:

virtual void s( ) const

{cout << "B::s( )\n";

}

protected:

virtual void t( ) const

{cout << "B::t( )\n";

}

public:

void usual( ) const

{cout << "B::usual( )\n";

p( );

s( );

t( );

}

};

class D : public B

{private:

void p( ) const

{cout << "D::p( )\n";

}

public:

void s( ) const

{cout << "D::s( )\n";

}

protected:

void t( ) const

{cout << "D::t( )\n";

}

};

class F : public D

{protected:

void s( ) const

{cout << "F::s( )\n";

}

private:

void t( ) const

{cout << "F::t( )\n";

}

public:

void p( ) const

{cout << "F::p( )\n";

}

private:

void usual()

{cout << "F::usual( )\n";

s( );

t( );

}

};

Задача. 11 (6 т.) Посочете и коментирайте грешките в дефинициите на конструкторите в

следната йерархия:

base

der

der1 der2

der3

#include <iostream.h>

class base

{private: int a1;

public:

void read(int x = 0)

{a1 = x;

}

void display()

{cout << "a1: " << a1 << endl;

}

};

class der : public base

{private: int d1;

public:

der(int x) : der(x=0);

void display()

{cout << "d1: " << d1 << endl;

cout << "base::display():" << endl;

base::display();

}

};

der::der(int x): base(x=5)

{cout << "constructor der\n";

d1 = x;

}

class der1 : public der

{private: int d1;

public:

void display()

{cout << "der1 - member d1: " << d1 << endl;

cout << "der::display():" << endl;

der::display();

}

};

class der2 : public der

{private: int d1;

public:

der2()

{d1=0;

}

der2(int x) : der(x=0);

void display()

{ cout << "der2 - member d1: " << d1 << endl;

cout << "der::display():" << endl;

der::display();

}

};

der2::der2(int x) : base(x)

{cout << "constructor der2\n";

d1 = x;

}

class der3 : public der2

{private: int d1;

public:

void display()

{cout << "der3 - member d1: " << d1 << endl;

cout << "der1::display():" << endl;

der1::display();

}

};

void main()

{der a(1); a.display();

der1 b; b.display();

der2 c(1); c.display();

der3 d; d.display();

} Задача 12 (8 т.) Задраскайте грешните обръщения към функции в главната функция main

на програмата. Какъв е резултатът от изпълнението й след отстраняване на грешките?

(Напишете резултата отстрани на програмата).

#include <iostream.h>

class Base

{public:

void help()

{cout << "help()\n";

f();

h();

g();

}

virtual void f()

{cout << "f()\n";

}

protected:

virtual void h()

{cout << "h()\n";

}

private:

virtual void g()

{cout << "g()\n";

}

};

class Der1 : public Base

{protected:

virtual void f()

{cout << "Der1 class\n";

Base::f();

}

public:

virtual void h()

{cout << "Der1-h()\n";

}

virtual void g()

{cout << "Der1-g()\n";

}

};

class Der2 : public Base

{ virtual void h()

{cout << "Der2 class\n";

Base::h();

}

protected:

virtual void g()

{cout << "Der2-g()\n";

}

virtual void f()

{cout << "Der2-f()\n";

}

};

void main()

{ Base \*x = new Base;

Base \*y = new Der1;

Der2 o;

Base \*z = &o;

x->f(); y->f(); z->f();

x->h(); y->h(); z->h();

x->g(); y->g(); z->g();

Der1 \*p = new Der1;

Der2 \*q = new Der2;

o.f(); p->f(); q->h();

o.g(); p->g(); q->h();

x->help(); y->help(); z->help();

o->help(); p->help();

delete x; delete y; delete p; delete q;

}

**Задача 9 (8 т.)**

Напишете резултата от изпълнението на програмата:

#include <iostream.h>

class A

{public:

A(int a = 0)

{n = a;

x = 1.5;

cout << "A.n::" << n << ", A.x::" << x << endl;

}

A(const A& p)

{n = p.n;

x = p.x;

cout << "A.n::" << n << endl

<< "A.x::" << x << endl;

}

A& operator=(const A& p)

{if(this!=&p)

{n = p.n + 1;

x = p.x + 1.5;

cout << "A.n:: " << n << endl

<< "A.x:: " << x << endl;

