

Praktische Informatik 1

Objekte und Klassen

Thomas Röfer
(mit Material von Karsten Hölscher)

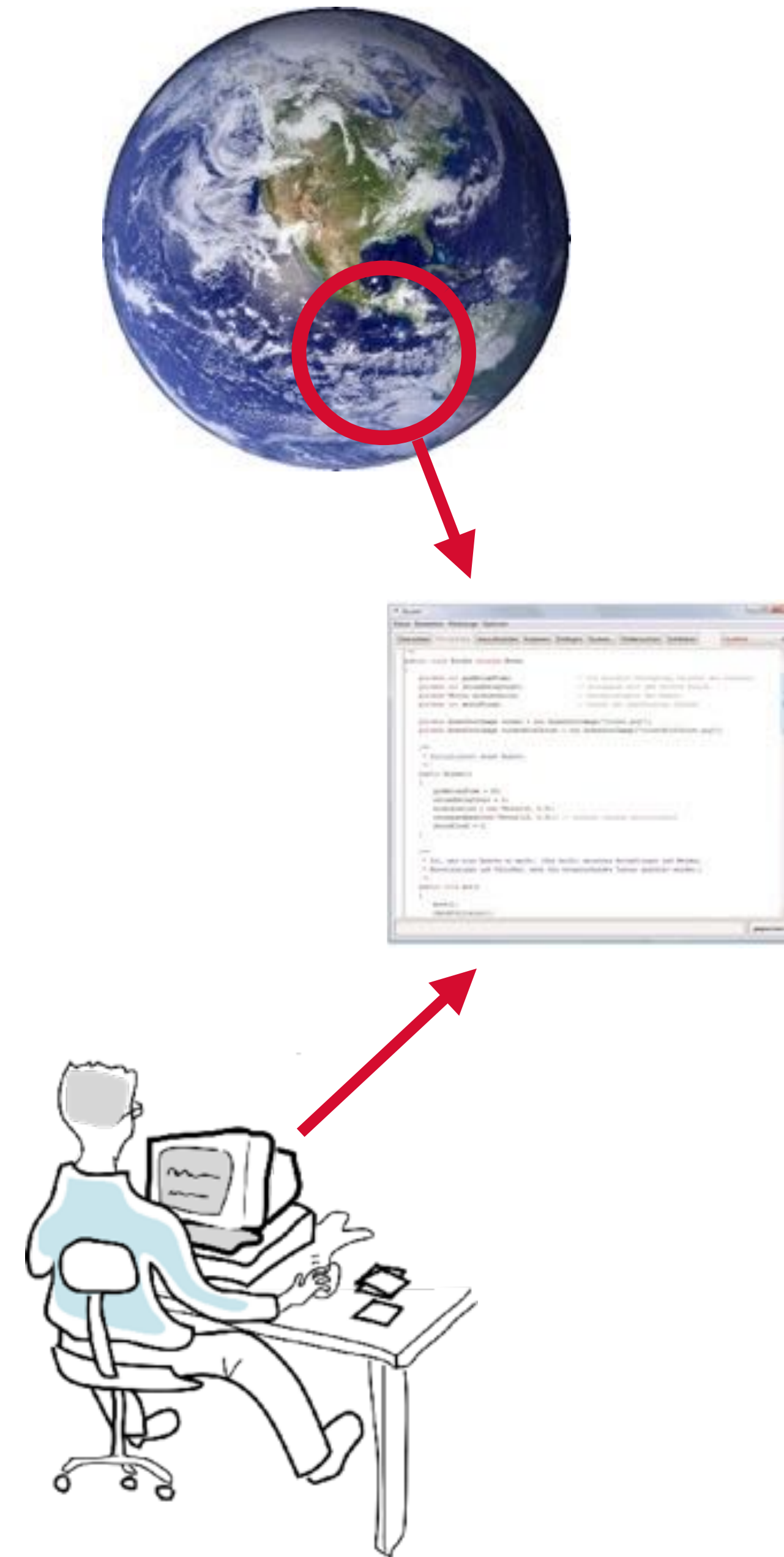
Cyber-Physical Systems
Deutsches Forschungszentrum für
Künstliche Intelligenz

