Objekte und Objektdiagramme

Abstraktion der Spielwelt

Unser Helden-Objekt befindet sich in einer rechteckigen Welt (ebenfalls eine Art von Objekt), die in einzelne quadratische Felder aufgeteilt ist. Abstrakt betrachtet interessiert uns also nur, an welcher x- und y-Koordinate sich der Held befindet und in welche Richtung er schaut. Wie im Einstieg liegt das Feld (0|0) oben links; `x` zählt die Spalten nach rechts, `y` die Zeilen nach unten.



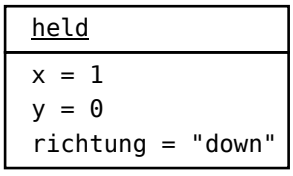
Objektdiagramme

In der objektorientierten Programmierung besitzen Objekte (Held, Knappe, Tür, Tor, Zettel, Monster …) einen eigenen Zustand und eigene Methoden. Aktionen werden als Nachrichten an das jeweilige Objekt formuliert, z. B. `held.geh()` oder `held.bediene_tor()`. Das erspart ständiges „Daten-Herumreichen“ und bindet das Verhalten an die zuständigen Objekte.

Im Framework wird das sichtbar:

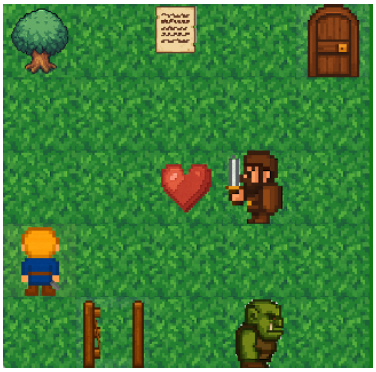
- **Der Zustand steckt im Objekt:** Jedes Objekt verwaltet seine „Variablen“ selbst.
- **Verhalten ist als Methode am Objekt implementiert:** `geh()`, `links()`, …
- **Message Passing:** Das aufrufende Objekt sendet eine Nachricht an ein Zielobjekt; das Ziel ändert seinen eigenen Zustand.

Der Zustand des Helden-Objekts im Bild oben kann wie folgt genauer beschrieben werden: Er befindet sich auf der x-Koordinate 1 und der y-Koordinate 0 und schaut nach unten. Wenn man dem Helden-Objekt den Namen `held` gibt, lässt sich sein Zustand mit folgendem **UML-Objektdiagramm** beschreiben (hierzu später mehr):



Dabei stehen `x` und `y` für die Koordinaten des Feldes, auf dem der Held steht, und `richtung` für die Richtung, in die er gerade schaut: `"up"`, `"down"`, `"left"` oder `"right"`.

Aufgabe 1



a) Beschreiben Sie alle Objekte im Bild oben mit Objektdiagrammen. Schreiben Sie jedes Diagramm wie oben: zuerst den Namen des Objekts, darunter jede Eigenschaft als `name = wert`. Mit **+ Objekt** legen Sie ein weiteres Diagramm an. Wenn Sie möchten, notieren Sie in einem dritten Abschnitt auch, was das Objekt kann.

(noch nicht bearbeitet)

b) Auch das Framework kennt den Zustand seiner Objekte. Öffnen Sie **Level 3B** (das Level 3A aus Heft 1), führen Sie ein kurzes Programm aus (z. B. `held.geh()`) und klappen Sie neben der Welt den **Objektinspektor** auf. Vergleichen Sie seine Angaben mit Ihren Objektdiagrammen.

(noch nicht bearbeitet)

Aufgabe 2

Objekte unserer Welt sind fassbare Gegenstände (z. B. Schuhe) oder auch Konstrukte unseres Denkens (z. B. ein Bankkonto). Betrachten wir Schuhe als Beispiel. Wenn man seine neuen Schuhe charakterisieren will, fallen einem sofort eine Reihe von Eigenschaften ein:



- Modell: Sneaker
- Größe: 40
- Farbe: rot
- Verschluss: Schnürsenkel
- …

a) Erstellen Sie ein passendes UML-Objektdiagramm zu einem beliebigen Paar Schuhe, das Sie besitzen.

(noch nicht bearbeitet)

b) Handelt es sich bei allen Eigenschaften der Schuhe um Text, also um Daten vom Typ String?

(noch nicht bearbeitet)

c) Welche Eigenschaft der Schuhe oben ist **kein** Text, sondern eine Zahl?

(noch nicht beantwortet)

Mit Objekten kann man auch bestimmte „Operationen“ ausführen: Zum An- und Ausziehen kann man die Schuhe z. B. in einem gewissen Sinn öffnen und schließen.

d) Listen Sie weitere Operationen auf, die mit Schuhen durchgeführt werden können. Ändern diese Operationen auch den Zustand der Schuhe?

(noch nicht bearbeitet)

e) Modellieren Sie ein Auto-Objekt, z. B. Ihr eigenes oder das Ihrer Eltern, als Objektdiagramm: Welche Eigenschaften hat es, und welche Operationen kann man damit ausführen?

(noch nicht bearbeitet)