

Zusatzaufgaben Klassen, Vererbung, Schnittstellen, Enums

Was ist die Ausgabe folgenden Codes

```
public class MyStuff {
    MyStuff(String n) { name = n; }
    String name;
    public static void main(String[] args) {
        MyStuff m1 = new MyStuff("guitar");
        MyStuff m2 = new MyStuff("tv");
        System.out.println("gleich: "+m2.equals(m1));
    }
    public boolean equals(Object o) {
        MyStuff m = (MyStuff) o;
        if(m.name != null)
            return true;
        return false;
    }
}
```

- gleich: true
- gleich: false
- das Programm lässt sich nicht übersetzen
- das Programm wirft einen Laufzeitfehler

A sei eine Klasse, B und D seien interfaces, C sei eine abstrakte Klasse und E ein enum welche der folgenden Deklarationen sind korrekt?

- class F implements B,C{...}
- class F extends C implements B{...}
- class F extends B,D{...}
- class F extends B,C{...}
- class F extends E{...}
- class F extends A implements B,D{...}
- class F extends A,B implements D{...}

Welche Aussagen treffen im Zusammenhang von Vererbung zu?

- Jede Klasse in Java erbt von der Klasse java.lang.Object.
- Wenn in einer Klasse eine Methode m() überschrieben wird, dann kann in dieser Klasse trotzdem die Methode m() der Superklasse aufgerufen werden.
- In Java können Klassen beliebig viele Superklassen haben.
- Interfaces ohne Methoden und Attribute sind unsinnig und werden daher in Java Programmen nicht verwendet
- Eine Java Klasse lässt sich nur kompilieren, wenn in den interfaces, die sie implementiert keine Namenskonflikte auftreten, also alle Methoden unterschiedliche Namen/Typen haben.

Welche Aussagen treffen über Vererbung und Konstruktoren zu?

- Konstruktoren können durch Vererbung überschrieben werden.
- Vor der Ausführung eines Konstruktors einer Klasse X werden immer Konstruktoren aller Superklassen von X ausgeführt.
- Im Konstruktor einer Klasse kann man den Standardkonstruktor der selben Klasse mit dem Befehl `this () ;` aufrufen.
- Jede Klasse hat mindestens einen Konstruktor.
- Der explizite Aufruf eines Konstruktors der Superklasse mit `super (...)` ; muss, falls vorhanden, immer der erste Befehl in der Implementierung eines Konstruktors sein.

Welche Aussagen treffen auf Java Aufzählungstypen (`enum`) zu?

- Objekte eines Aufzählungstyps haben immer auch Methoden
- Alle Attribute eines Aufzählungstyps müssen statisch (`static`) sein
- In Aufzählungstypen muss immer ein Konstruktor implementiert werden
- Aufzählungstypen lassen sich durch Klassen erweitern
- Aufzählungstypen können Schnittstellen implementieren

Welche Aussagen treffen auf das Arbeiten mit dynamischer Polymorphie in Java zu?

- Methoden von Schnittstellen (`interface`) haben immer implizit den Sichtbarkeitsmodifikator `public`
- Wenn ein Objekt zum Typ X deklariert wird, dann kann dieses Objekt nur mit einem Konstruktor der Klasse X erzeugt werden
- Alle Java Klassen und Aufzählungstypen (`enum`) haben mindestens eine gemeinsame Superklasse.
- Jede Klasse kann höchstens eine Schnittstelle (`interface`) implementieren.
- Weil Mehrfachvererbung in Java verboten ist kann jede Klasse höchstens eine Superklasse haben.