Multisensorische Interaktive Systeme
Fachbereich 3, Universität Bremen



Objekt-orientierte Programmierung

- Computerprogramm
 - Modell eines Ausschnitts der realen Welt
- Objekt-orientierte Sichtweise
 - Ausschnitt besteht aus Objekten, die im Anwendungsbereich vorkommen
 - Diese Objekte müssen in Computermodell geeignet repräsentiert werden



Objekte

- „Dinge“ der realen Welt bzw. eines Problembereichs

z.B. dieses Pferd:



...oder dieses Auto:



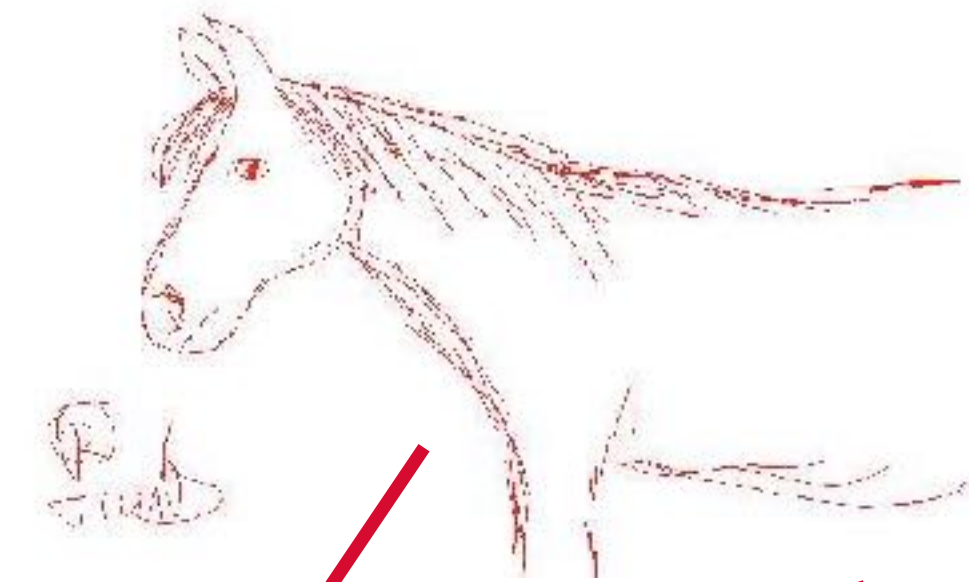
...oder diese Spielfiguren:



Klassen

- Eine **Klasse** beschreibt eine bestimmte **Art** von **Objekten**
- **Objekte** repräsentieren individuelle **Instanzen** einer **Klasse**
- Zu **einer** Klasse kann es **mehrere** Instanzen geben
- In Java: Objekte werden durch Klassen **erzeugt**

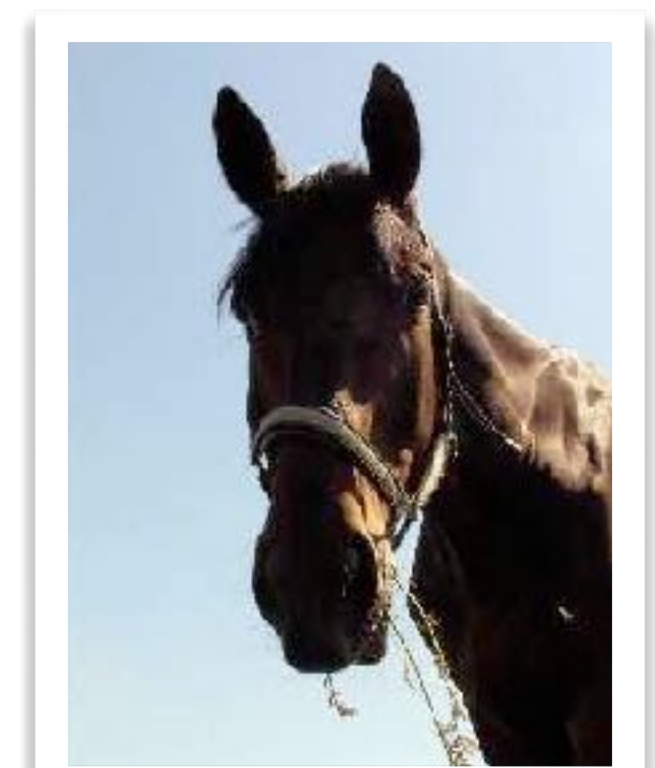
Klasse Pferd:



Objekt:
konkretes Pferd



Landina

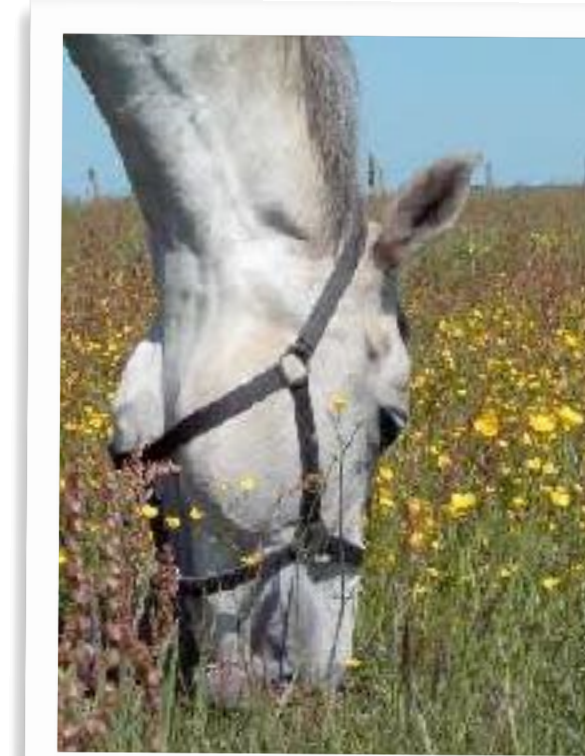


Pitu

Zwei **Instanzen**
der Klasse Pferd

Zustand

- Objekte haben **Attribute**, die ihren **Zustand** beschreiben
- Die Klasse definiert, welche Attribute die Objekte haben
- In Java wird der Zustand durch **Werte** repräsentiert, die in **Datenfeldern** gespeichert werden



