

## Kapitel 3 Die erste Klasse selbst erstellen: MAMPFI

Lernziele:

Eine Klasse in Java erstellen:

Attribute, Datentypen, Konstruktor, Referenzattribute, Kommentare, Zuweisung  
 Modellierung: Objektkommunikation

### 3.1 Die Klasse MAMPFI

In Kapitel 1 hast du das Spiel Pacman analysiert, ein Klassendiagramm mit vielen verschiedenen Klassen und deren Beziehungen erstellt. Da eine individuelle Note nicht schadet, soll das von uns erstellte Spiel **Krümel & Monster** heißen. Entsprechend gibt es in dem Spiel Objekte der Klasse KRÜMEL, die von einem Objekt der Klasse MAMPFI gefressen werden, das wiederum auf dem Speiseplan von Objekten der Klasse MONSTER steht.

Begonnen wird mit der Klasse MAMPFI. Entsprechend Vorschlägen für Attribute aus Kapitel 1 ergibt sich das Klassendiagramm in der Abbildung 1.

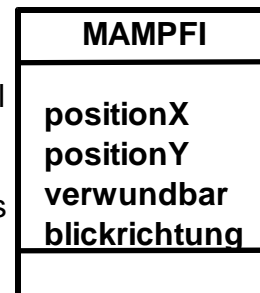


Abbildung 1:  
 Klassendiagramm der Klasse MAMPFI (nur Attribute)

Das Klassendiagramm ist eine Art Plan, das hinsichtlich des Ziels, die Klasse in Java umzusetzen hilft. Die folgende Tabelle zeigt die Struktur von Klassen in der Programmiersprache Java. Die Struktur ist genau die gleiche wie im Klassendiagramm.

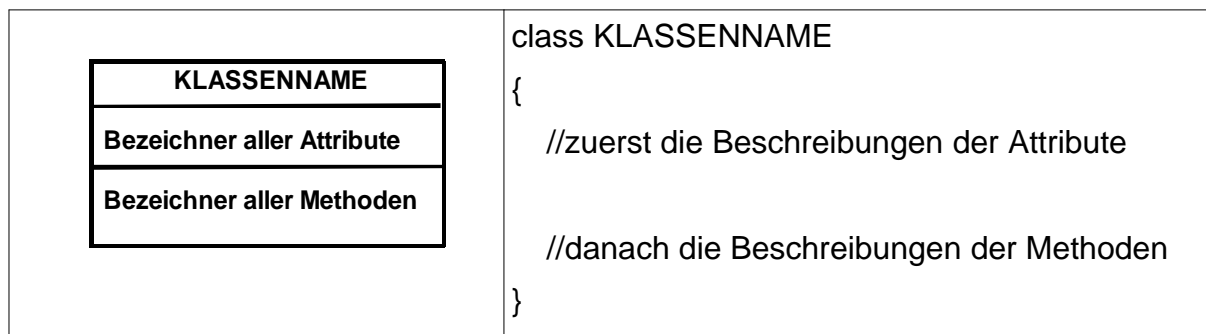


Abbildung 2: Allgemeiner Aufbau einer Klasse im Vergleich zwischen Klassendiagramm und Java-Quelltext. Zuerst muss die Klasse mit einem Namen benannt werden, dann folgen die Attribute und Methoden.

Zu Beginn des Java-Quelltextes einer Klasse steht immer das Wort `class`. Danach folgt der Name der Klasse. Die eigentliche Beschreibung der Klasse wird von geschweiften Klammern eingeschlossen. Dies entspricht dem Rechteck beim Klassendiagramm (wobei der Klassenname bei Java im Gegensatz zum Klassendiagramm außerhalb der Klammern ist).

Wörter wie `class`, die in einer Programmiersprache eine feste Bedeutung haben, nennt man **Schlüsselwörter**.

#### Hinweise:

- Bei Java und auch bei vielen anderen Programmiersprachen muss die Groß- / Kleinschreibung beachtet werden. So wären Klasse1, KLASSE1 und klasse1 drei unterschiedliche Bezeichner, die ohne Fehlermeldungen parallel existieren

können. Die Namenskonvention in dieser Lernsequenz ist, dass Klassennamen nur aus Großbuchstaben bestehen und Attribute nur aus Kleinbuchstaben.

- Quelltexte sind besser lesbar, wenn der Text innerhalb der geschweiften Klammern deutlich eingerückt wird. Vier Leerzeichen sind dafür empfehlenswert. BlueJ macht dies automatisch.

Im Klassendiagramm in Abbildung 1 sind nur die Bezeichner der Attribute notiert. Als zweiten Schritt der Planung muss man sich überlegen, welcher Datentyp für die einzelnen Attribute sinnvoll ist<sup>1</sup>.



### Aufgabe 3.1

Nenne sinnvolle Datentypen für die Attribute der Klasse MAMPFI.

Da man mit ganzen Zahlen einen Ort in einem Gitter festlegen kann, ist `int`<sup>2</sup> für die Positionen ein geeigneter Datentyp. Weil das Attribut *verwundbar* nur zwei verschiedene Werte (`true` oder `false`) annehmen kann, ist `boolean`,<sup>3</sup> ein geeigneter Datentyp. Die Blickrichtung kann über die Himmelsrichtungen 'N', 'W', 'S' und 'O' klar beschrieben werden. Aus diesem Grund wird der Datentyp `char`<sup>4</sup> gewählt. Abbildung 3 zeigt ein um die Datentypen **erweitertes Klassendiagramm**. Der Vergleich mit dem Quelltext zeigt, dass die Struktur identisch ist.