}

return \*this;

}

private:

int n;

double x;

};

class B

{public:

B(double b = 2.5)

{n = 2;

x = b;

cout << "B.n::" << n << ",B.x::" << x << endl;

}

private:

int n;

double x;

};

class C

{public:

C(double b = 3.5)

{n = 3;

x = b;

cout << "C.n::" << n << ",C.x::" << x << endl;

}

C(const C& p)

{n = p.n + 3;

x = p.x + 3.5;

cout << "C.n:: " << n << endl

<< "C.x:: " << x << endl;

}

private:

int n;

double x;

};

class D : protected B, A, public C

{public:

D(int x = 1, int y = 2, int z = 3): A(x), B(y), C(z)

{n = z;

m = x\*y;

cout << "D.n::" << n << ",D.m::" << m << endl;

}

D& operator=(const D& p)

{if(this!=&p)

{A::operator =(p);

n = p.n;

m = p.m;

}

return \*this;

}

private:

int n, m;

};

void main()

{D d1, d2(1,2,3), d3(4, 5);

D d4 = d1;

d3 = d2;

}

**Задача 5 (10 т.)**

Разгледайте програмата:

#include <iostream.h>

class Base

{public:

virtual void virt1()

{cout << "Base::virt1() \n";

}

Base()

{cout << "Base()\n";

virt1();

virt2();

virt3();

}

private:

virtual void virt2()

{cout << "Base::virt2()\n";

}

protected:

virtual void virt3()

{cout << "Base::virt3()\n";

}

};

class Der1 : public Base

{ void virt1()

{cout << "Der1::virt1()\n";

}

protected:

void virt2()

{cout << "Der1::virt2()\n";

}

public:

void virt3()

{cout << "Der1::virt3()\n";

}

};

class Der2 : public Der1

{protected:

void virt1()

{cout << "Der2::virt1()\n";

}

public:

void virt2()

{cout << "Der2::virt2()\n";

}

private:

void virt3()

{cout << "Der2-virt3()\n";

}

};

void main()

{ Base b;

Der1 d1; Der2 d2;

Base \*p = &d1;

Der1 \*q = &d2;

b.virt1();

b.Base();

p->virt1();

p->virt2();

p->virt3();

q->virt1();

q->virt2();

q->virt3();

q->Base();

p = &d2;

p->virt1();

p->virt2();

p->virt3();

Der1 \*r = new Der2;

r->virt2();

r->virt1();

r->virt3();

delete r;

}

a) Намерете и обяснете грешките в процедурата main на горната програма.

b) Кои връзки в програмата се разрешават статично и кои - динамично?

c) Какъв е резултатът от изпълнението на програмата след отстраняване на неправилните обръщения

към виртуалните функции?

