Kapitel 2 Objekte in der Entwicklungsumgebung BlueJ

Lernziele :

Entwicklungsumgebung BlueJ: Objekte erzeugen, Objektinspektor, Methodenaufrufe

2.1 Objekte erzeugen

Ziel ist es in den kommenden Wochen Schritt für Schritt ein Spiel in der Art von Pacman zu programmieren. Es gibt viele Programmiersprachen, in denen man dies durchführen könnte. In dieser Lernsequenz wird Java verwendet. Weiterhin gibt es hilfreiche Werkzeuge, sogenannte Entwicklungsumgebungen, die helfen Programme zu erstellen, Fehler zu suchen, Dokumentationen zu verfassen usw. Eine sehr gute Entwicklungsumgebung für Anfänger ist BlueJ. Diese wird im Folgenden vorgestellt:

Compiler

Aufgabe 2.1

Du erhältst von deinem Lehrer ein BlueJ-Projekt mit dem Namen Kreisform. Kopiere es in deinen persönlichen Speicherbereich. Starte nun BlueJ und Öffne über *Projekt --> Projekt Öffnen* das soeben gespeicherte Projekt.

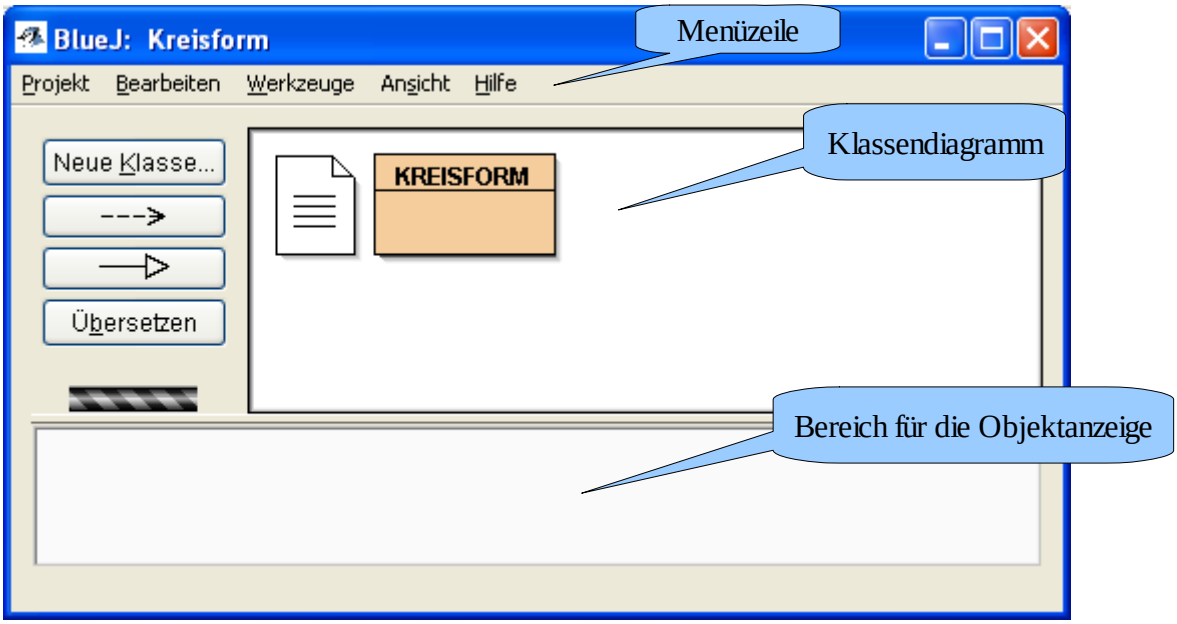

Abbildung 1: Aufbau des BlueJ Projektfensters: Menüzeile, Klassendiagramm und Bereich für die Objektanzeige

Das Projekt enthält eine Klasse mit dem Namen KREISFORM. Es lassen sich nun Objekte dieser Klasse erzeugen, indem du mit der rechten Maustaste auf das Klassensymbol klickst und im Kontextmenü „*new Kreisform*“ auswählst.