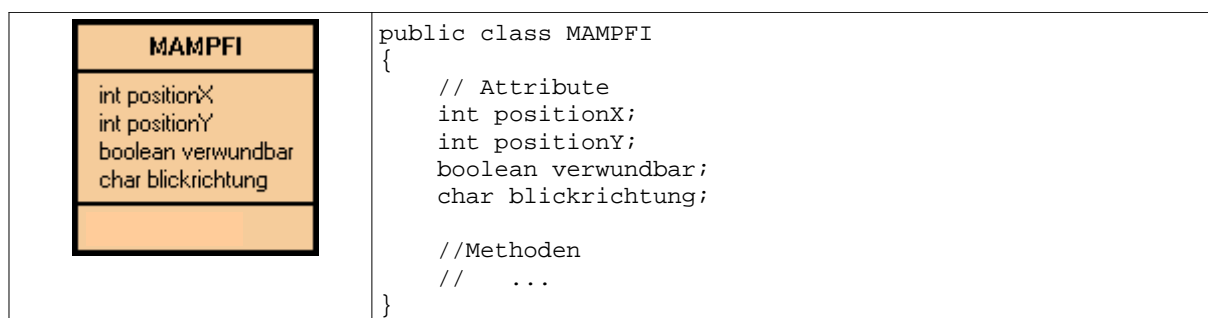


Abbildung 3: Ein um die Datentypen erweitertes Klassendiagramm der Klasse MAMPFI im Vergleich mit dem Java Quelltext

Hinweise:

- Die Festlegung von Datentyp und Bezeichner eines Attributs nennt man **Attributdeklaration**.
- Erläuterungen im Quelltext, die nur für außen stehende Leser, nicht für den Computer bestimmt sind, nennt man **Kommentare**. In Java können diese Kommentare mit zwei Schrägstrichen `//` beginnen und gehen dann bis zum Ende der Zeile. Größere Kommentarbereiche oder Einschübe innerhalb einer Zeile beginnen mit `/*` und enden mit `*/`.

In Abbildung 3 rechts werden sie beispielsweise verwendet, um deutlich zu machen wo die Attribute bzw. Methoden notiert werden.



<sup>1</sup> Erinnere dich an das Erstellen einer Datenbanktabelle. Auch dort musste der Datentyp für die Werte jeder Spalte beim Anlegen einer Tabelle festgelegt werden.

<sup>2</sup> engl.: integer – ganze Zahl

<sup>3</sup> engl.: boolean – Wahrheitswert

<sup>4</sup> engl.: character – Zeichen

## Aufgabe 3.2

a) Schreibe eine Klasse MAMPFI entsprechend dem Quelltext in der Abbildung 3.

Gehe dazu wie folgt vor:

- Klicke die Schaltfläche „Neue Klasse ...“ und gib dort den Klassennamen ein (Abb. 4).
- Durch das bestätigen mit OK erscheint eine Klassensymbol. Öffne das Editor Fenster für diese Klasse durch einen Doppelklick auf das Klassensymbol.
- Lösche in dem sich öffnenden Editor-Fenster den gesamten Text und trage den Quelltext entsprechend der Planung in Abbildung 3 ein.

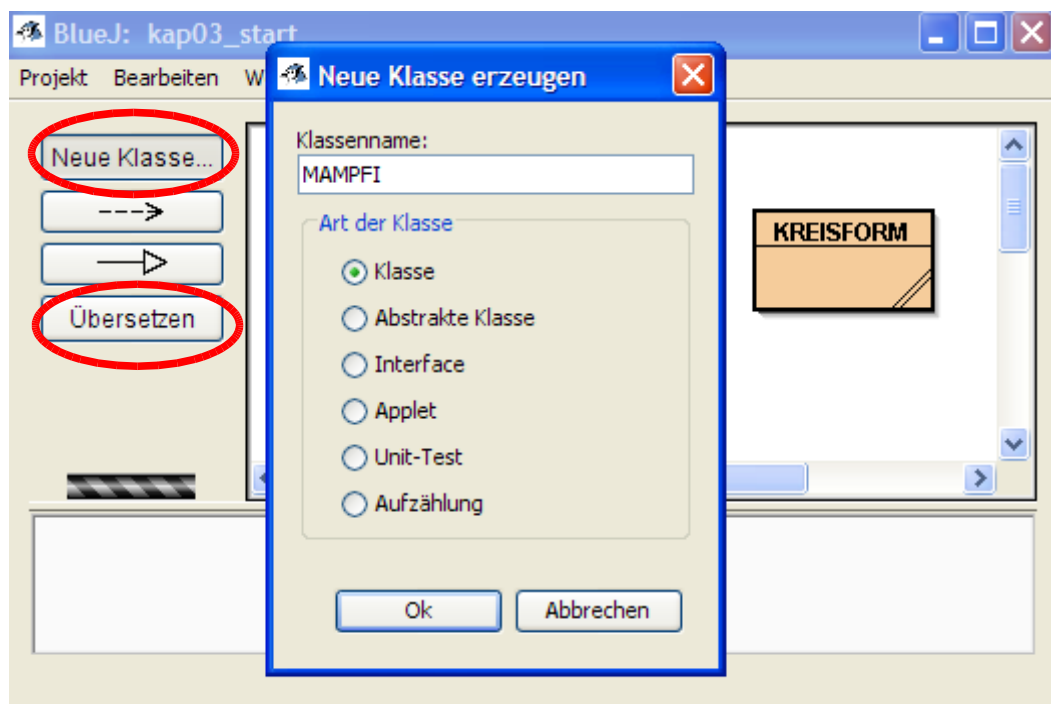


Abbildung 4:

Erzeugen einer neuen Klasse mit BlueJ über die Schaltfläche „Neue Klasse ...“  
Übersetzen dieser Klasse über die Schaltfläche „Übersetzen“



Compiler

- b) Übersetze<sup>5</sup> diese Klasse (Abbildung 4) und erzeuge ein Objekt.  
c) Betrachte die Attributwerte im Objektinspektor. Was ist nicht in Ordnung?

<sup>5</sup> engl. Compiler: Übersetzer

### 3.2 Konstruktor

Betrachtet man ein Objekt der Klasse MAMPFI im Objektinspektor (Abb. 5) sind die Positionsangaben noch nicht aussagekräftig, da noch kein Labyrinth (Spielfeld) vorhanden ist. Jedoch fällt auf, dass das Attribut `blickrichtung` einen seltsamen Wert hat, der sicher keine Himmelsrichtung darstellt. Weiterhin ist der Wert des Attributs `verwundbar` `false`, obwohl das Objekt `mampfi` sicher zu Beginn des Spieles verwundbar ist. Das System hat dem Objekt Werte zugewiesen, die nicht passen. Es ist nötig dem Objekt gleich nach dem Erzeugen passende bzw. gültige Attributwerte zuzuweisen. Dies geschieht durch eine besondere Methode die **Konstruktor** genannt wird. Der Konstruktor wird nach dem Erzeugen eines Objekts sofort vom System automatisch aufgerufen. In Java hat ein Konstruktor die Form:



Abbildung 5: Attributwerte von `mampfi` im Objektinspektor – der Wert `false` des Attributs `verwundbar` passt nicht zum Startzustand von `mampfi`, der Wert des Attributs `blickrichtung` ist unsinnig

```
KLASSENNAME()
{
    // Anweisungen
}
```

Um nun `mampfi` verwundbar starten zu lassen und ihm die Blickrichtung Norden zuzuordnen muss man in der Klasse MAMPFI nach den Attributen den Konstruktor in folgender Art ergänzen:

```
public class MAMPFI
{
    // Attribute
    // ...

    // Konstruktor für Objekte der Klasse MAMPFI
    MAMPFI()
    {
        //Die Position kann mangels Labyrinth noch nicht festgelegt werden
        verwundbar = true;
        blickrichtung = 'N';
    }
}
```

Abbildung 6: Java Quelltext des Konstruktors MAMPFI (fett markiert) eingebettet in die Klasse MAMPFI



#### Über Anweisungen der Form

```
attributname = WertNeu;
```

Variablen-  
konzept

wird einem Attribut ein Wert zugewiesen. Anweisungen dieser Art heißen Zuweisungen. So erhält beispielsweise im Konstruktor in Abbildung 6 das Attribut `verwundbar` den Wert `true` und `blickrichtung` den Wert `'N'`.

#### Hinweise:

- Die Anweisung `verwundbar = true;`

wird wie folgt gesprochen: „Dem Attribut verwundbar wird der Wert `true` zugewiesen.“

Diese Formulierung macht deutlich, dass eine Zuweisung keinen Zustand beschreibt, sondern eine Aktion. Durch eine Zuweisung wird der Wert eines Attributs geändert.

- Die Position des Konstruktors ist immer nach der Attributdeklaration.



### Aufgabe 3.3

Notiere in deinem Heft, wie man folgende Zuweisung als Text spricht:

```
blickrichtung = 'N';
```



### Aufgabe 3.4

Ergänze den Konstruktor in der Klasse MAMPFI. Übersetze die Klasse und erzeuge ein Objekt. Betrachte die Attributwerte im Objektinspektor.

## 3.3 Objektbeziehungen

Nun hast du zwar mit der Entwicklungsumgebung ein Objekt der Klasse MAMPFI erzeugt, „aber“ es ist nur im Objektinspektor sichtbar. Ziel ist es natürlich wie beim Pacman ein „Symbol“ zu sehen.



### Aufgabe 3.5

Welche andere Klasse könnte bei der Erfüllung dieses Ziels helfen? In welchem Bezug sollte diese andere Klasse zur Klasse MAMPFI stehen? Beschreibe möglichst genau.



Im Kapitel 2 wurde durch das Erzeugen eines Objekts der Klasse KREISFORM eine Darstellung von Pacman, ein Pacmansymbol erstellt. Man kann so ein Objekt nutzen, um ein MAMPFI-Objekt darzustellen.

