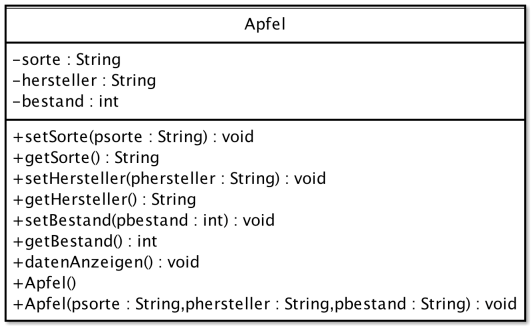
11 Die Geburt der Frösche — Java–Klassendefinitionen

Konzepte in diesem Kapitel

Objektorientierung/Java: Java–Klassendefinition, Datenfeld, Konstruktor, Standardmethode, Klassendokumentation

11.1 Der Aufbau einer Klassendefinition

Beispiel: Für eine Lagerverwaltungssoftware wurde die Klasse `Apfel` modelliert. Sie speichert die Apfelsorte, den Namen des Herstellers und den vorhandenen Lagerbestand. Die Klasse `Apfel` soll in Java programmiert werden.



Der Kopf einer Klassendefinition

Der Kopf einer Klassendefinition besteht aus der Sichtbarkeit, dem Schlüsselwort `class` und dem Namen der Klasse, der immer mit einem Großbuchstaben beginnt:

```
Sichtbarkeit          Name der Klasse
public    class  Apfel {
    Datenfelder
    Konstruktoren
    Standardmethoden
    sonstige Methoden
}
```

Die Datenfelder (Attribute)

Jedes Attribut einer Klasse wird in einer speziellen Variablen (*Datenfeld*) gespeichert. Da man auf die Datenfelder von außen nicht zugreifen darf, werden sie als `private` (verborgen) gekennzeichnet:

```
Sichtbarkeit  Datentyp  Name
private       String    sorte;
```

Die Konstruktoren

Ein *Konstruktor* ist eine spezielle Methode. Er sorgt dafür, dass ein Objekt bei seiner Erzeugung in einen gültigen Anfangszustand versetzt („initialisiert“) wird. Ein Konstruktor ist immer `public`, hat denselben Namen wie die Klasse und *keinen* Rückgabetyp, z.B.:

```
Sichtbarkeit  Name    Parameterliste
public    Apfel  () {
    sorte = "Boskop";
    hersteller = "FruchtCoop GmbH";
    bestand = 265;
}
```

Dieser Konstruktor erzeugt immer `Apfel`–Objekte der Sorte „Boskop“ vom Hersteller „FruchtCoop GmbH“ mit einem Bestand von 265 Stück. Ein erweiterter Konstruktor mit Parametern ermöglicht es, dem Objekt bei der Erzeugung Attributwerte zuzuweisen:

```
public Apfel(String psorte, String phersteller, int pbestand) {
    sorte = psorte;
    hersteller = phersteller;
    bestand = pbestand;
}
```

Eine Klassendefinition kann mehrere Konstruktoren mit verschiedenen Parameterlisten enthalten. Für die Erzeugung eines Objektes wählt man einen geeigneten aus.

Die Standardmethoden

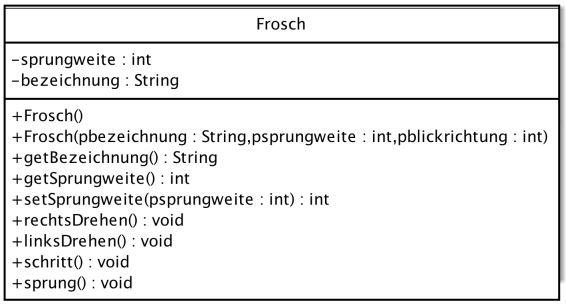
Standardmethoden dienen dazu, den Wert eines Datenfeldes abzufragen (Get–Methoden) oder zu ändern (Set–Methoden), z.B.:

```
public  String  getSorte  () {
    return sorte;
}
```

```
public  void  setSorte  (String psorte) {
    sorte = psorte;
}
```

Aufgabe 1: Erzeugen Sie eine neue Klasse `Frosch` als Unterklasse von `Actor`. Informieren Sie sich in der Schnittstelle von `Actor` über geeignete Methoden zum Abfragen und Ändern der Position und der Blickrichtung.

Aufgabe 3: Implementieren Sie die beiden im Klassendiagramm gezeigten Konstruktoren. Der zweite Konstruktor akzeptiert nur die Blickrichtungen `0`, `90`, `180` und `270`. Bei anderen Eingaben bleibt die Blickrichtung `0`.



Aufgabe 4: Implementieren Sie die Methoden. Beachten Sie, dass ein Frosch immer in Blickrichtung läuft oder springt.