LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Peter Brichzin

Krümel und Monster

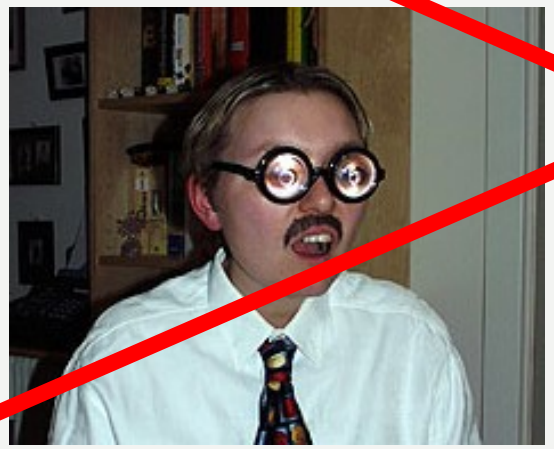
Themenzentrierter Ansatz für den Informatikunterricht der Jahrgangsstufe 10 am Beispiel einer Spielprogrammierung

Inhalt

- Informatik als neues Schulfach
- Fachdidaktik Informatik:
  Bedeutung des Modellierens im informationszentrierten Ansatz
- Unterrichtskonzept:
  didaktische Aspekte, Erfahrungen
- Zusammenfassung

Was ist Informatik?

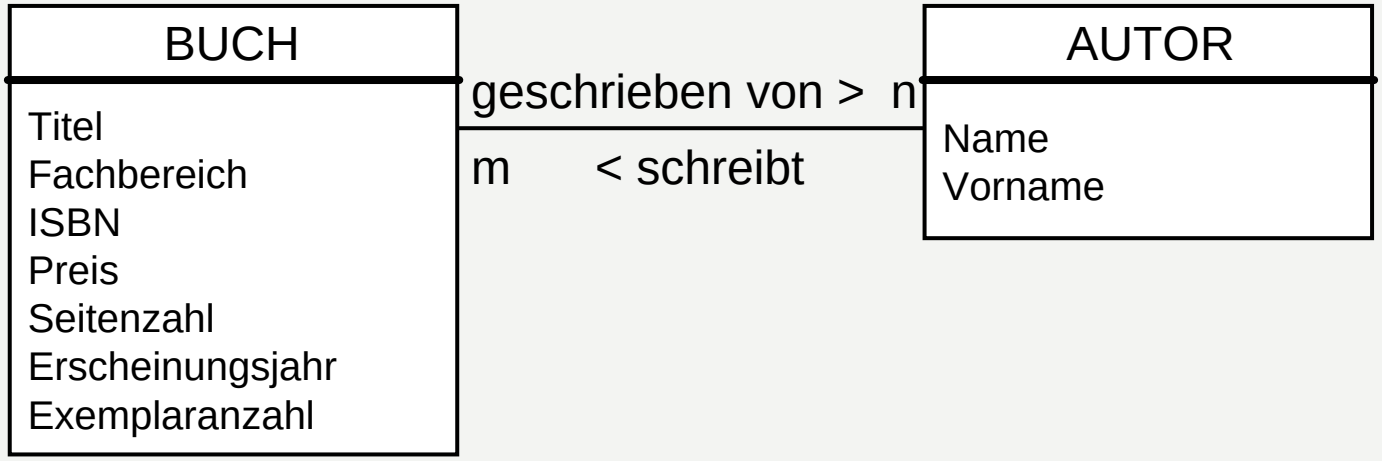
Informatik = "schöne bunte Word-Dokumente und PowerPoint-Folien erstellen"
Informatik = "Computer in Einzelteile zerlegen und wieder zusammenbauen"
Informatik = "**nur** Pascal programmieren"



Eine wichtige Fragestellungen der (Schul-) Informatik

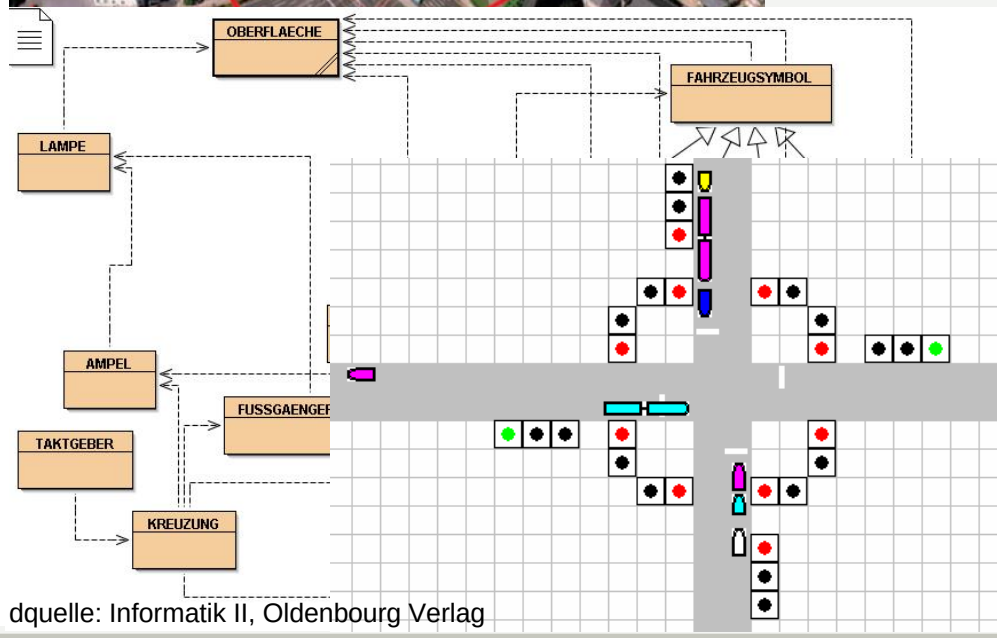
Wie kann man **Information** so darstellen, dass sie für eine Maschine eindeutig ist (formalisierte Darstellung)?

Beispiel: Bibliotheksverwaltung



Vom algorithmuszentrierten Ansatz zum informationszentrierten

*„Mit dem Vorschlag eines neuen, informationszentrierten Ansatzes für den Informatikunterricht am Gymnasium wollen wir eine **Verschiebung der inhaltlichen Schwerpunkte weg von Algorithmik und Programmierung hin zu fundamentalen Strukturierungstechniken für den Rohstoff "Information"** bewirken. … Wir wollen dabei nach dem Vorbild moderner Methoden der Softwareent-wicklung vor allem **graphische Modellierungstechniken** einsetzen, die … eine **detaillierte, aussagekräftige Beschreibung und Erfassung komplexer Systeme** ermöglichen. Die Verwendung des Rechners am Ende der Unterrichtseinheiten dient vor allem der Veranschaulichung der erarbeiteten Modelle.“ [Hubwieser, Broy, 1996]*