Es erscheint folgendes Fenster:

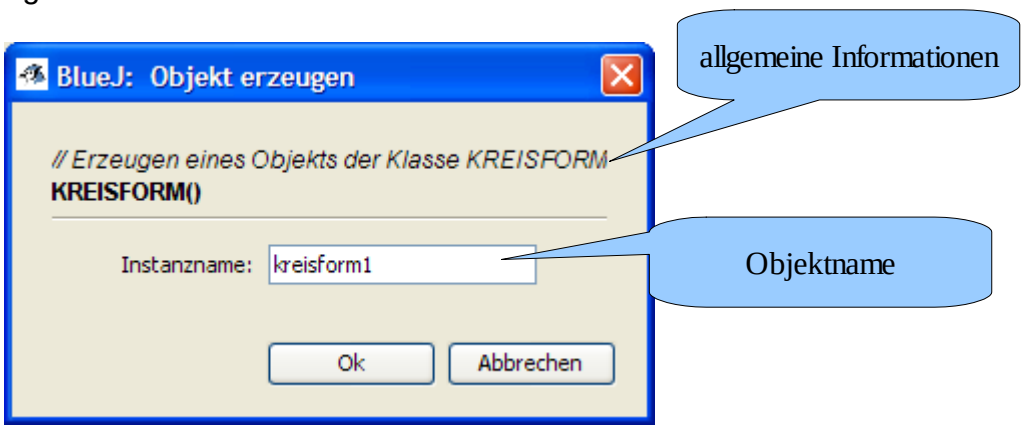

Abbildung 2: Eingabefenster beim Erzeugen eines Objekts: Der Objekt-name kann gewählt werden.

Durch Bestätigung über die Schaltfläche mit der Beschriftung OK wird das Objekt erzeugt. Es öffnet sich ein neues Fenster, in dem das Objekt zu sehen ist (Abbildung 3).

Abbildung 3: Anzeigefenster mit dem Objekt der Klasse KREISFORM

Aufgabe 2.2
Erzeuge drei Objekte der Klasse KREISFORM mit den Namen kreisform0, kreisform1 und kreisform2.

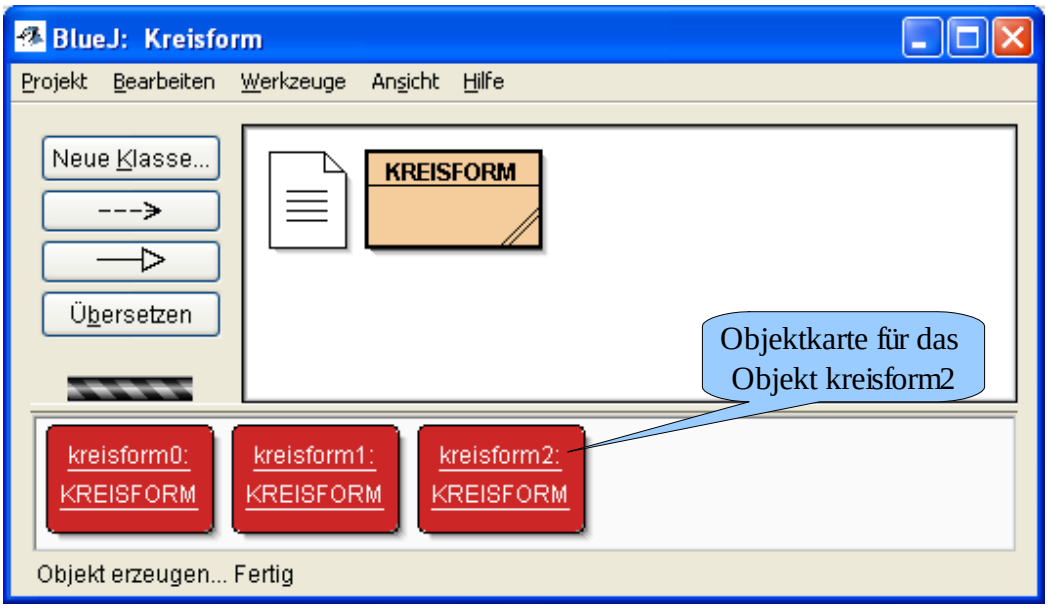

Abbildung 4: Das BlueJ Projektfenster nach dem erzeugen von drei Objekten. Im Bereich der Objektanzeige wird jedes Objekt durch eine rote Objektkarte symbolisiert.

