

Zwischentest Objektorientierte Programmierung 2006

Aufgabe 1: Fehlersuche

In folgendem Programmcode sind 6 Fehler eingebaut. Streichen Sie die Fehler an und kommentieren Sie sie mit kurzen Stichworten. (6 Punkte)

```
#include <iostream>

using namespace stdin;

class A
{
    int var1;
public:
    A()
    {
        var1=15;
    }

    C()
    {
        cout<<"Funktion b";
    }
}

class B : public A
{
    int var2;
public:
    B()
    {
        var2 = 23;
        var1 = 24;
    }
};

int main()
{
    B test;
    delete test;
    return 0;
}
```

Aufgabe 2: Virtuelle Funktionen

```
class A                                //Headerdateien der Einfachheit halber weggelassen
{
    public:
        void test()
        {
            cout<<"test A"<<endl;
        }
};

class B : public A
{
    public:
        void test()
        {
            cout<<"test B"<<endl;
        }
};

int main()
{
    A* ptrA = new A;
    B* ptrB = new B;
    A* ptrC = new B;
    ptrA->test();
    ptrB->test();
    ptrC->test();
    return 0;
}
```

a)

Welche Ausgabe hat das Programm? (3 Punkte)

b) Welche Ausgabe hat das Programm, wenn die Methode test() in Klasse A wie folgt geändert wird? (3 Punkte)

```
virtual void test()
{
    cout<<"test A"<<endl;
}
```

Aufgabe 3: Vererbung

```
class A          //Headerdateien der Einfachheit halber weggelassen
{
    private:
        int x;
    protected:
        float y;
    public:
        int m;
        A(){}
        virtual void test() = 0;
        int getx() { return x; }
};

class B : public A{
    protected:
        int w;
    public:
        B(){}
        void test(){
            w = getx();
            cout<<"Hallo";
        }
};

class C : public A
{
    protected:
        int w;
    public:
        C(){}
};
```

a) Sind folgende Definitionen in einem Hauptprogramm möglich (mit Begründung)? (3 Punkte)

A a; _____

B b; _____

C c; _____

b) Weshalb wird in der Klasse B nicht direkt auf die Elementvariable x zugegriffen, sondern über die Methode getx()? (3 Punkte)

Aufgabe 4: Typwandlung

```
class A          //Headerdateien der Einfachheit halber
weggelassen
{
    public:
        A(){}
        void testA(){
            cout<<"Test A"<<endl;
        }
};

class B : public A
{
    int x;
    public:
        B(){ x=10; }

        void testB(){
            cout<<"Test B " << x <<endl;
        }
};

int main()
{
    A a;
    B b;
    B* b2;

    b2=_____ //Zu ergänzen in a)

    b2->testB(); //B*/

    b=0;         //C*/

    return 0;
}
```

a) Ergänzen Sie das Hauptprogramm in zulässiger Weise so, dass b2 auf das Objekt a verweist.
(2 Punkte)

b) Welche Ausgabe erzeugt das Programm bei dem mit /*B*/ bezeichneten Aufruf der Memberfunktion testB()? (2 Punkte)

c) Wie ist die Definition der Klassen A bzw. B zu ergänzen, dass die mit /*C*/ markierte Zuweisung im Hauptprogramm möglich wird? (3 Punkte)