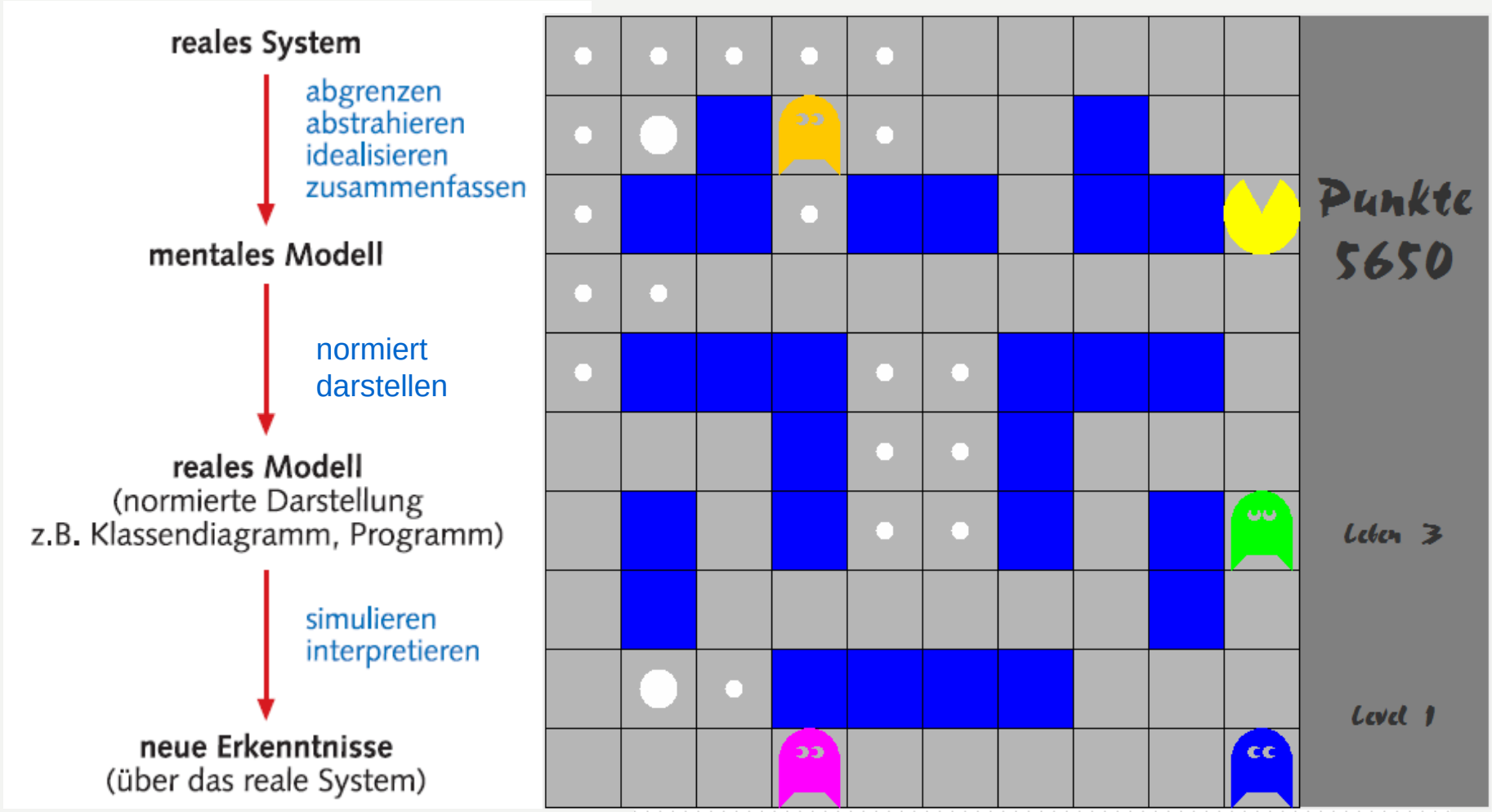
dquelle: Informatik II, Oldenbourg Verlag

Themenzentrierter Ansatz:

Behandlung aller Lerninhalte einer Einheit im Rahmen eines großen Themas

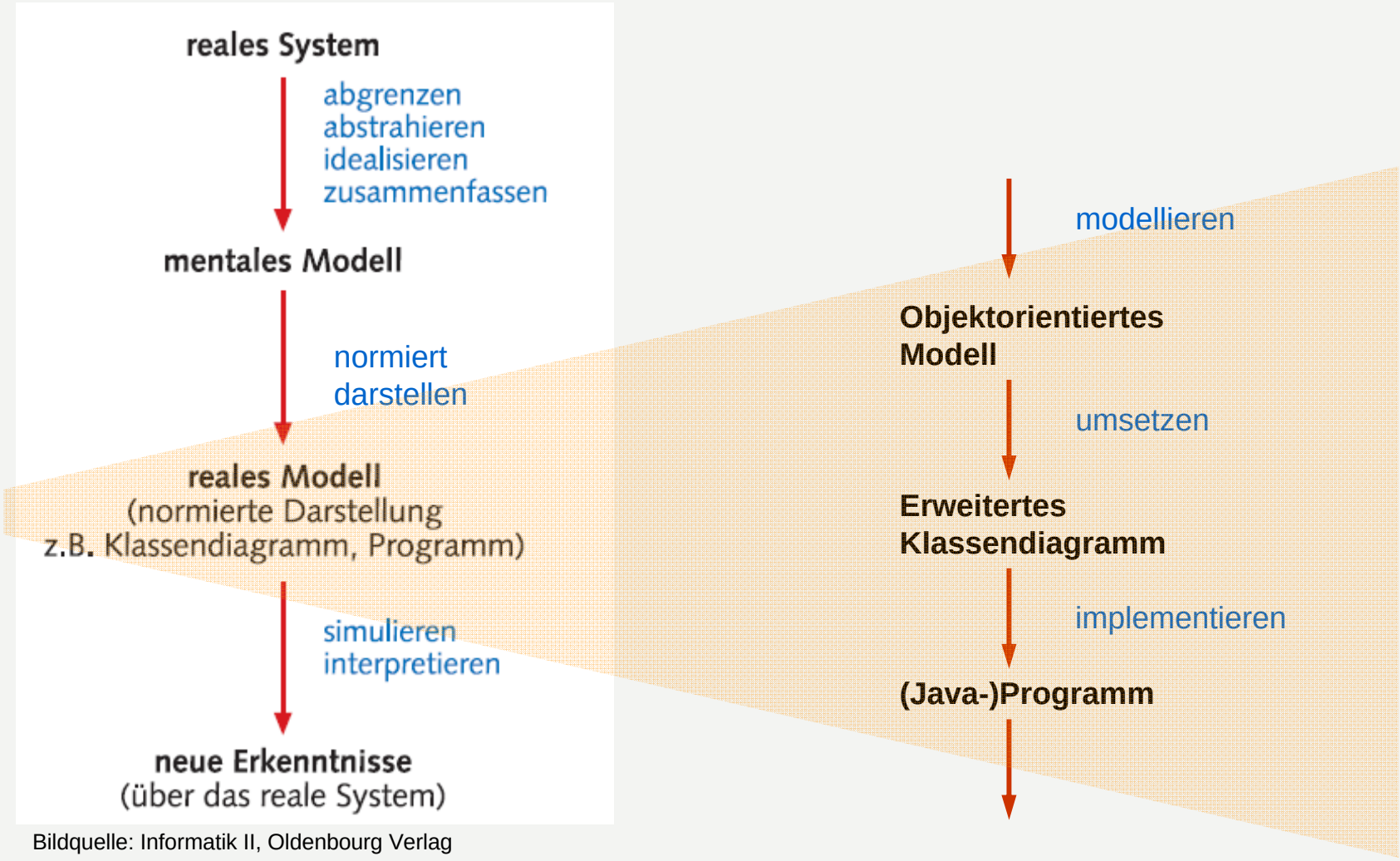
- *Behandlung großer, „sinnvoller“ Aufgaben – Softwareprojekte*
- *Hohes Anspruchsniveau durch Komplexität des Themas*
- *Permanente Motivation durch die Zielvorgabe des Themas*
- *Motivation des „nächsten Schritts“ ergibt sich automatisch durch die Themenstellung*
- *Eindenken in neue Themenstellungen ist nicht nötig*

Fachdidaktik Informatik
Modellieren (in der Informatik)



