

Fragen, Antworten, Kommentare zur aktuellen Vorlesung

Bitte an das Ausfüllen des Evaluationsbogens (s. Mail vom 16.12) denken.

Frage: Ich habe beim googeln über Interfaces gesehen, das da auch eine Programmierung drin stattfinden kann. Das finde ich nicht in den Vorlesungsfolien.

Antwort: Das stimmt, dies ist nicht Teil der Veranstaltung, da die ergoogelte Idee nicht für die klassische Objektorientierung steht und Java-spezifisch ist. Zentrale Bedeutung hat das vorgestellte Interface-Konzept als vollständig abstrakte Klasse, die so als Vorgabe zur Realisierung dient und flexibel gegen andere Programmierungen ausgetauscht werden kann.

Das Schlüsselwort Interface gibt es nebenbei in C++ selbst nicht, aber die Idee als vollständig abstrakte Klasse findet sich dort genauso wieder und hat die gleiche elementare Bedeutung wie in anderen objektorientierten Sprachen.

In Java wurde das Interface-Konzept aufgeweicht, da für neue Java-Versionen der Wunsch entstand neue Methoden in Interfaces zu packen. Wird dies durch eine einfache Ergänzung gemacht, müssten in allen Klassen, die das Interface realisieren, diese Methode neu implementiert werden. Damit wäre eine neue Java-Version inkompatibel mit der vorherigen, was immer eine schlechte Idee ist (fragen sie PHP oder Python). Der leicht schmutzige Trick ist, einfach zur neuen Methode doch eine konkrete Implementierung anzugeben, die mit dem Schlüsselwort default gekennzeichnet wird. Diese Methode kann dann in der realisierenden Klasse implementiert also überschrieben werden, muss es aber nicht. Das folgende Beispiel zeigt dies und noch eine Ergänzung.

```
public interface MeinInterface {

    public void zeich42(); // abstrakt, muss realisiert werden

    default public void zeich43() { // mit Standard-Implementierung
        System.out.println("XLIII");
    }

    default public void zeich44() {
        System.out.println("XLIV");
    }

    public static void klassenmethode() {
        System.out.println("geht in Java-Interface, "
            + "hat aber mit Vererbung nix zu tun.");
    }
}

public class MeineRealisierung implements MeinInterface{

    MeineRealisierung(){
    }
}
```

```

@Override
public void zeich42(){
    System.out.println("42");
}

@Override
public void zeich44(){
    System.out.println("44");
}

public static void klassenmethode() {
    System.out.println("geht in Java-Interface, "
        + "hat aber mit Vererbung nix zu tun.");
}
}

public class Main {
    public static void main(String[] s) {
        MeinInterface mi = new MeineRealisierung();
        mi.zeich42();
        mi.zeich43();
        mi.zeich44();
        MeineRealisierung.klassenmethode(); //unsauber
        MeinInterface.klassenmethode(); // besser
    }
}

```

Die Ausgabe lautet:

```

42
XLIII
44
geht in Java-Interface, hat aber mit Vererbung nix zu tun.
geht in Java-Interface, hat aber mit Vererbung nix zu tun.

```