Abbildung 7 zeigt die Objektdiagramme zwei miteinander in Beziehung stehenden Objekte: Das Objekt `mampfi` der Klasse MAMPFI (mit dem Wert 'N' des Attributs `blickrichtung`) wird dargestellt durch ein Objekt `symbol` der Klasse KREISFORM (mit dem Wert 120 des Attributs `Startwinkel`).

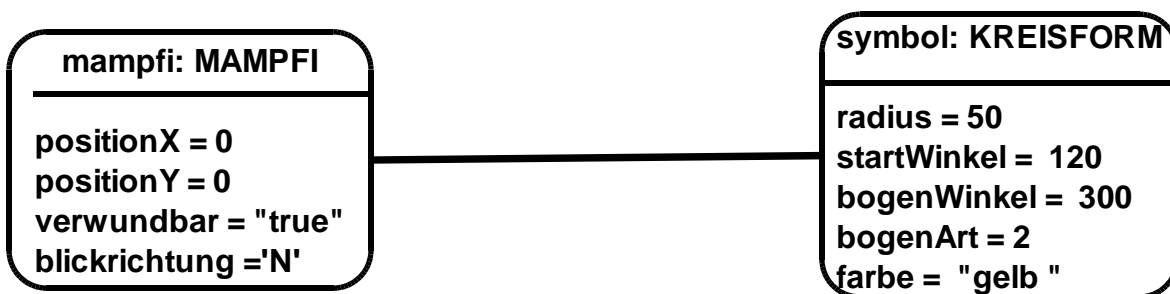


Abbildung 7: Beziehung zwischen den Objekten `mampfi` und `symbol` – Passend zur Blickrichtung Norden von `mampfi` ist der Wert des Attributs `Startwinkel` der Kreisform 120

Abbildung 8 zeigt die Beziehung im Klassendiagramm.



Abbildung 8: Klassendiagramm mit Beziehungen: „Ein Objekt der Klasse MAMPFI wird dargestellt durch genau ein Objekt der Klasse KREISFORM“ bzw. „Ein Objekt der Klasse KREISFORM gehört zu genau einem Objekt der Klasse MAMPFI.“

Dabei nimmt ein Objekt der Klasse MAMPFI Dienste eines Objekts der Klasse KREISFORM in Anspruch. Je nachdem in welche Richtung mampfi schaut, muss das zugehörige Objekt der Klasse KREISFORM, hier symbol genannt, dazu passen.

Ändert mampfi seine Blickrichtung, so ist es wichtig für eine korrekte Darstellung, dass der Wert des Attributs startwinkel des zugehörigen Symbols angepasst wird. Abbildung 9 zeigt eine passende Kombination der Attributwerte der Objekte mampfi und symbol nachdem im Vergleich zur Abbildung 7 die Blickrichtung geändert wurde.

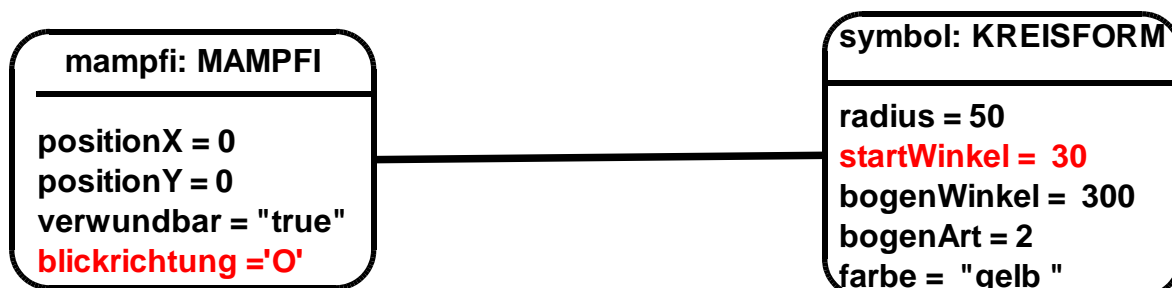


Abbildung 9: Beziehung zwischen den Objekten mampfi und symbol – passend zur Blickrichtung Osten von mampfi ist der Wert des Attributs Startwinkel der Kreisform 30

Die Anpassung der Attributwerte des Symbols funktioniert jedoch nicht automatisch, denn wenn das Objekt mampfi seine Blickrichtung ändert weiß das zugehörige Symbol davon zunächst nichts.



### Aufgabe 3.6

Wie kann das Objekt mampfi dem Objekt symbol mitteilen, eine bestimmte Blickrichtung darzustellen?

Denke beispielsweise an die Unterstufe: Wie hast du einem Rechteck-Objekt mitgeteilt, dass es seine Füllfarbe ändern soll?

Oder wie hast du in Kapitel 2 Objekten der Klasse KREISFORM mitgeteilt die Richtung der Öffnung zu ändern?