**Задача 15. (8 т.)**Намерете резултата от изпълнението на програмата.

**#include <iostream.h>**

**class A**

**{public:**

**A(int a, double\* b)**

**{n = a;**

**x = new double;**

**\*x = 1.5;**

**cout << "A: " << n << "," << \*x << endl;**

**}**

**A(const A& p)**

**{n = p.n;**

**x = new double;**

**\*x = \*p.x;**

**cout << "A(const A&): " << n << endl**

**<< \*x << endl;**

**}**

**A& operator=(const A& p)**

**{if(this != &p)**

**{delete x;**

**n = p.n + 5;**

**x = new double;**

**\*x = \*p.x + 1.5;**

**cout << "A::operator=(p): " << n << endl**

**<< \*x << endl;**

**}**

**return \*this;**

**}**

**~A()**

**{cout << "~A()\n";**

**delete x;**

**}**

**private:**

**int n;**

**double\* x;**

**};**

**class B**

**{ public:**

**B(int a, double b)**

**{n = a;**

**x = b;**

**cout << "B: " << n << "," << x << endl;**

**}**

**~B()**

**{cout << "~B()\n";**

**}**

**private:**

**int n;**

**double x;**

**};**

**class C**

**{public:**

**C(int a, double b)**

**{n = a;**

**x = b;**

**cout << "C: " << n << "," << x << endl;**

**}**

**~C()**

**{cout << "~C()\n";**

**07.07.2007г. ДИ ОКС „Бакалавър” по Информатика, СУ-ФМИ фак. № стр. 20/28**

**}**

**C(const C& p)**

**{n = p.n + 3;**

**x = p.x + 1.5;**

**cout << "C(const C&): " << n << endl**

**<< x << endl;**

**}**

**private:**

**int n;**

**double x;**

**};**

**class D : public B, protected C, A**

**{public:**

**D(int x=1, int y=3, double z=2.5): A(x, &z), B(y+x, z-x), C(x, z)**

**{n = y;**

**m = x;**

**cout << "D: " << n << "," << m << endl;**

**}**

**~D()**

**{cout << "~D()\n";**

**}**

**D& operator=(const D& p)**

**{if(this!=&p)**

**{A::operator =(p);**

**cout << "D::operator=(p)\n";**

**n = p.n;**

**m = p.m;**

**}**

**return \*this;**

**}**

**private:**

**int n, m;**

**};**

**void main()**

**{D x(2), y(3, 7), z(5, 3, 1);**

**D t = x;**

**z = y;**

**}**

**Задача 1. (5 т.)** Ако, в резултат на успешна компилация на зададения по-долу код на С, е създаден изпълним файл с име **progA** и в текущата директория има текстов файл **fileA.txt** със съдържание

**xyz123-y**

**111bbb**

**Zzzzzz5yyyyy**

напишете вдясно какво ще се изведе на стандартния изход в случай на успешно изпълнение на **progA** след стартиране с командния ред

./progA fileA.txt

#include <stdio.h>

#include <fcntl.h>

main( int argc, char \*argv[] )

{

int fd, i = 0 , j ;

char c;

if ( ( fd = open ( argv[1], O\_RDONLY ) ) == -1 )

{ printf (“\n Cannot open %s ”, argv[1] ); exit(1); }

while ( read ( fd, &c, 1 ) )

{ ++i;

if ( c = = '\n' )

{ if ( i < 10 )

for (j = i ; j < 15 ; j++ )

write(1,”$”,1);

i=0;

}

write(1,&c,1);

}

close(fd);

V I G

**Задача 5.** Даден е следният програмeн код на езика С,

в който са използвани системни примитиви на ОС UNIX и LINUX:

#include <fcntl.h>

main(int argc, char \*argv[ ] )

{

int fd ;

close ( 1 );

if ( ( fd=open (argv [ 1 ], O\_CREAT|O\_WRONLY|O\_TRUNC, 0666) ) == -1 ) exit ( 1 ) ;

write ( fd, "Hello ", 6 ) ;

write ( 1, "END", 3 ) ;

}

Заградете с кръгче всички възможни резултати от изпълнението на този програмен код.

a) Низът “Hello END” ще бъде изведен на терминала.

б) Низът “Hello ” ще бъде записан във файл,

чието име е подадено като първи аргумент в командния ред,

а низът “END” ще бъде изведен на терминала.

в) Низът “Hello END” ще бъде записан във файл,

чието име е подадено като първи аргумент в командния ред.

г) Процесът ще приключи връщайки код на завършване 1.

**C**

Задача **4.** (5т.) Зададен е следния фрагмент от програма:

int filed, i;

filed = creat ( “exam\_txt”, 0777 );

close ( 1 );

i = dup (filed);

write ( i, “TEST\n”, sizeof(“TEST \n” ) );

write ( 1, “TEST \n”, sizeof(“TEST \n” ) );

Като резултат от изпълнението на последователността от зададените оператори**:**

o на терминала ще се изведе два пъти низа “TEST”

o на терминала ще се изведе низа “TEST” и във файла “exam\_txt” ще се запише низа “TEST”

o във файла “exam\_txt” ще се запише два пъти низа “TEST”

**Задaча 2 (3 т.)**

Даден е следният фрагмент от програма на С, в който са използвани системни примитиви на ОС UNIX и

LINUX:

int fd;

write ( 1, “START\n”, sizeof(“START\n” );

close ( 1 );

fd = creat ( “file.txt”, 0666 );

write ( fd, “START\n”, sizeof(“START\n” );

write ( 1, “START\n”, sizeof(“START\n” );

Напишете вдясно какво и къде ще бъде изведено като резултат от изпълнението на този фрагмент.