Attributname	Wert
Name	Landina
Rasse	Hannoveraner
Alter	10
Stockmaß	1,78



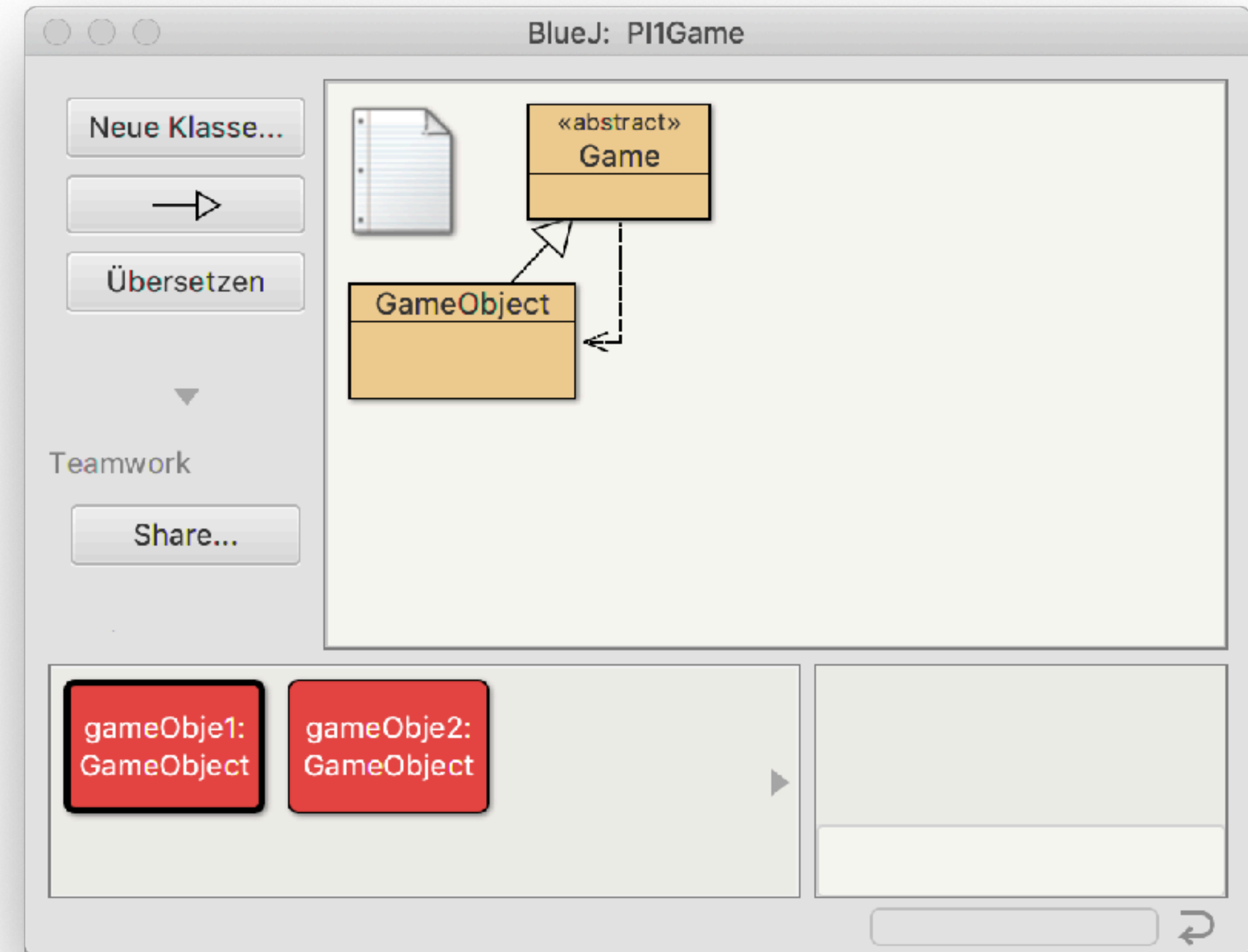
Attributname	Wert
Name	Pitu
Rasse	Oldenburger
Alter	12
Stockmaß	1,74

Funktionalität

- Objekte haben eine **Funktionalität**
 - Häufig ändert sie den Zustand des Objekts
- Diese wird in ihrer Klasse beschrieben
- In Java wird die Funktionalität in **Methoden** definiert
- Die Funktionalität können wir nutzen, indem wir eine Methode **aufrufen**