## 3.4 Objektkommunikation

In der letzten Aufgabe in Kapitel 2 hast du über Methodenaufrufe bei einer Pacman Darstellung die Blickrichtung durch das Setzen des Startwinkels verändert. Nun kannst nicht nur du eine Botschaft mit einem Methodenaufruf an ein Objekt senden, sondern Objekte können auch untereinander über Methodenaufrufe kommunizieren. Abbildung 10 veranschaulicht so eine Objektkommunikation:

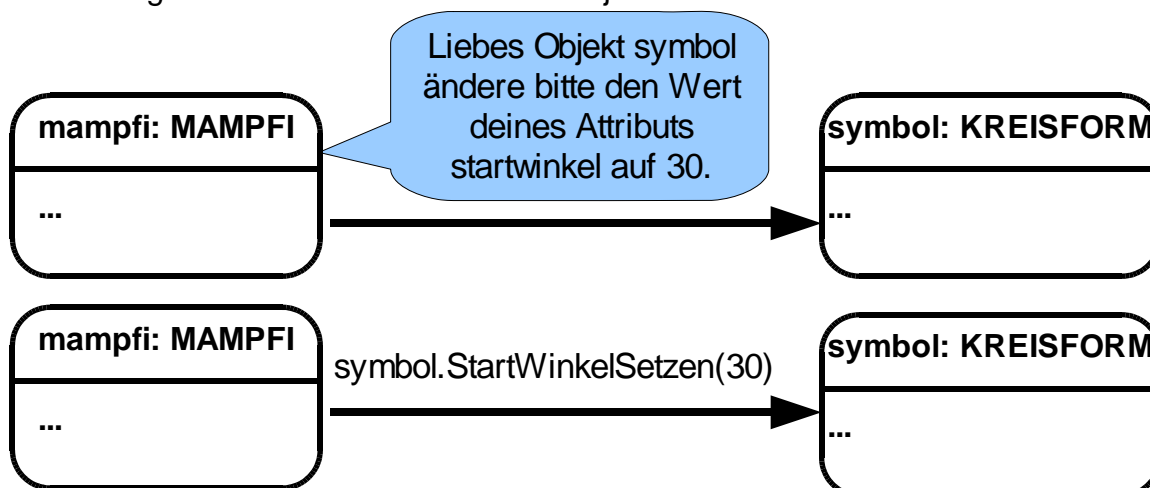


Abbildung 10: Kommunikation zwischen dem Objekten mampfi und symbol – mampfi fordert das Objekt symbol auf, den Wert seines Attributs Startwinkel auf den 30 zu setzen

Im ersten Teil der Abbildung 10 ist die Nachricht in einer „Comix-Sprechblase“ notiert. Diese Notation verstehen Informatiksysteme jedoch nicht. Dort ist die Punktnotation für Methodenaufrufe nötig. Abbildung 11 zeigt die Bestandteile.

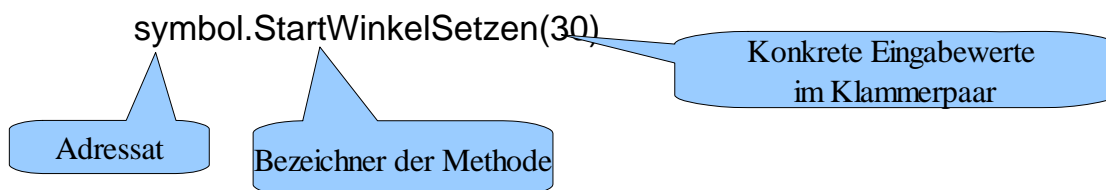


Abbildung 11: Bestandteile eines Methodenaufrufs in Punktnotation: Adressat, Methodenname, Klammerpaar und - abhängig von der Methode- keinen, einen oder mehrere Eingabewerte

**Hinweise:**

- Keine Methode, kein Methodenaufruf ohne Klammerpaar! Die Klammern sind ein perfektes Erkennungszeichen. Sollten keine Eingabewerte nötig sein, so steht innerhalb der Klammern nichts, z.B. der Methodenaufruf `karol.LinksDrehen()`.
- Ein Vergleich zwischen Abbildung 12 aus Kapitel 2 und Abbildung 9 in diesem Kapitel ergibt keinen strukturellen Unterschied. Anstelle der externen Aufruferin sendet nun das Objekt `mampfi` die Botschaft mit dem Methodenaufruf.



**Aufgabe 3.7**

Spielt die Objektkommunikation in Abbildung 9 als Rollenspiel nach. Verwendet dabei als Objektamen Euren richtigen Namen. Fertigt Euch ein Namensschild an, auf dem auch notiert ist, ob ihr ein Objekt der Klasse `MAMPFI` oder `KREISFORM` seid. Der Lehrer gibt Euch weitere Informationen, wie ihr wichtige Attributwerte symbolisieren könnt.

**3.5 Objektbeziehungen in Java realisieren**

Ein Nachrichtenaustausch zwischen Objekten kann nur dann stattfinden, wenn das sendende Objekt die Nachricht dem empfangenden Objekt zustellen kann. Wie bei einer Postzustellung muss das Sendeobjekt das Empfangsobjekt kennen, es muss sich ähnlich einem Adressbucheintrag einen Verweis (Referenz) auf das Empfangsobjekt merken.

Damit das Objekt `mampfi` das Objekt `symbol` ansprechen kann, erhält es ein besonderes Attribut dessen Wert eine Referenz auf das Objekt `symbol` ist. Attribute, die auf andere Objekte referenzieren werden Referenzattribute genannt. Die folgende Abbildung veranschaulicht die Referenz.