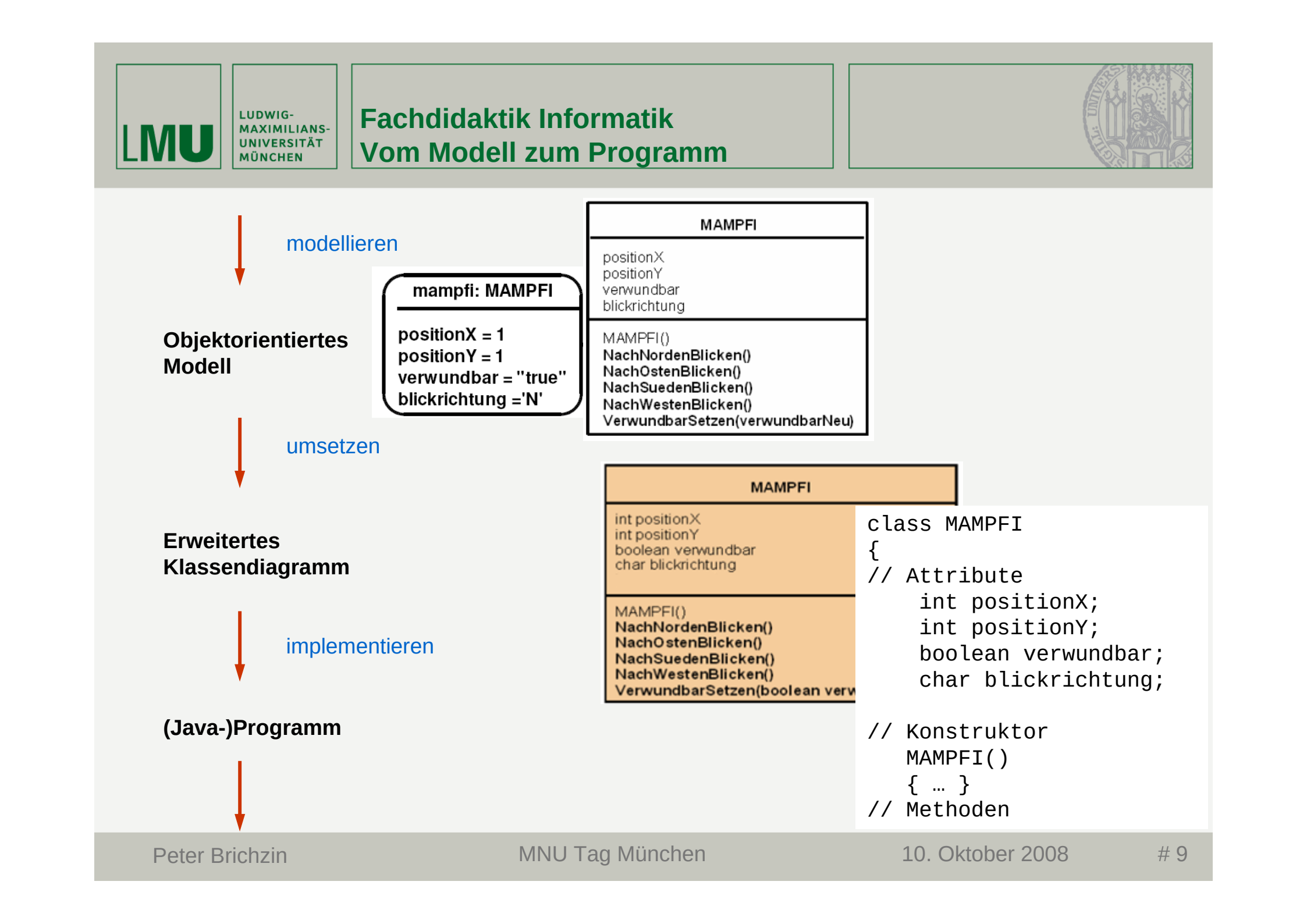
Bildquelle: Informatik II, Oldenbourg Verlag

Fachdidaktik Informatik
Modellieren (in der Informatik)



Bildquelle: Informatik II, Oldenbourg Verlag

Fachdidaktik Informatik
Vom Modell zum Programm

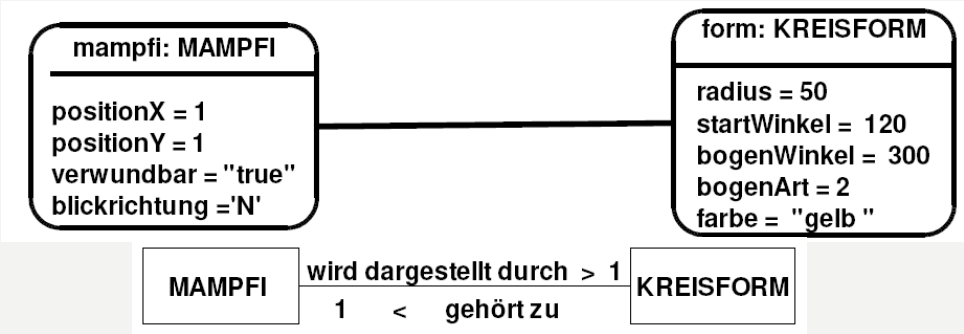


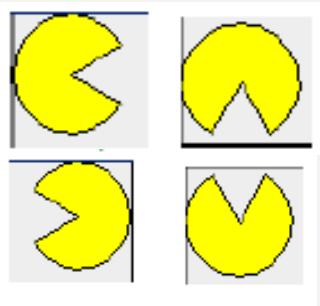

```
class MAMPFI
{
// Attribute
    int positionX;
    int positionY;
    boolean verwundbar;
    char blickrichtung;

// Konstruktor
   MAMPFI()
   { … }
// Methoden
```

Objektbeziehung - Objektkommunikation

Objektbeziehung:

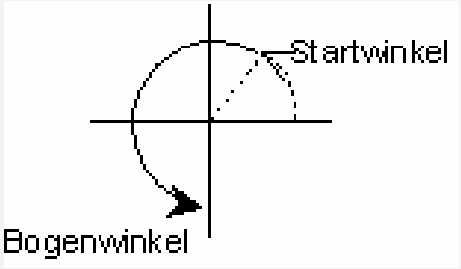
Darstellung von mampfi durch eine grafische Form / ein Symbol





Änderung der Blickrichtung von mampfi

→ Änderung des Werts des Attributs *startwinkel* des zugehörigen Kreisform-Objekts



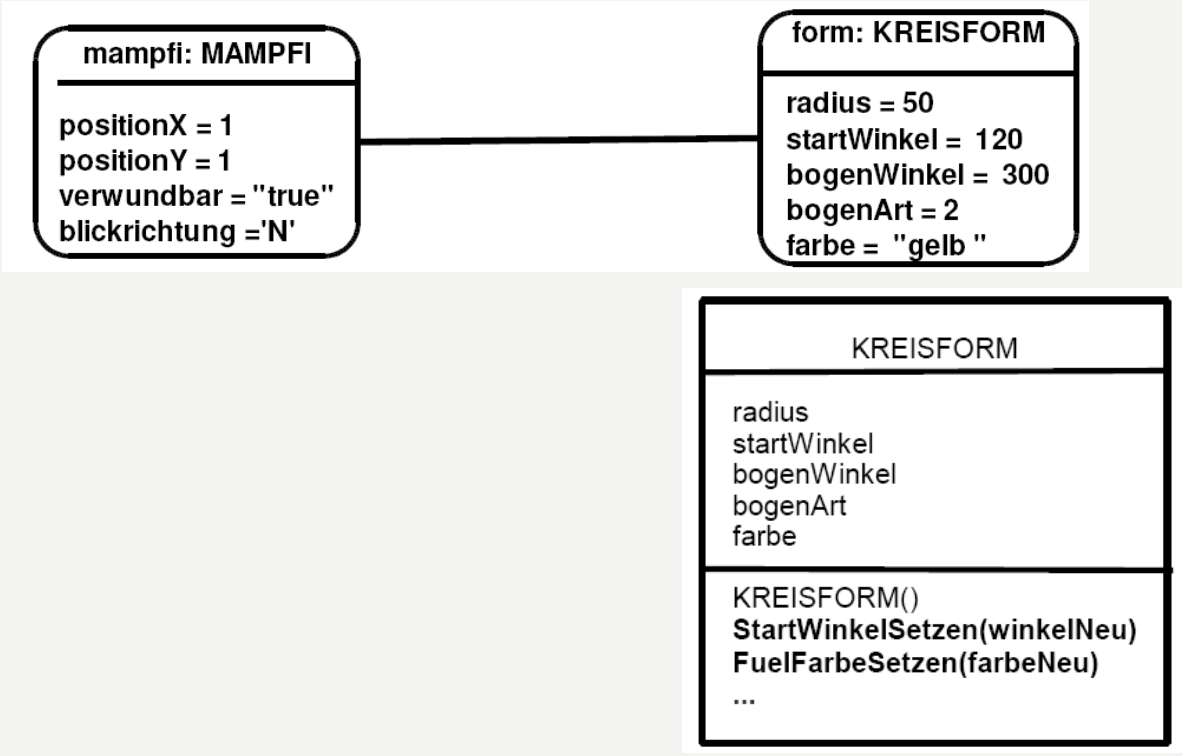
Nur möglich, wenn

a) das Kreisform-Objekt eine Methode nach außen passend zum Änderungswunsch anbietet (Schnittstelle).

b) das Objekt mampfi seinen Änderungswunsch in Form eines Methodenaufrufs an sein Kreisform-Objekt kommuniziert.

