

---

Aufgaben zur Klausur **Objektorientierte Programmierung** im WS 95/96 (IA42)

Zeit: 60 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 6 Seiten

---

## Aufgabe 1:

Gegeben sei das folgende Programm:

```
#include <iostream.h>

int linecnt = 0;
int adc, aic, acc, ass, ads;

void begin () {
    adc=aic=acc=ass=ads=0;
}
void end () {
    ++linecnt;
    cout << linecnt << " : " << adc << ", " << aic << ", " << acc
        << ", " << ass << ", " << ads << endl;
}

class A {
    int i;
public:
    A() : i(1) { ++adc; }
    A(int i1) : i(i1) { ++aic; }
    A(const A & a1) : i(a1.i) { ++acc; }
    A & operator = (const A & a1) {
        i = a1.i;
        ++ass;
        return *this;
    }
    ~A() { ++ads; }
};

class B : public A {
    A d;
public:
    B() {};
    B(int i1) : A(i1),d(i1+1) {}
};

int main() {
    begin();
    { B o1; A o2; } end();

    begin();
    { A o1(3); B o2(15); } end();

    begin();
    { B o1, o2(o1), o3(o2); A o4(o1); } end();
```

```
begin();
{ B o3[3]; } end();

begin();
{ struct { struct {B f11,f12;} f1; A f2; } o4; } end();

begin();
{ B o1; o1 = 15; o1 = B(); } end();

return 0;
}
```

Welche Ausgabe erzeugt dieses Programm

- 1) .....
- 2) .....
- 3) .....
- 4) .....
- 5) .....
- 6) .....

## Aufgabe 2:

Gegeben sei das folgende Programm, in dem mehrfache Vererbung verwendet wird:

```
#include <iostream.h>

int sum = 0;

class X {
    int xi;
public:
    X(int i = 42) : xi(i) { sum += i; }
    int value() const { return xi; }
};

class Y : public X {
    int yi;
public:
    Y(int i) : X(2*i), yi(i) { sum += i; }
    int value() const { return yi + X::value(); }
};

class Z : public X {
    int zi;
public:
    Z(int i) : X(3*i), zi(i) { sum += i; }
    int value() const { return zi + X::value(); }
};

class P : public Y, public Z {
    int pi;
public:
    P(int i) : Y(2*i), Z(3*i), pi(i) { sum += i; }
    int value() const { return pi + Y::value() + Z::value(); }
};

int main() {
    P p(1);
    cout << sum << ", " << p.value() << ", "
         << p.Y::value() << ", " << p.Z::value() << endl;
    return 0;
}
```

Welche Ausgabe erzeugt dieses Programm

1) .....

### Aufgabe 3:

Gegeben sei das folgende Programm, das mit virtuellen Funktionen arbeitet:

```
#include <iostream.h>

int linecnt = 0;

class A { public:
    virtual const char * f1 () const { return " A"; }
    virtual const char * f2 () const { return " A"; }
    virtual void g () const { cout << ++linecnt << f1() << f2() << endl; }
};

class B : public A { public:
    virtual const char * f1 () const { return " B"; }
};

class C : public B { public:
    virtual const char * f2 () const { return " C"; }
};

int main() {
    B b1; b1.g();
    C c1; c1.g();

    A d1(b1); d1.g();
    B e1(c1); e1.g();

    A & a3 = b1; a3.g();
    A & a4 = c1; a4.g();

    C * p2 = & C(); p2->g();
    A * p3 = & B(); p3->g();
    A * p4 = p3; p4->g();

    return 0;
}
```

Welche Ausgabe erzeugt dieses Programm

1) .....

2) .....

3) .....

4) .....

5) .....

6) .....

7) .....

8) .....

9) .....

