Interfaces

- Ein Interface enthält nur Methodenköpfe und Konstanten
- Methodenköpfe sind automatisch public

[public] interface identifier [extends interface1 [, interface2 …]] {

. . .

}

Beispiel: java\geometrie\Forms2D.java

```
package java.geometrie;

public interface Forms2D {

        double berechneFlaeche();

        double berechneUmfang();

        }
```

Ein Interface ist eine Vorschrift. In unserem Beispiel schreibt das Interface vor, welche Methoden für ein 2D-Objekt mindestens vorhanden sein müssen.

Verwendung:
class Rechteck implements Forms2D
→ alle Interface-Methoden müssen implementiert werden

Eine Klasse kann mehrere Interfaces implementieren
class BlaBla implements Forms2D, GrafikObjekt

Verwendung: java\geometrie\Rechteck.java

```
package java.geometrie;

public class Rechteck implements java.geometrie.Forms2D {

        . . .

        public double berechneFlaeche() {

                return getBreite() * getLaenge();

                }

        public double berechneUmfang() {

                return 2.0 * (getBreite() + getLaenge());

                }

        }
```

Verwendung: java\geometrie\Kreis.java

```
package java.geometrie;
public class Kreis implements java.geometrie.Forms2D {
    . . .
    public double berechneFlaeche() {
        return Math.PI * getRadius() * getRadius();
        }
    public double berechneUmfang() {
        return 2.0 * Math.PI + getRadius();
        }
    }
```

Interface Polymorphismus

Wie schon im Fall der Vererbung von Klassen gibt es Polymorphismus auch im Zusammenhang mit Interfaces.

```
class GeometrischeObjekte {
    public static void main(String[] args) {
        Forms2D R1;
        R1 = new Rechteck();
        Forms2D K1;
        K1 = new Kreis();
        . . .
        }
    }
```

Default Methoden:

Bis zur Version Java 7 standen in einem Interface nur Methodenköpfe. Seit Java 8 kann ein Interface unter Anderem auch Default-Methoden enthalten. Eine Default-Methode ist eine Methode mit einer Default-Implementierung.

```
package java.geometrie;
public interface Forms2D {
    default double berechneFlaeche() {return 0;}
    default double berechneUmfang() {return 0;}
    }
```

Hintergrund dieser neuen Möglichkeit ist folgender:

Ohne Default-Methoden ist es mitunter sehr aufwändig bis unmöglich, ein bestehendes Interface um eine Methode zu erweitern, denn das würde bedeuten, dass weltweit alle Klassen, die dieses Interface implementiert haben, angepasst werden müssen (die zusätzliche Methode muss in allen Klassen implementiert werden), auch wenn diese Klassen diese neue Methode gar nicht benötigen.

Das Problem mit dieser Neuerung ist, dass es dem Compiler nicht mehr auffällt, wenn Sie z.B. in der Klasse Rechteck vergessen, die Methode berechneUmfang() zu implementieren (der Compiler sieht die Default-Implementierung und daher ist es für den Compiler ok). Damit wird der Grundgedanke von Interfaces, eine Vorschrift darzustellen, welche Methoden in einer Klasse unbedingt implementiert werden müssen, verwässert.

Meine Empfehlung lautet daher: Verzichten Sie in der Regel auf Default-Methoden.

Unterschiede Interface – abstrakte Klasse

- Eine abstrakte Klasse dient in der Regel als Mutterklasse für abgeleitete Klassen und wird mit *extends* geerbt.
- Ein Interface ist eine Vorschrift, welche Methoden unbedingt in der Klasse, die dieses Interface mittels *implements* implementiert, programmiert werden müssen.
- Alle Methoden eines Interfaces sind quasi "abstract", da in einem Interface nur Methodenköpfe stehen.
- Ein Interface hat keine Variablen bzw. Attribute.
- Eine Klasse kann mehrere Interfaces implementieren.
- Eine Klasse kann von einer anderen Klasse erben und zusätzliche Interfaces implementieren.