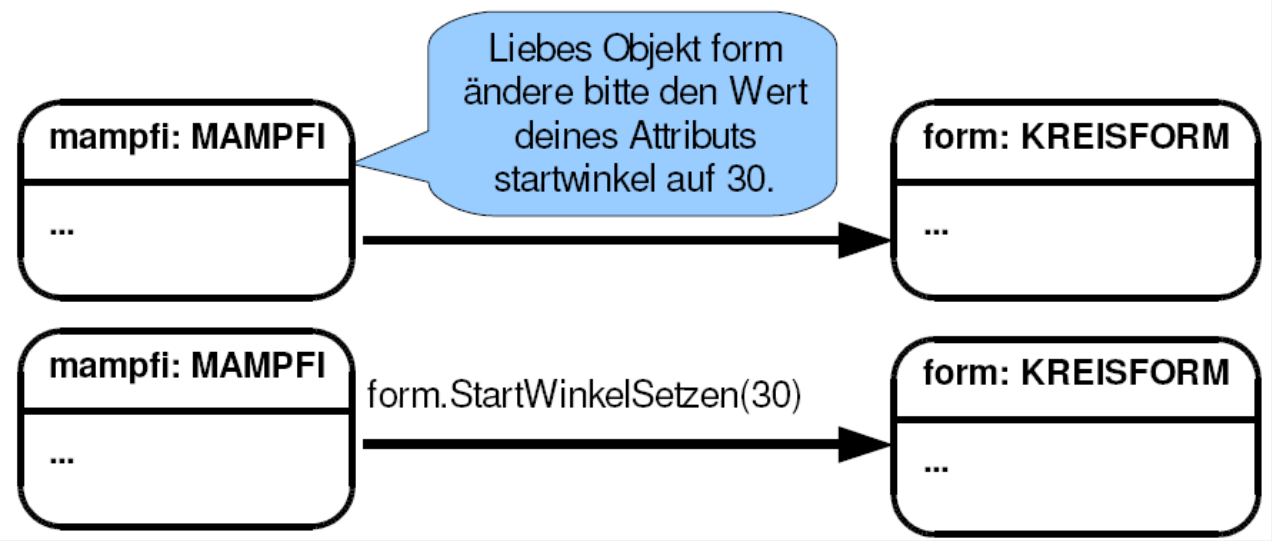
Objektbeziehung - **Objektkommunikation**

a) Methode *StartWinkelSetzen(winkelNeu)* als **Schnittstelle** von Kreisform-Objekten



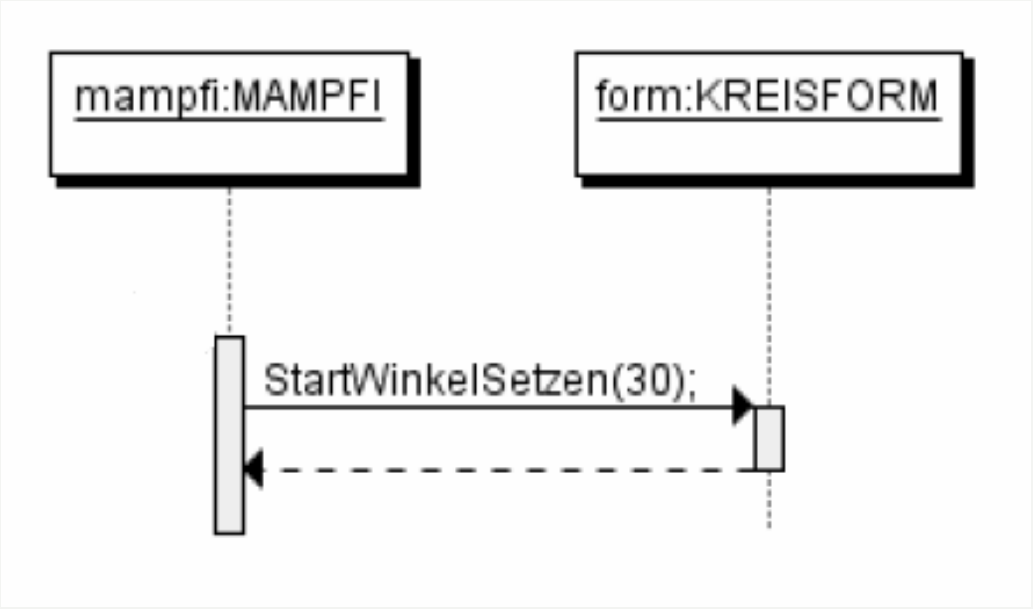
Objektbeziehung - **Objektkommunikation**

b) **Objektkommunikation** von mampfi zum Kreisformobjekt durch einen Methodenaufruf



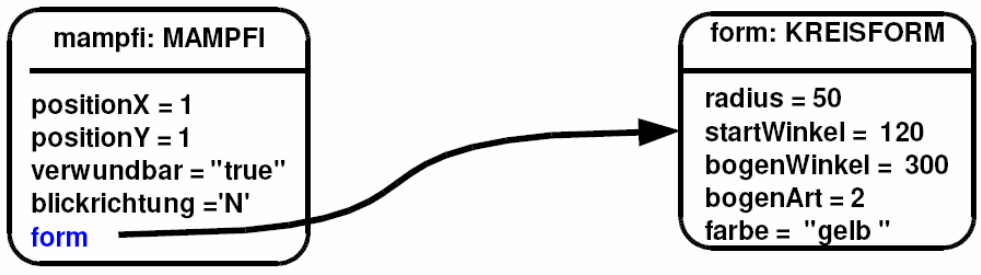
Objektbeziehung - Objektkommunikation

b) **Objektkommunikation** von mampfi zum Kreisformobjekt durch einen Methodenaufruf



Objektbeziehung - Objektkommunikation

c) Voraussetzung für Objektkommunikation: Sendeobjekt mampfi kennt das Empfangsobjekt form



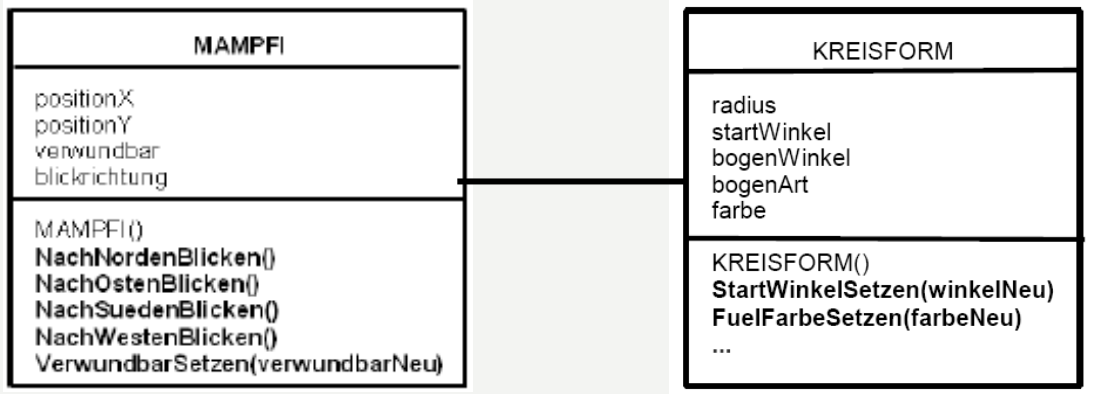
Ähnlich einem Adressbucheintrag muss das Sendeobjekt einen Verweis / eine Referenz auf das Empfangsobjekt haben.