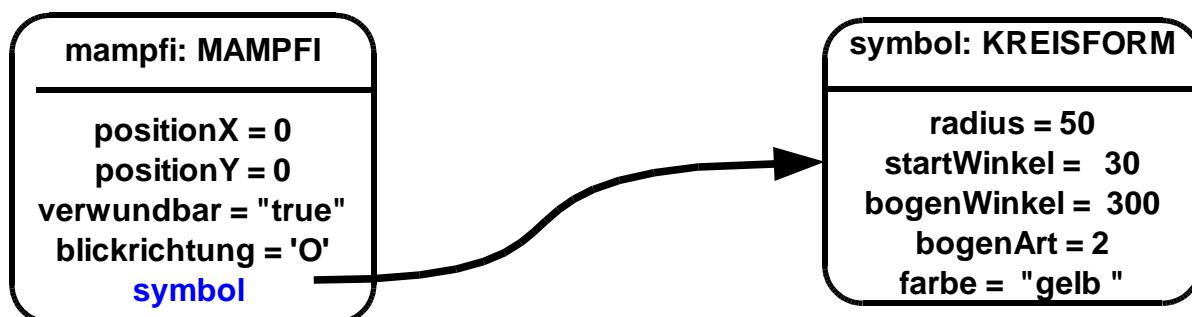


Abbildung 12: Referenz zwischen Objekten – das Referenzattribut `symbol` verweist auf das Objekt `symbol`

Vergleicht man Abbildung 9 mit Abbildung 12, so ist der einzige Unterschied das zusätzliche Attribut symbol beim Objekt mampfi.

Referenzattribute werden im erweiterten Klassendiagramm aufgelistet. Für die Klasse MAMPFI ergibt sich das erweiterte Klassendiagramm in Abbildung 13. Da das Referenzattribut symbol auf ein Objekt der Klasse KREISFORM verweist, wird dieser Klassenname als Datentyp vor dem Referenzattributnamen notiert.

Im Vergleich zu Abbildung 3 wurde neben dem Referenzattribut auch der in Kapitel 3.2 besprochene Konstruktor als erste Methode in Abbildung 13 ergänzt.

Wie schon in Kapitel 3.1 ist eine Übertragung in Java Quelltext nicht mehr schwer, wenn die Planung in Form eines erweiterten Klassendiagramms durchgeführt wurde. Der Java Quelltext der Klasse MAMPFI ist im rechten Teil der Abbildung 13 zu sehen:

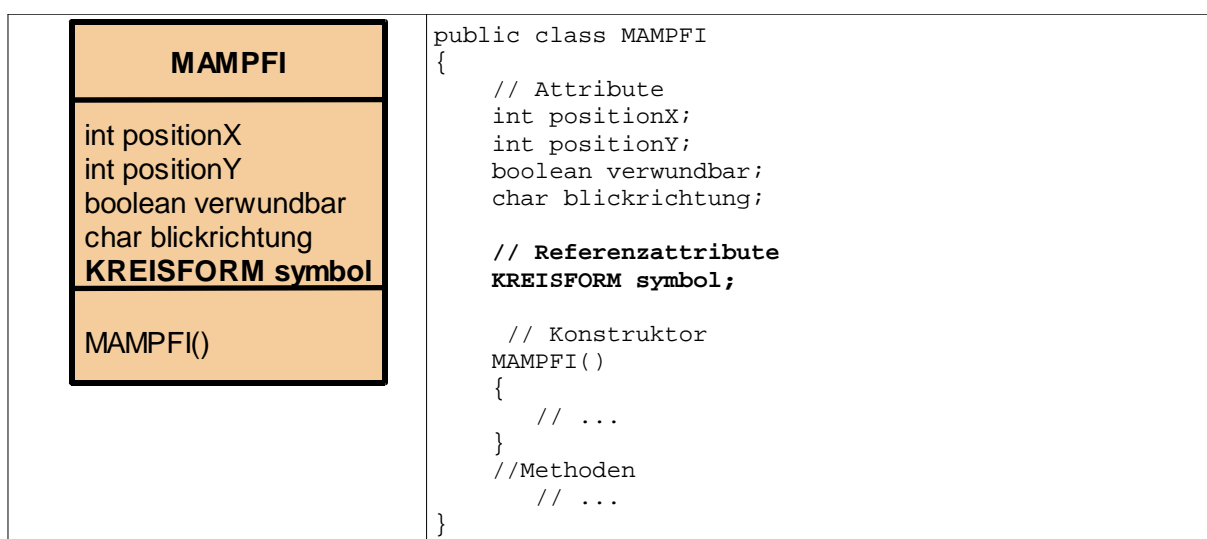


Abbildung 13:

Klassendiagramm und Quelltext der Klasse MAMPFI erweitert durch das Referenzattribut symbol



### Aufgabe 3.8

- a) Ergänze die Klasse MAMPFI um das Referenzattribut. In deinem BlueJ Projekt kannst du deine Klassen jederzeit verändern, indem du mit einem Doppelklick auf das Klassensymbol den Quelltexteditor öffnest.
- b) Übersetze die Klasse und erzeuge ein Objekt. Betrachte die Attributwerte im Objektinspektor. Was fällt auf?

### Hinweise:

- Die Pfeilrichtung in Abbildung 12 macht deutlich, dass nur Mampfi eine Referenz hat. Das Objekt symbol kennt das Objekt mampfi nicht und kann ihm auch keine Nachrichten schicken.



