



Kapitel 4: Interfaces

Lernziele

- [LZ 4.1] Erklären können, was ein Java Interface ist und was es enthalten kann
- [LZ 4.2] Die Interface-Syntax anwenden können für Definition und Nutzung von Interfaces
- [LZ 4.3] Das Comparable-Interface kennen und anwenden können
- [LZ 4.4] Das Comparator-Interface kennen und anwenden können
- [LZ 4.5] Die UML-Darstellung von Interfaces in Klassenmodellen kennen und anwenden



4. Interfaces

[LZ 4.1] Erklären können, was ein Java Interface ist und was es enthalten kann

- Erklären Sie anhand eines ausführlichen Beispiels Syntax und Bestandteile eines Java Interface
- Was kann nicht in einem Interface definiert werden?

4. Interfaces

[LZ 4.2] Die Interface-Syntax anwenden können für Definition und Nutzung von Interfaces

Definieren Sie ein Interface „Wiegbar“, welches eine Methode `double gewichtInKG()` enthält. Definieren Sie weiterhin zwei Klassen `Obst` und `Auto`, die das Interface implementieren (Konstruktoren s. `main`-Methode unten). Testen Sie ihren Code anhand folgender `main`-Methode, in der ein Array von Objekten, die alle das `Wiegbar`-Interface anbieten, aufgebaut wird. Abschließend wird über das Array iteriert (Polymorphismus, s. Kapitel 8) und das Gewicht jedes Objekts ausgegeben.

```
static public void main(String[] args) {
    Wiegbar[] objekteMitGewicht = new Wiegbar[] {
        new Auto("Audi Q5", 1830), new Obst("Birne", 0.175)
    };
    for (Wiegbar w : objekteMitGewicht)
        System.out.println(w.gewichtInKG());
}
```



4. Interfaces

[LZ 4.3] Das Comparable-Interface kennen und anwenden können

- Geben Sie die Definition des Comparable-Interface an.
- Schreiben Sie eine Obst-Klasse (Obst hat eine Bezeichnung und ein Gewicht in Gramm)
- Obst soll anhand seines Gewichts vergleichbar sein.
- Schreiben Sie eine main-Methode, die ein Array von Obst-Objekten belegt und es sortiert und ausgibt.

4. Interfaces

[LZ 4.4] Das Comparator-Interface kennen und anwenden können

- Geben Sie die Definition des Comparator-Interface an.
- Nutzen Sie die Obst-Klasse des vorherigen Beispiels. Geben Sie einen Comparator für Obst an, der Obst anhand der Bezeichnung absteigend vergleicht.
- Nutzen Sie diesen Comparator, um ein Array von Obst-Objekten einmal nach Gewicht (Standard-Vergleich) und einmal nach der Bezeichnung zu vergleichen und auszugeben.

4. Interfaces

[LZ 4.5] Die UML-Darstellung von Interfaces in Klassenmodellen kennen und anwenden

- Gegeben seien folgende Elemente:
 - eine Klasse Tisch, die vom der Klasse Möbelstück erbt
 - ein Interface Gegenstand mit einer Methode String getBezeichnung()
 - ein Interface Wiegbar mit einer Methode int getGewichtInGramm(), welches von Gegenstand ableitet
 - Tisch implementiert Wiegbar

- Erstellen Sie ein UML-Klassenmodell für obige Aussagen.



4. Interfaces

Übungsaufgabe (leicht)

- Definieren Sie ein Interface für die 4 Grundrechenarten basierend auf ganzen Zahlen.
- Implementieren Sie eine Klasse „Rechner“, die das Interface implementiert.
- Testen Sie die Klasse in `Rechner.main()`

4. Interfaces

Übungsaufgabe (Prüfungslevel)



Queue

- Eine Queue oder auch Warteschlange genannt ist eine häufig in der Informatik vorkommende Datenstruktur. Es lassen sich Werte (int) zu einer Queue hinzufügen und wieder entfernen. Es werden immer die zuerst hinzugefügten Werte entfernt. Dieses Verwaltungsprinzip nennt man deswegen auch first-in-first-out (FIFO). Eine Warteschlange kann zum Beispiel bei der Prozessumschaltung in einem Betriebssystem verwendet werden: der zuerst hinzugefügte (wartende) Prozess bekommt als nächstes wieder Prozessorzeit zugeordnet, vor allen nach ihm hinzugefügten Prozessen.
- Entwerfen Sie ein Interface Queue, mit dessen Methoden Werte hinzugefügt und wieder weggenommen werden können. Ebenso soll eine Methode existieren, die überprüft, ob die Queue leer ist oder nicht.
- Implementieren Sie diese Queue mit einem Feld. Wenn mehr Werte als das Feld fassen kann hinzugefügt werden, dann soll das Feld dieser Queue-Implementierung automatisch vergrößert werden: es kann zum Beispiel durch ein doppelt so großes Feld ersetzt werden.
- Testen Sie Ihre Implementierung ausgiebig. Insbesondere mit anfangs kleinen Felder und mehreren Hinzufüge- und Wegnahme-Operationen.