→ Referenzattribut

Objektbeziehung - Objektkommunikation

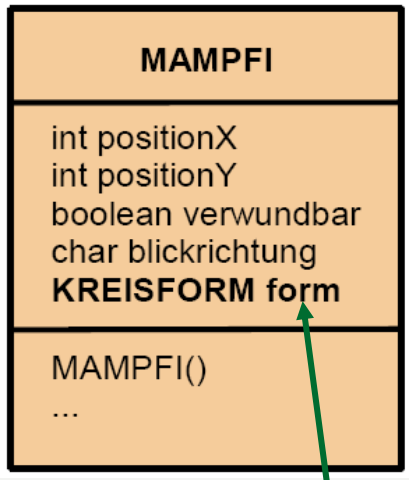
modellieren

Objektorientiertes Modell



umsetzen

Erweitertes Klassendiagramm



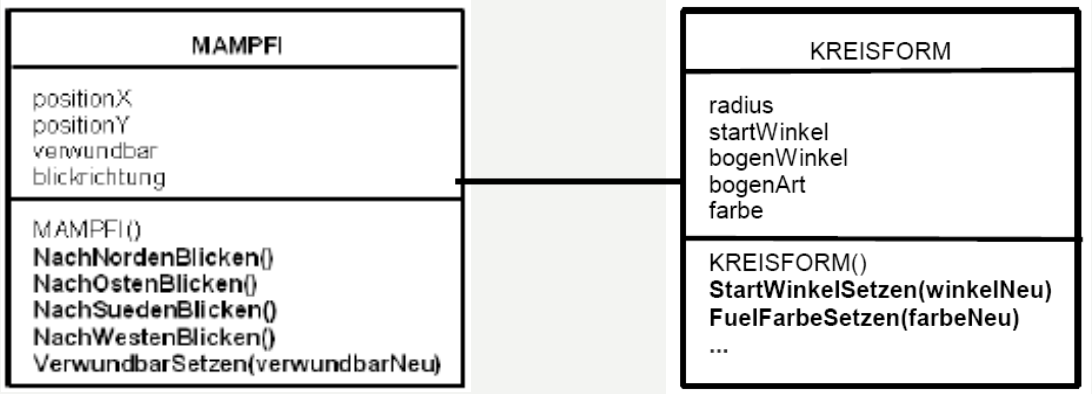
implementieren

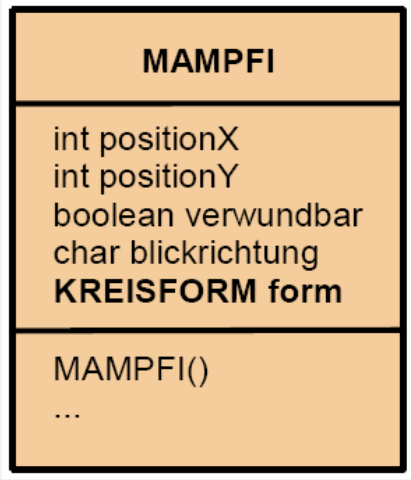
(Java-)Programm

```
class MAMPFI
{
// Attribute
    int positionX;
    int positionY;
    boolean verwundbar;
    char blickrichtung;
// Referenzattribute
    KREISFORM form

// Konstruktor
   MAMPFI()
   { … }
// Methoden
```

Referenzattribut

Objektbeziehung - Objektkommunikation





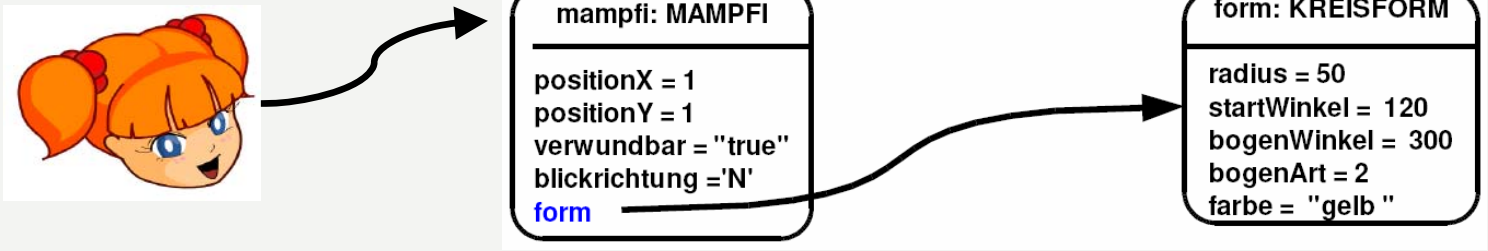

```
class MAMPFI
{
// Attribute
    int positionX;
    int positionY;
    boolean verwundbar;
    char blickrichtung;
// Referenzattribute
    KREISFORM form

// Konstruktor
   MAMPFI()
   { … }
// Methoden
```

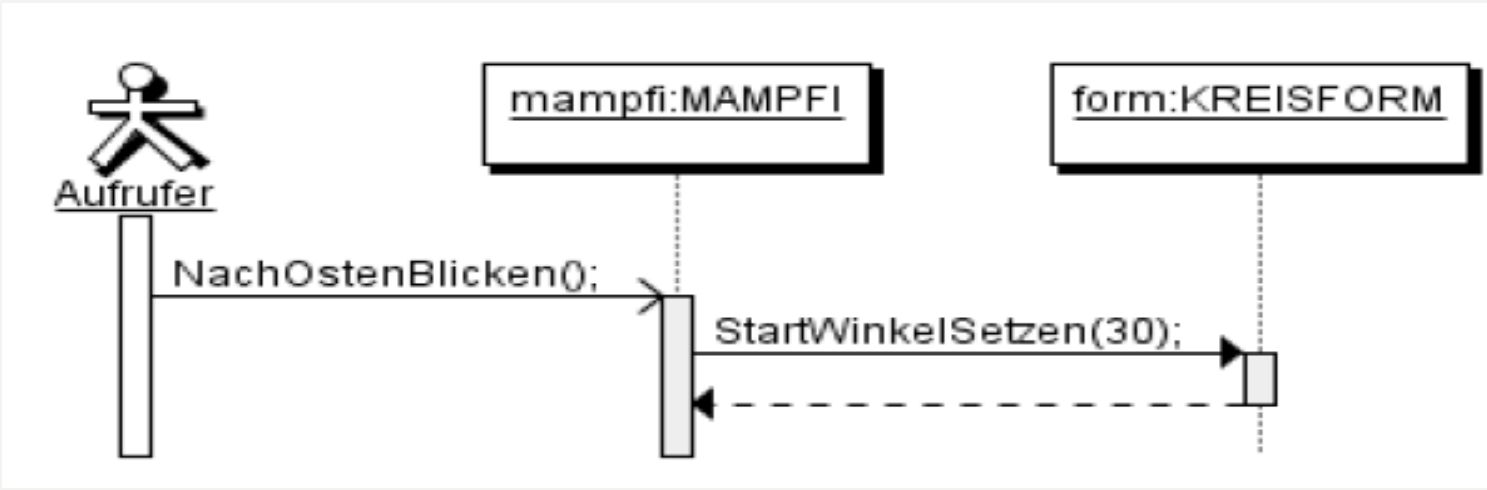
Referenzattribut

´Hinweise:

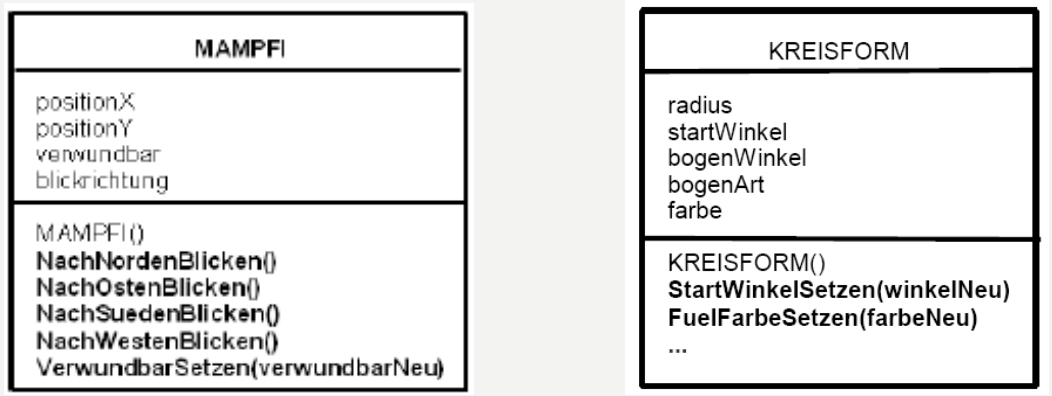
- I.A. keine Richtungsvorgabe der Beziehung im Klassendiagramm
- Implementierung des Referenzattributs in der Klasse (hier MAMPFI), deren Objekte eine Dienstleistung der Objekte der anderen Klasse benötigen.
- Im Beispiel links kennt das Objekt der Klasse KREISFORM nicht das Mampfi-Objekt, das seine Dienste in Anspruch nimmt.
- Bei bidirektionalen Beziehungen muss in beiden Klassen ein Referenzattribut implementiert werden.
- Häufige Fehlvorstellung: Jedes Objekt der Klasse MAMPFI enthält ein Objekt der Klasse KREISFORM