Im Objektinspektor (Abbildung 14) erscheint zwar das (Referenz-)Attribut `symbol`, aber die Angabe `null` ist ein Hinweis darauf, dass es dafür noch keinen Wert gibt. Du kennst den Begriff `null` mit der Bedeutung „kein Wert“ aus der Arbeit mit Datenbanken.



### Aufgabe 3.9

Zähle alle Methodenaufrufe auf, die du in Aufgabe 2.8 durchgeführt hast, um einen nach Norden schauenden Pacman zu zeichnen. Alle sind auch hier nötig.

An welcher Stelle im Java

Quelltext wäre es sinnvoll die eben wiederholten Methodenaufrufe zu notieren?

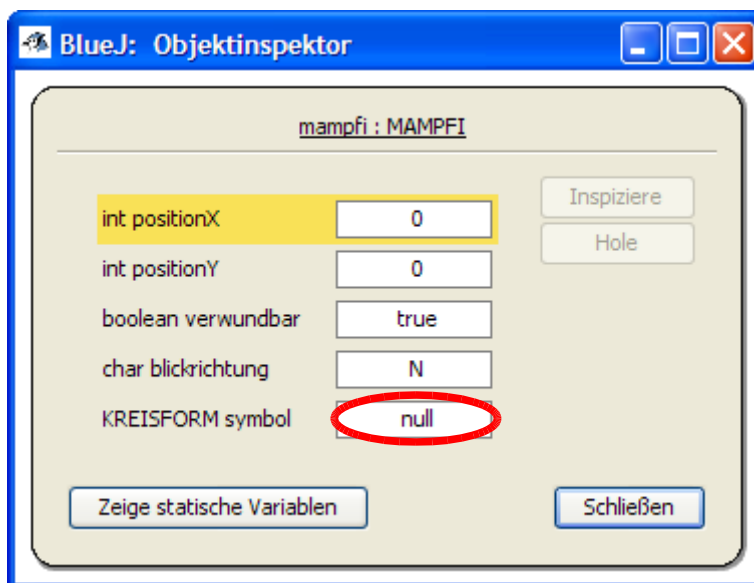


Abbildung 14: Die Angabe `null` zeigt, dass der `form` noch kein Wert, keine Referenz zugewiesen wurde

Der erste Schritt in Aufgabe 2.8 war, ein Objekt der Klasse `KREISFORM` mit Hilfe des `new` Operators zu erzeugen. Dies wurde durch einen entsprechenden Mausklick im Kontextmenü erreicht. Der Aufruf des `new` Operators ist auch durch eine als Text formulierte Anweisung möglich. Notiert werden muss dazu das Schlüsselwort `new`, gefolgt von dem Konstruktor `KREISFORM()`:

```
new KREISFORM();
```

Ausgangspunkt in Abbildung 13 war, dass das Referenzattribut `symbol` noch auf kein Objekt verweist. So ist es erforderlich, das erzeugte Objekt der Klasse `KREISFORM` dem Referenzattribut `symbol` zuzuweisen. Dies erfolgt über den Zuweisungsoperator „`=`“.

```
symbol = new KREISFORM();
```

### Hinweis:

Mache dir bewusst, dass in dieser Programmzeile zwei Aktionen geschehen. Dabei ist die Reihenfolge der Aktionen umgekehrt zur Reihenfolge der Notation als Quelltext. Ähnlich zur Zuweisung in Kapitel 3.2 würde man die Quelltextzeile wie folgt sprechen:

„Es wird zuerst ein neues Objekt der Klasse `KREISFORM` erzeugt und eine Referenz darauf dem Referenzattribut `symbol` zugewiesen“.

Ist das Objekt erzeugt, müssen die Werte der Attribute `radius`, `startWinkel`, `bogenWinkel`, `bogenArt` und `farbe` wie in Aufgabe 2.9 über Methodenaufrufe richtig gesetzt werden. Dies ist nun kein Problem, weil über die Referenz das Objekt `mampfi` mit seinem Symbol kommunizieren kann (Abbildung 10). So kann `mampfi` Botschaften mit Methodenaufrufen in Punktnotation an das Objekt `symbol` schicken. Alle Methodenaufrufe, inklusiv Aufruf des `new` Operators müssen in den Konstruktor `MAMPFI` geschrieben werden, da gleich nach dem Erzeugen von dem Objekt `mampfi` unsere Spielfigur auch graphisch dargestellt werden soll.

Der gesamte Konstruktor der Klasse MAMPFI lautet somit:

```
MAMPFI()
{
    verwundbar = true;
    blickrichtung = 'N';

    symbol = new KREISFORM();
    symbol.RadiusSetzen(50);
    symbol.StartWinkelSetzen(120);
    symbol.BogenWinkelSetzen(300);
    symbol.BogenArtSetzen(2);
    symbol.FarbeSetzen("gelb");
}
```

Abbildung 15: Java Quelltext des Konstruktors MAMPFI - Im Vergleich zur ersten Variante in Abbildung 5 wurde ergänzt: Erzeugung eines Objekts der Klasse KREISFORM und Zuweisung dem Referenzattribut symbol; Aufruf von Methoden des Objekts der Klasse KREISFORM, um dort passende Attributwerte zu setzen.



