

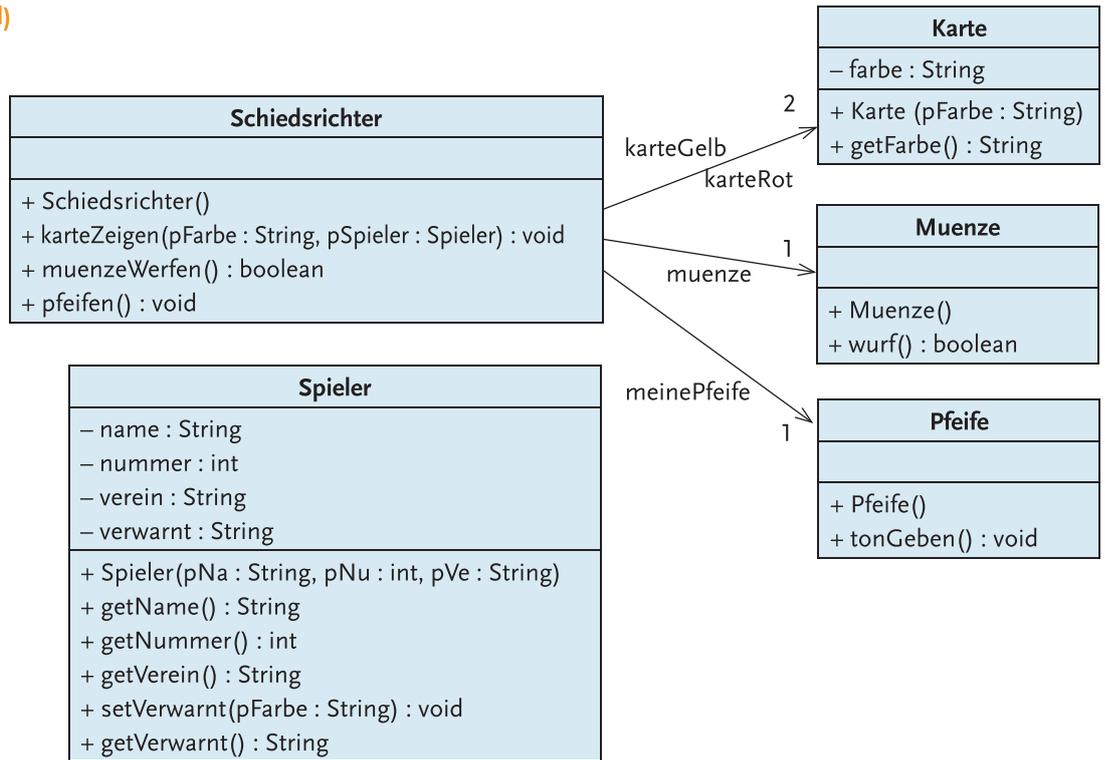
## 5.4 Klassen und Beziehungen implementieren

SB S. 114

1. a)
- | Zapfsaeule   |
|--|
| - preisBenzin : double<br>- preisDiesel : double<br>- gewaehlterKraftstoff : String<br>- abgegebene Liter : double |
| + Zapfsaeule()<br>+ gibKraftstoffAb(pArt : String, pMenge : double) : void<br>... alle get- & set-Methoden         |
- b)
- | Kamera   |
|--|
| - aufoesungHor : int<br>- aufoesungVer : int<br>- speicherGesamt : int<br>- speicherBelegt : int         |
| + Kamera(pAH: int, pAV : int, pSG : int, pSB : int)<br>+ filmen() : void<br>... alle get- & set-Methoden |
- c)
- | Luftballon   |
|--|
| - groesse : int<br>- farbe : String<br>- aufgeblasen : boolean   |
| + Luftballon(pGroesse: int, pFarbe : String, pAufgeblasen : boolean)<br>+ luftAufnehmen() : void<br>+ platzen() : void<br>... alle get- & set-Methoden |



d)



**Klasse Schiedsrichter**

Die Klasse *Schiedsrichter* kann über das Zeigen von Karten, Münzwürfe und Pfeifen Einfluss auf das Spiel nehmen.

**Konstruktor**

**Schiedsrichter**

Ein Objekt mit Festlegung der Attribute wird instanziiert.

**Auftrag**

**void karteZeigen (String pFarbe, Spieler pSpieler)**

Ein Objekt der Klasse *Spieler* wird mit einer Karte verwarnt, sodass dessen Attribut *verwarnt* auf *pFarbe* gesetzt wird.

Return: –

Parameter: Als Parameter werden der Spieler und die Farbkarte übergeben.

**Auftrag**

**boolean muenzeWerfen ()**

Für zu treffende Entscheidungen gibt der Wurf mit 50%iger Wahrscheinlichkeit *true* oder *false* zurück.

Return: Der Wurf liefert *true* oder *false* zurück.

Parameter: –

**Auftrag**

**void pfeifen ()**

Der Pfiff hat in dieser Simulation noch keine Auswirkungen.

Return: –

Parameter: –

**SB S. 115**

3. a) Entscheidend bei der Simulation ist, dass das Einräumen des Regals vom Mitarbeiter „angestoßen“ wird. Im Prinzip wird die Ware selbst nicht durch Methodenaufrufe aktiv, es wird lediglich eine Referenz zuerst vom Mitarbeiter und dann vom Regal auf die Ware gesetzt.