Rollenspiel: Objektbeziehung - Objektkommunikation

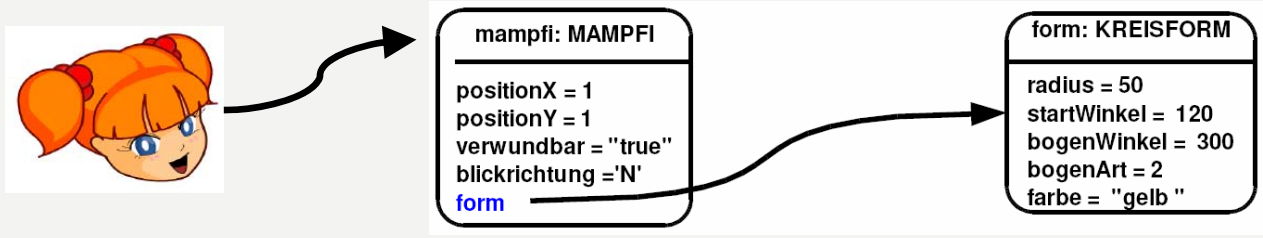




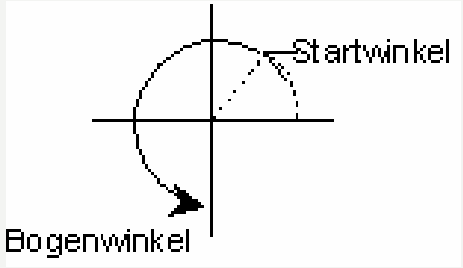
Rollenspiel
Referenzattribute

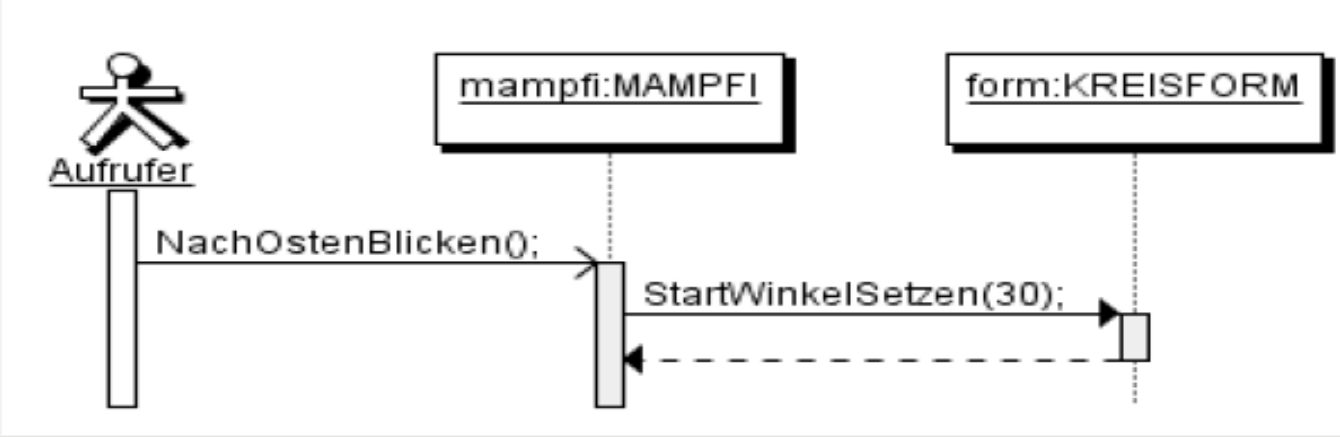


Schnittstellen



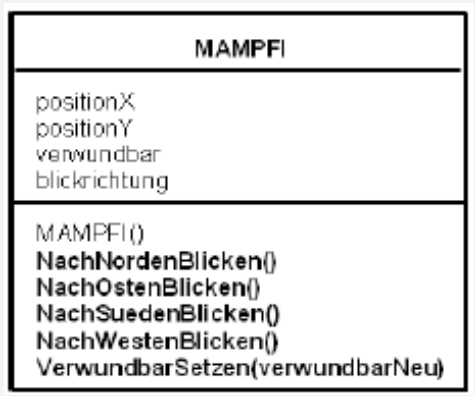
Objekt-beziehung



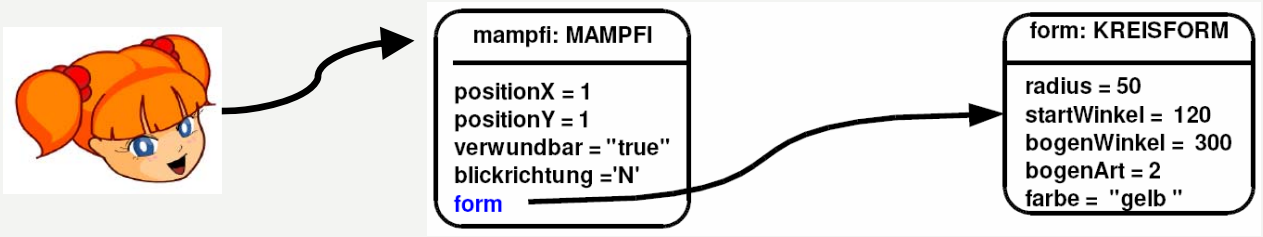


Objekt-kommunikation

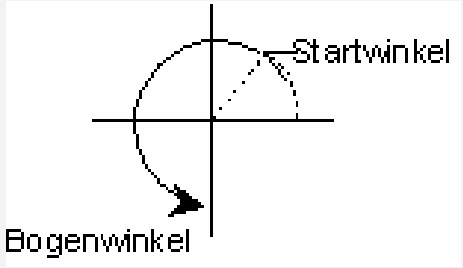
Rollenspiel Referenzattribute



Schnittstellen



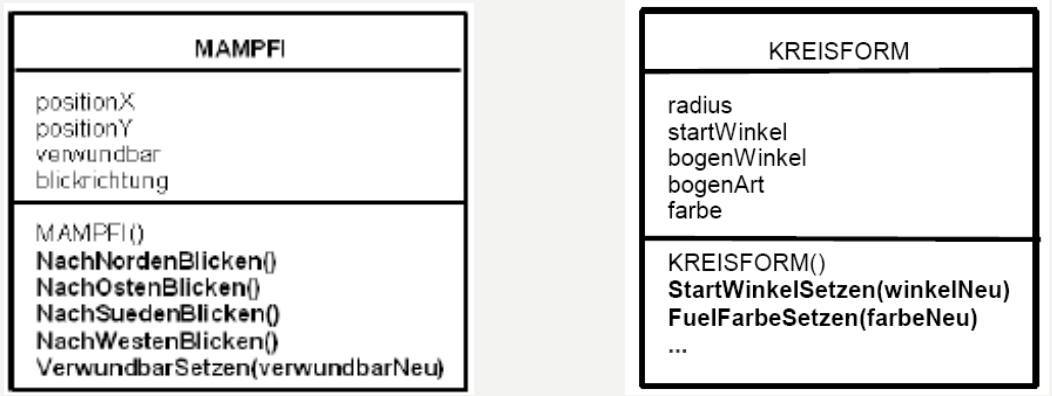
Objekt-beziehung



mampfi.VerwundbarSetzen(false)