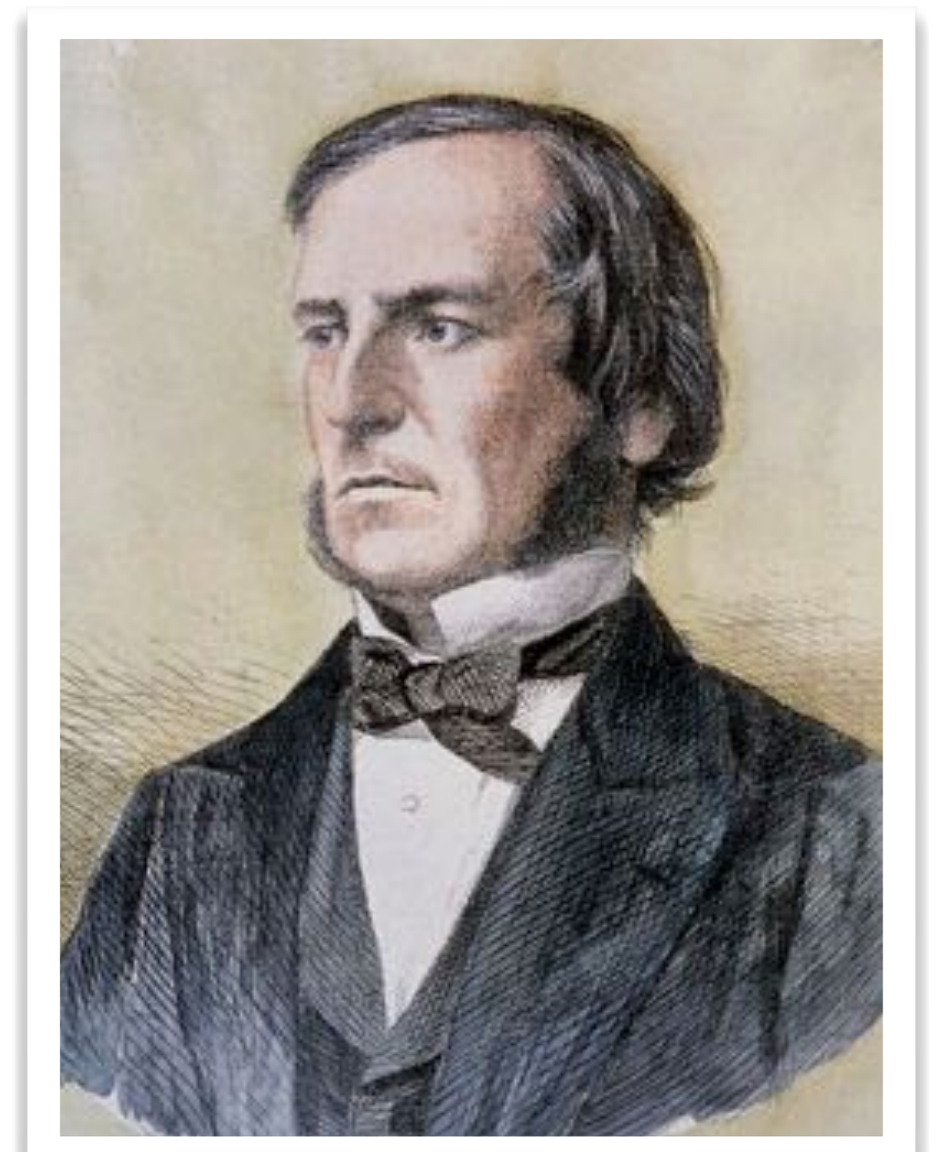


PI-1-Spielrahmen: Demo



Typen

- (**Daten**)**typen** legen fest, **welche Art Daten** an einer Stelle verwendet werden dürfen bzw. zurückgeliefert werden
- Wichtige Typen
 - **int** (Ganze Zahlen), Literale: **4, -8, 15, -16, 2342**
 - **boolean** (Wahrheitswerte), Literale: **false, true**
 - **String** (Zeichenketten, ist eine Klasse!),
Literale: **""**, **"Laaangweilig!!!!"**, **"Text mit Leerstellen"**, **"2342"**
- **Literale**: Wie man Werte direkt niederschreiben kann



George Boole
1815-1864

Methoden

- Kommunikation mit **Objekten** durch **Aufruf** von **Methoden**

- Methoden haben

- einen **Namen**

```
int getX()
```

- einen **Rückgabetyp**

```
int getY()
```

- eine (möglicherweise leere) Liste von **Parametern**

```
void setRotation(int rotation)
```

- Der **Name** und die **Liste der Parametertypen** wird als die **Signatur** einer Methode bezeichnet