### Aufgabe 3.10

Ergänze den Konstruktor der Klasse MAMPFI. Übersetze die Klasse und erzeuge ein Objekt. Betrachte die Attributwerte im Objektinspektor.

Überprüft man nun den Wert zum Attribut symbol im Objektinspektor, so ist die Angabe null verschwunden. Ein Pfeil ganz ähnlich zu Abbildung 12 zeigt die Referenz an. Durch einen Doppelklick auf den Pfeil öffnet sich das Objektinspektorfenster des referenzierten Objekts. Dort kann man die Attributwerte des Objekts der Klasse KREISFORM einsehen.

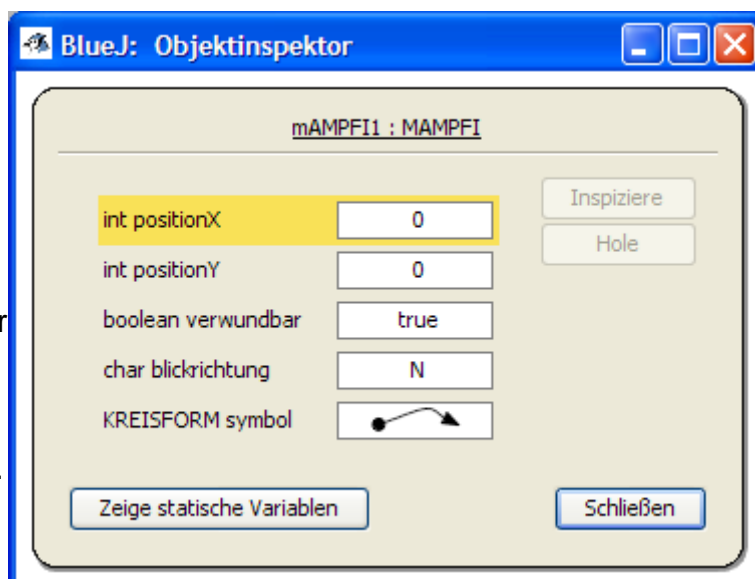
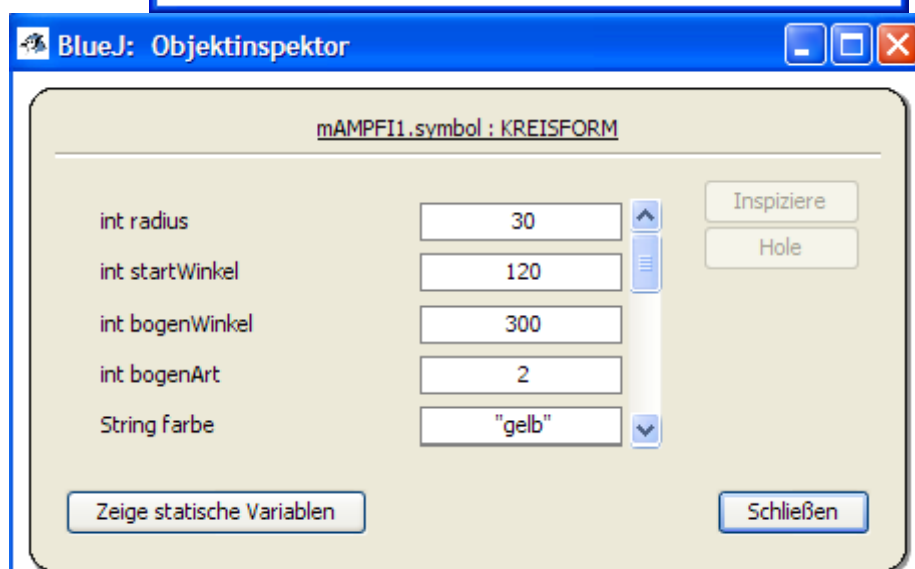


Abbildung 16: Objektinspektor der Objekte mampfi und symbol – durch einen Doppelklick beim Pfeil im Objektinspektor des Objekts mampfi, der neu beim Referenzattribut form erschienen ist, gelangt man zum Objektinspektor des referenzierten Objekts symbol



## Aufgabe 3.11



Welchen Vorteil hat die eben erstellte Klasse im Vergleich zur Lösung von Aufgabe 2.8?

Mache dir in diesem Zusammenhang Gedanken zur Frage wie lange Objekte leben. Notiere Antwortvorschläge schriftlich ins Heft.

### 3.4 Zusammenfassung

## Aufgabe 3.12

In diesem Kapitel hast du deine erste Klasse in Java erstellt. Vieles wurde in diesem Zusammenhang an Neuem vorgestellt.

Erstelle eine Zusammenfassung in dein Heft, bei der folgende Punkte enthalten sein sollten:

- Datentypen
- Konstruktor
- Kommentar
- Referenzattribute
- Erweitertes Klassendiagramm
- Aufbau einer Klasse in Java (nur Attribute, Referenzattribute und Konstruktor)
- Zuweisung
- Objektkommunikation
- new-Operator

## zusätzliche Informationen



Informatik II: Programme erstellen S. 6ff

Compiler



Informatik II: Variablenkonzept S. 26

Variablen-  
konzept