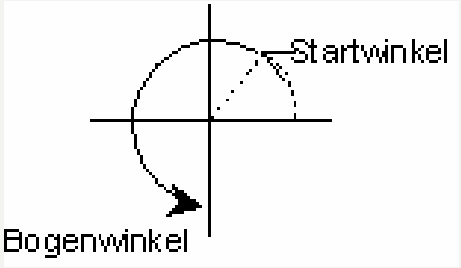
Objekt-kommunikation

Rollenspiel
Referenzattribute



Schnittstellen

Objekt-beziehung



Objekt-kommunikation

mampfi.NachWestenBlicken()

Einführung für Anfänger:

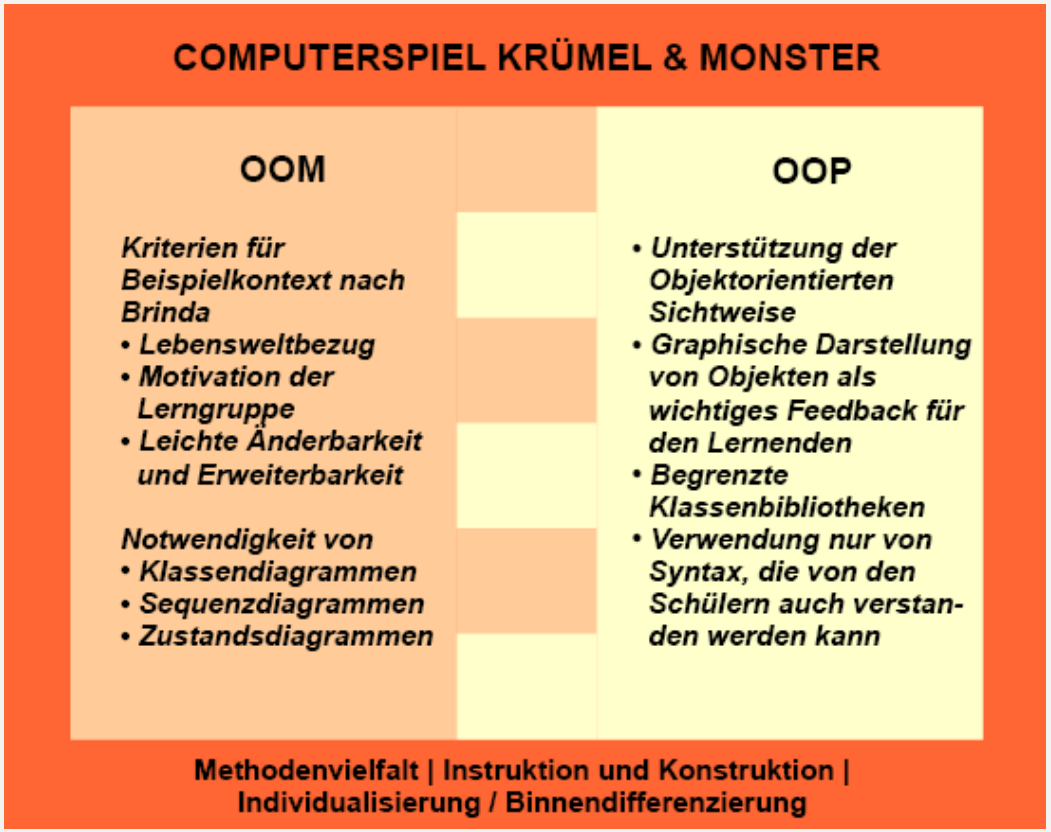
- *Grundwissen OOM (Bayern Jgst. 6, 7 und 9)*
- *Keine Vorkenntnisse OOP*
- *Verwendung nur von Syntax, die bereits behandelt wurde*
- *Graphische Darstellung von Objekten als wichtiges Feedback für den Lernenden*

→ *BlueJ: Entwicklungsumgebung für Java die die objektorientierte Sichtweise unterstützt*

→ *Backend zur Unterstützung der graphischen Darstellung und zum Verbergen unbekannter Syntax*

→ *Backend zur Begrenzung der Klassenbibliotheken, jedoch mit der Offenheit für individuelle Ausgestaltung durch die Lernenden*

Unterrichtskonzept
Anforderungen



Kriterien an das Unterrichtskonzept

- *Themenzentrierter Ansatz*
- *Einführung für Anfänger*
- *OOM und OOP*
- ***Methodenvielfalt bzw. – offenheit***
- ***Aufgabentypen***
- ***Kummulatives Lernen***
- ***Individualisierung / Binnendifferenzierung***

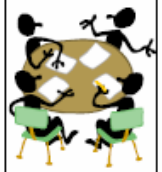
Methodenvielfalt bzw. – offenheit :

- ***Methodenvielfalt:***
  - ***Instruktionen (über neue Elemente der OOM / OOP)***
  - ***Konstruktion (OOM / OOP – Feedback durch die Entwicklungsumgebung)***
  - ***Rollenspiel***
  - ***Selbstständige Zusammenfassungen / Lerntagebuch***
  - ***Aufbau eines Glossars***
- ***Unterrichtsform:***
  - ***Rein Schülerzentriert über Lernpfad (Leitprogramm)***
  - ***Lehrerzentrierte Instruktion / schülerzentrierte Umsetzung als Programm***
  - ***Group Extreme Programming***

Unterrichtskonzept didaktische Aspekte

Aufgaben-typen:

Alle Aufgaben sind mit Icons versehen. Die Bedeutung wird im Folgenden erklärt:

	Fragen, die direkt den Inhalt weiterführen. Sie sind zum Innehalten gedacht, zur Kontrolle, ob man selbst wüsste, wie es weitergeht. Oft steht die Antwort im anschließenden Text.			Aufgaben, deren Bearbeitung direkt am Rechner stattfinden soll. Nutze dabei immer das Feedback, dass dir das Programm mit dem du arbeitest gibt.
	Aufgaben, bei denen es sinnvoll ist, schriftlich einen Eintrag ins Heft zu machen. Dazu gehören beispielsweise Klassendiagramme, die wichtig für die weitere Planung sind.			Aufgaben, bei denen es nötig ist mit dem Banknachbarn bzw. innerhalb einer Kleingruppe zu diskutieren und zusammen ein Ergebnis zu erarbeiten.
	Am Ende jeden Kapitels ist es Aufgabe, das Wesentliche zusammenzufassen. Nimm diese Aufgabe ernst, du wirst viele neue Begriffe und Techniken kennen lernen. Eine Zusammenfassung hilft dir neues Wissen im Gedächtnis zu verankern, Zusammenhänge zu schaffen und bietet die Möglichkeit später nachzuschlagen.			Aufgaben, die nicht unbedingt für das weitere Verständnis bearbeitet werden müssen. Sie sind für diejenigen gedacht, die etwas schneller sind und noch Kraft für einen Spezialauftrag haben.

Kumulatives Lernen:

9.1 Die Klasse KRUEMEL

Aufgabe 9.1
Entwirf eine Klasse KRUEMEL als Klassendiagramm auf Papier mit Bleistift. Welche weitere Klasse ist zur Darstellung von Krümeln nötig? Wie äußert sich dies im (erweiterten) Klassendiagramm?

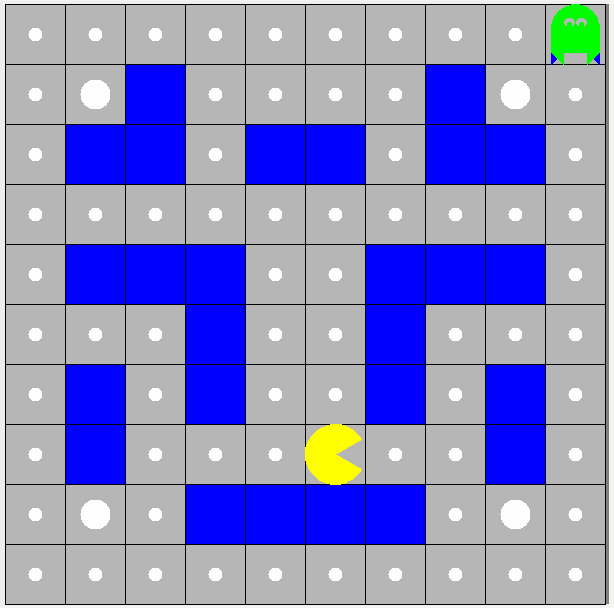
Aufgabe 9.2
Setze den Entwurf aus Aufgabe 9.1. in Java um. Teste nach jedem Teilschritt.