Rückgabotyp

- **void**: Methode liefert kein Ergebnis zurück
 - Methodenaufruf entspricht einem **Befehl**
 - Ändert internen Zustand des Objekts (ohne viele Berechnungen → **Setter**) oder macht Ausgabe
- Anderer Typ: Methodenaufruf entspricht einer **Frage**
 - Berechnet etwas, oft ohne den internen Zustand des Objekts zu verändern
 - **Getter**: liefert nahezu direkt Attributwerte zurück

```
void playSound(String fileName)
```

```
void setRotation(int rotation)
```

```
boolean isVisible()
```

```
int getRotation()
```


Parameter

- Jeder Parameter hat
 - einen **Namen**
 - einen **Typ**
- Bezeichnung
 - **Formale Parameter**: Parameterdefinition in der Signatur
 - **Aktuelle Parameter**: Belegung der Parameter mit Werten während eines Methodenaufrufs
- Trennung mehrerer Parameter durch Kommas

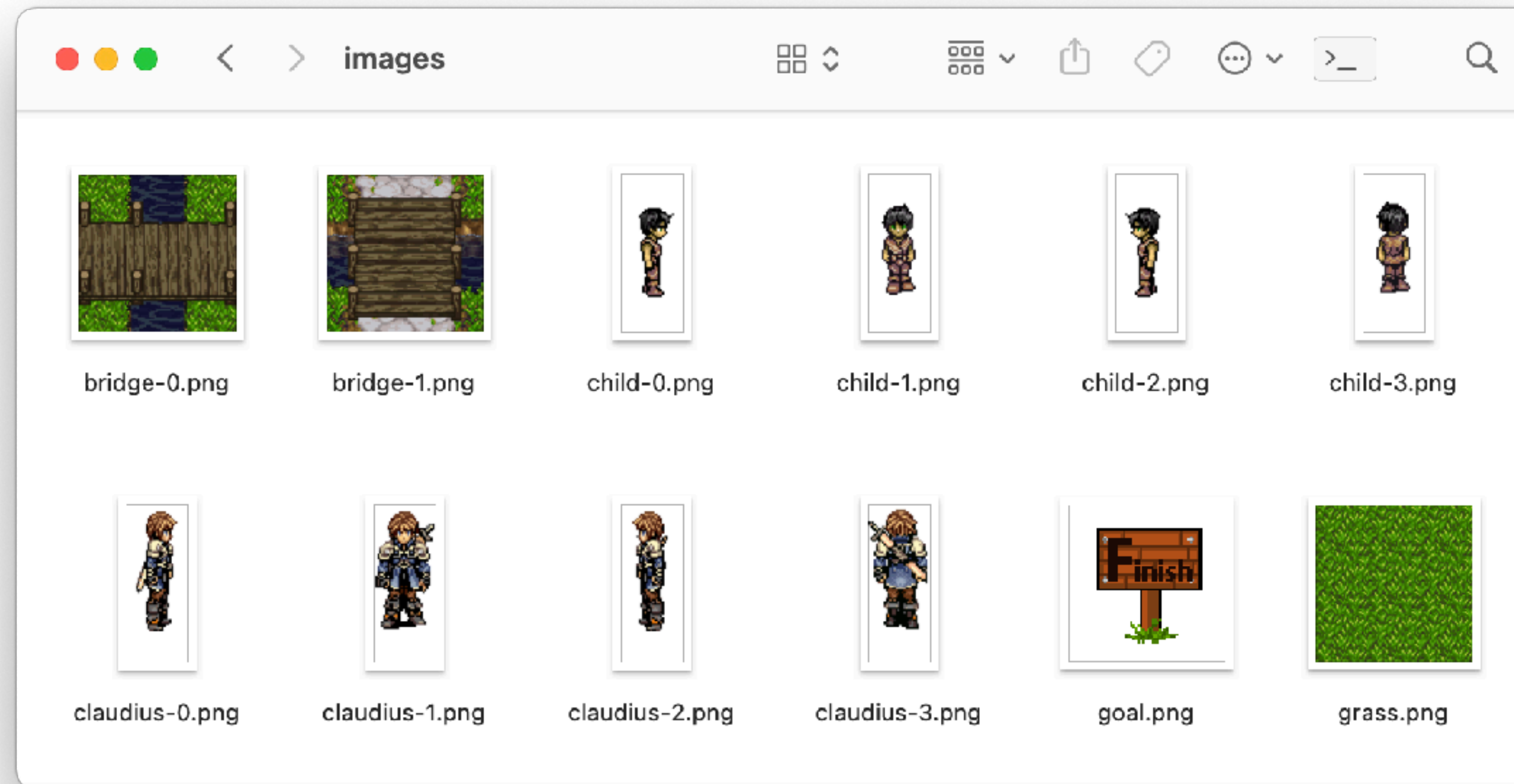
```
void setRotation(int rotation)
```

```
void setRotation(int rotation)
```

```
setRotation(2)
```

