



Aufgabe 13.1 (Ü) Polymorphie und Interfaces

Gegeben seien folgende Java-Klassen:

```
class A {
    public void f() { System.out.println("f()_in_A"); }
    public void g() { System.out.println("g()_in_A"); f(); }
}
class B extends A {
    public void f() { System.out.println("f()_in_B"); }
    public void h() { System.out.println("h()_in_B"); f(); g(); }
}
class C extends A {
    public void h() { System.out.println("h()_in_C"); f(); g(); }
}
interface I {
    public void m();
}
abstract class D {
    public abstract void n(boolean b);
    public void o() { System.out.println("o()_in_D"); n(false); }
}
class E extends D {
    public void n(boolean b) {
        System.out.println("n()_in_E");
        if(b) o();
    }
    public void m() { System.out.println("m()_in_E"); }
}
class F extends E implements I {
    public void n(boolean b) {
        System.out.println("n()_in_F");
    }
}
```

a) Folgende Anweisungen werden vom Compiler erfolgreich übersetzt:

- | | | |
|-----------------------------------|----------------------------------|---------------------------------|
| i) A a = new A(); | <input type="checkbox"/> Richtig | <input type="checkbox"/> Falsch |
| ii) B b = new B(); A t = b; | <input type="checkbox"/> Richtig | <input type="checkbox"/> Falsch |
| iii) B b = new A(); | <input type="checkbox"/> Richtig | <input type="checkbox"/> Falsch |
| iv) D d = new D(); | <input type="checkbox"/> Richtig | <input type="checkbox"/> Falsch |
| v) D d = new F(); | <input type="checkbox"/> Richtig | <input type="checkbox"/> Falsch |
| vi) F f = new F(); I i = f; | <input type="checkbox"/> Richtig | <input type="checkbox"/> Falsch |

b) Welche Ausgaben produzieren die folgenden Anweisungen?

i) `A a = new A(); a.g();`

.....
.....
.....
.....
.....

ii) `B b = new B(); b.h();`

.....
.....
.....
.....
.....

iii) `C c = new C(); c.g();`

.....
.....
.....
.....
.....

iv) `B b = new B(); A a = b; a.g();`

.....
.....
.....
.....
.....

v) `D d = new E(); d.n(true);`

.....
.....
.....
.....
.....

vi) `F f = new F(); f.o();`

.....
.....
.....
.....
.....

Aufgabe 13.2 (Ü) Fragestunde

Nutzen Sie die restliche Zeit dazu Ihrem Tutor fragen zu stellen. Bitte bereiten Sie Ihre fragen vor und lassen Sie diese vorher Ihrem Tutor per Email zukommen.

Aufgabe 13.3 (H) Interfaces

In dieser Aufgabe sollen die Klassen aus der Hausaufgabe von Blatt 12 (Geometrische Flächen und Prismen) erweitert werden.

- a) Einerseits sollen Klassen vom Typ `Prisma` und `Grundflaeche` untereinander und miteinander vergleichbar gemacht werden indem beide Klassen das Interface `java.lang.Comparable` implementieren.
 - Prismen werden nach ihrem Volumen (ist das Volumen gleich, dann nach Oberfläche) verglichen;
 - `Grundflaechen` nach ihrer Fläche verglichen;
 - Prismen werden über ihre `Grundflaeche` mit anderen `Grundflaechen` verglichen;
- b) Testen sie die Implementierung in dem sie ein Array von `Grundflaeche` mittels `java.util.Arrays.sort()` sortieren.
- c) Als zweiten Schritt sollen die Methoden `istQuadrat` und `zuQuadrat` in ein Interface `Quadrierbar` ausgelagert werden, welches nur von solchen `Grundflaechen` implementiert werden soll, die auch ein Quadrat darstellen können.
- d) Fügen sie eine weitere `Grundflaeche` `Dreieck` zur Modellierung von rechtwinkligen Dreiecken zu den bereits bestehenden `Grundflaechen` hinzu. Ein (rechtwinkliges) Dreieck wird durch die Länge der Hypotenuse und Ankathete parametrisiert.
- e) Zuletzt erstellen wir ein weiteres Interface `Polygon` mit nur einer Methode `int getEckenAnzahl()`, welche die in einer Grundfläche enthaltenen Ecken zurückgibt.
- f) Implementieren sie `Polygon` für alle `Grundflaechen`, die eine endliche Anzahl von Ecken besitzen.
- g) Testen sie `Quadrierbar` und `Polygon` in dem sie über ein sortiertes Array vom Typ `java.lang.Comparable`, welches `Prisma` und `Grundflaechen` enthält, iterieren und mittels `instanceof` die Verfügbarkeit beider Interfaces abfragen und, falls vorhanden, die aus diesen Interfaces ermittelbaren Informationen ausgeben.