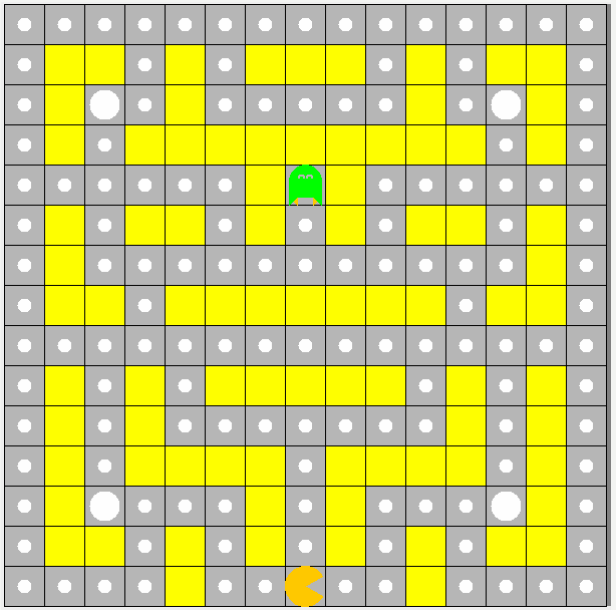
Aufgabe 9.3
Fülle das Labyrinth mit Krümeln. In diesem ersten Entwurf sollen Mauern noch nicht vorkommen. Es ist im ersten Entwurf auch noch einfacher sich auf einer Sorte von Krümeln zu beschränken.

Hinweise:
Aus Sicht des bisher erlernten kannst du diese Aufgaben völlig alleine lösen. Vergiss nicht nach Änderungen ausführlich zu testen, ob dein Ziel erreicht wurde.
Solltest du dich unsicher fühlen, findest du im Folgenden verschiedene Fragen und Tipps, die dir helfen sollen, die nächsten Schritte zu finden.

Individualisierung / Binnendifferenzierung :

- ***Individuelle Ausprägung***
  - ***Spielgestaltung***
  - ***Spielablauf, z.B. nach dem Fressen eines Monsters***
  - ***Spielstrategie, z.B. „Intelligenz“ der Monster***





Individualisierung / Binnendifferenzierung :

- ***Individuelle Ausprägung***
- ***Aufgaben vom Typ „Rampe“***

Aufgabe 14.16
Das Spielende lässt sich noch deutlicher darstellen. Folgende Möglichkeiten gibt es:
a) Mampfi macht den Mund auf und zu und bei den Monstern bewegen sich die Füße.
Das Ende dieser Bewegungen würde zum Spielende gut passen.
b) Mampfi könnte verschwinden, indem der Öffnungswinkel des Kreissektors mit dem Mampfi dargestellt wird Grad für Grad auf 0 reduziert wird und somit das Symbol von Mampfi verschwunden ist.
Versuche diese Ideen umzusetzen oder entwickle eigene.

Individualisierung / Binnendifferenzierung :

- *Individuelle Ausprägung*
- *Aufgaben vom Typ „Rampe“*
- *Abgestuftes Hilfesystem*

noch mehr Tipps zu Aufgabe 14.12

Rumpf der Methode PositionSetzen(xNeu, yNeu)

(xNeu >= 0) && (xNeu <= labyrinth.BreiteGeben()-1) &&
(yNeu >= 0) && (yNeu <= labyrinth.HoeheGeben()-1)

true | false

```
positionX = xNeu
positionY = yNeu
```

```
FormAktualisieren();
```

Individualisierung / Binnendifferenzierung :

- ***Individuelle Ausprägung***
- ***Aufgaben vom Typ „Rampe“***
- ***Abgestuftes Hilfesystem***
- ***Lerntagebuch (Journal)***
- ***Moderation von Foren***

Zusammenfassung

Unterrichtskonzept zum Thema OOM und OOP

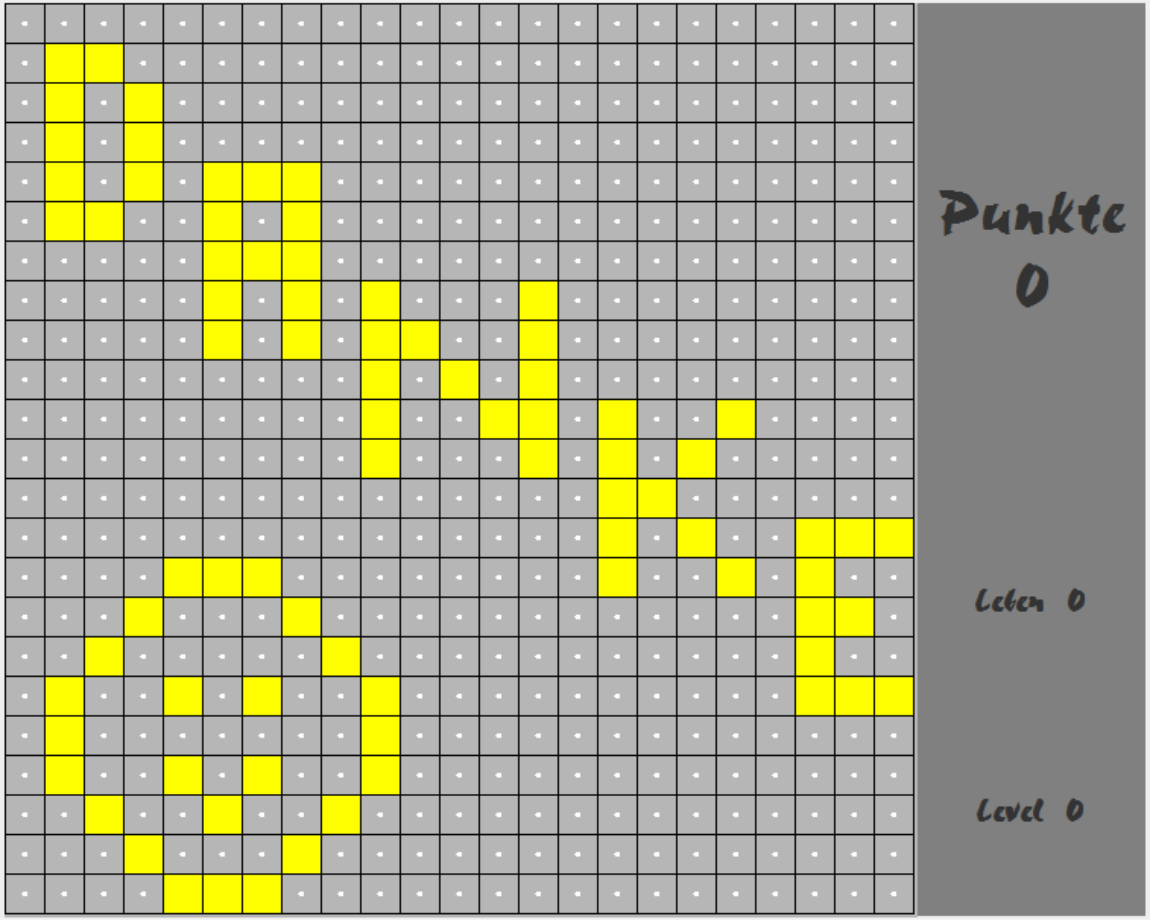
- Umfang von ca. 7 Monaten
- Themenzentrierte Abdeckung aller Fachinhalte
- Unterstützung durch eine didaktische Entwicklungsumgebung
- Methodenvielfalt | Instruktion und Konstruktion | Individualisierung und Binnendifferenzierung
- Neuer Ansatz für diese Thematik

Offene Punkte

- Erhöhung der Lösungsvielfalt
- Optimierung des Backends
- Aufgabenmaterial außerhalb des themenzentrierten Ansatzes orientiert an den Bildungsstandards

→Fertigstellung voraussichtlich bis Schuljahr 09/10

Zusammenfassung



www.brichzin.de

brichzin@lmu.de