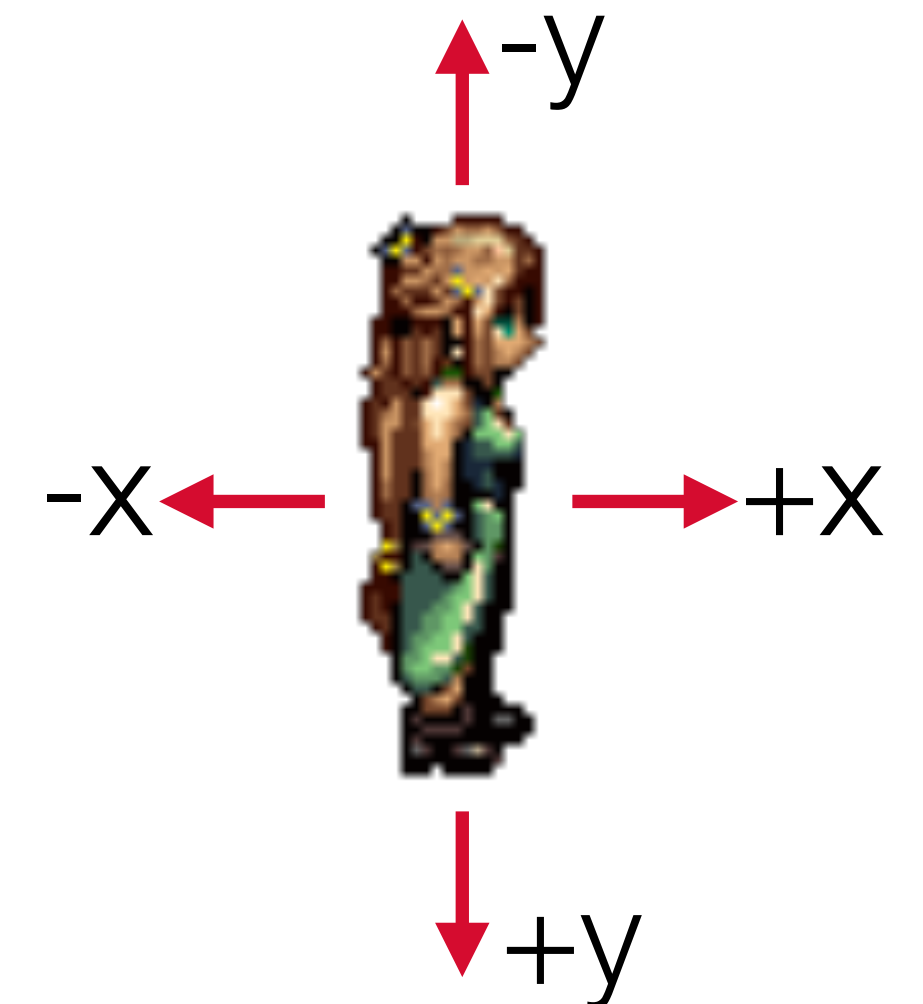
```
void setLocation(int x, int y)
```


Dateien den Spielrahmens: Demo

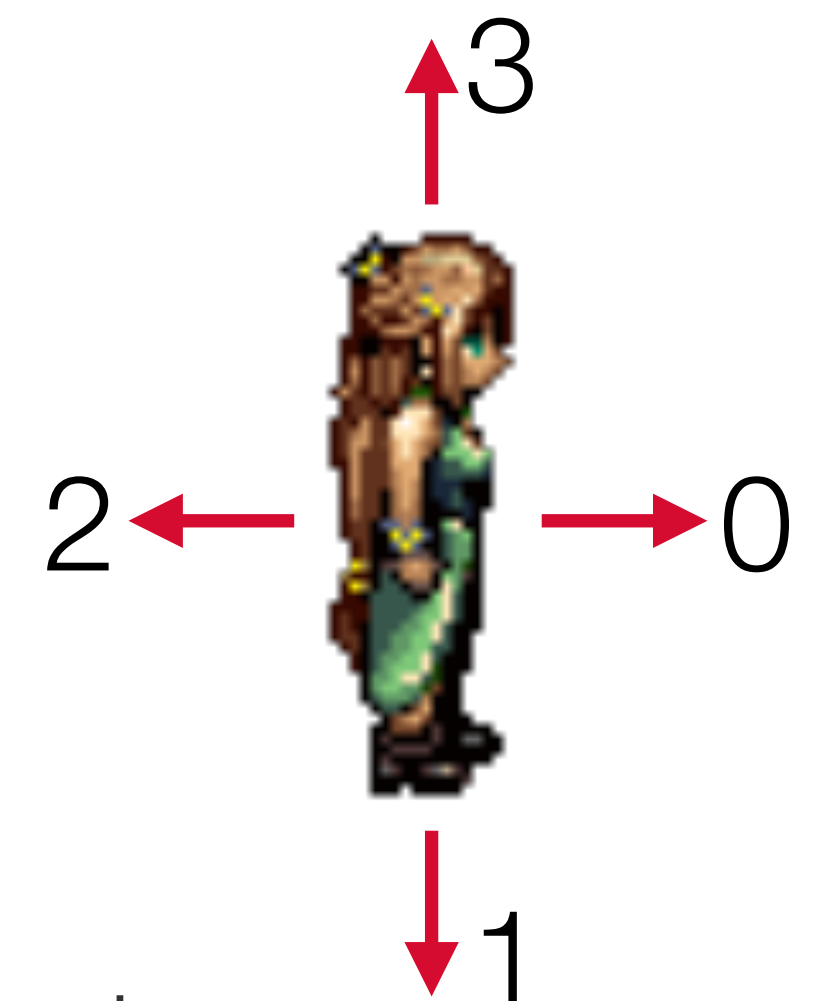


Der Spiel-Rahmen

- Positionen: Der Ursprung ist oben links
- Rotationen (**0** bis **3**)
 - Bild "**images/**<bildname>**-**<rotation>**.**<endung>" wird verwendet, wenn es existiert
 - Ansonsten wird "**images/**<bildname>**.**<endung>" verwendet
 - Wird keine Endung angegeben, wird **.png** verwendet
- Verdeckung
 - Objekte werden in der Reihenfolge ihrer letzten Bewegung/Platzierung gezeichnet



Position



Rotation

Age of Exception: Demo

- Von Robyn Kruse
- Bestes Spiel PI-1 2020/21
- Wieder Wahl des besten Spiels am Ende dieses Semesters!



Zusammenfassung der Konzepte

- **Objekt**, **Klasse** und **Instanz**
- **Attribut** und **Zustand**
- **Typ**
- **Methode**, **Signatur** und **Methodenaufruf**
- **Rückgabetyt** und **Parameter**