Hinweise:

- Es ist sinnvoll aussagekräftige Namen zu vergeben und durch eine einheitliche Schreibweise (Namenskonvention) zu visualisieren, ob es sich um eine Klasse oder ein Objekt handelt. In dieser Lernsequenz werden Objekte mit Kleinbuchstaben und Klassen mit Großbuchstaben geschrieben.

- Häufig liegt das Fenster mit der graphischen Anzeige deiner erzeugten Objekte im Hintergrund. Du kannst es beispielsweise über die Taskleiste in den Vordergrund holen.
- Von einer Klasse können beliebig viele Objekte erzeugt werden**(Grundwissen)**. Dies ist durch die vielen roten Objektkarten der einen Klasse KREISFORM deutlich zu sehen. Anschaulich kannst du dir eine Klasse als Bauanleitung denken. Die Objekte werden beim Erzeugen nach dieser Anleitung gebaut.
- Ein Objekt einer Klasse wird in der Fachsprache auch als **Instanz** einer Klasse bezeichnet. Beispielsweise verwendet BlueJ diesen Ausdruck (Abbildung 2).

2.2 Klassendiagramm als wichtige Informationsquelle

Nun hast du Objekte erzeugt, aber welche Eigenschaften und Fähigkeiten haben diese Objekte?

Sehr hilfreich als Orientierung ist ein Klassendiagramm, denn genau dort sind die Attribute und Methoden aufgelistet. Das Diagramm der Klasse KREISFORM ist in der Abbildung 5 zu sehen.

Hinweis:
Ist eine entsprechende Erweiterung von BlueJ installiert (siehe 2.5), dann kann über einen Klick mit der rechten Maustaste auf die Klasse das Klassendiagramm über das Kontextmenü *Display UML* angezeigt werden.

Abbildung 5:
Das Klassendiagramm gibt Auskunft über die Attribute und Methoden der Klasse KREISFORM.

2.3 Attributwerte ansehen und durch Methodenaufrufe verändern

Durch einen Doppelklick auf eine Objektkarte wird der Objektinspektor geöffnet. Dort siehst du die Attribute und ihre aktuellen Werte.

Abbildung 6: Über den Objektinspektor erhälz man Informationen über die Werte der einzelnen Attribute eines Objekts.

Sollen nun Attributwerte verändert werden, benötigt man die Methoden eines Objekts. Durch das Senden einer Botschaft beispielsweise mit dem Aufruf der Methode *RadiusSetzen* wird der Wert des Attributs *radius* neu gesetzt.
In BlueJ sind Methodenaufrufe mit dem Kontextmenü möglich, d.h. durch einen Klick mit der rechten Maustaste auf eine Objektkarte, werden die Methoden angezeigt, die ein Objekt anbietet. In Abbildung 7 ist das Kontextmenü mit den Methoden eines Objekts der Klasse KREISFORM gezeigt:

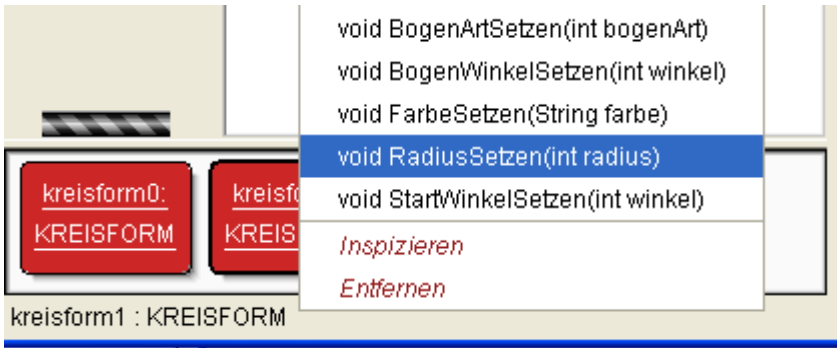

Abbildung 7: Im Kontextmenü einer Objektkarte werden alle verfügbaren Methoden eines Objektes angezeigt. Benötigt eine Methode einen Eingabewert, so steht der Eingabeparameter mit Datentyp hinter dem Methodennamen in Klammern.

Bei einem Methodenaufruf erscheint ein ähnliches Fenster wie beim Erzeugen von Objekten. Damit die Methode *RadiusSetzen* arbeiten kann, benötigt sie die Information, auf welchen neuen Wert das Attribut *radius* gesetzt werden soll.

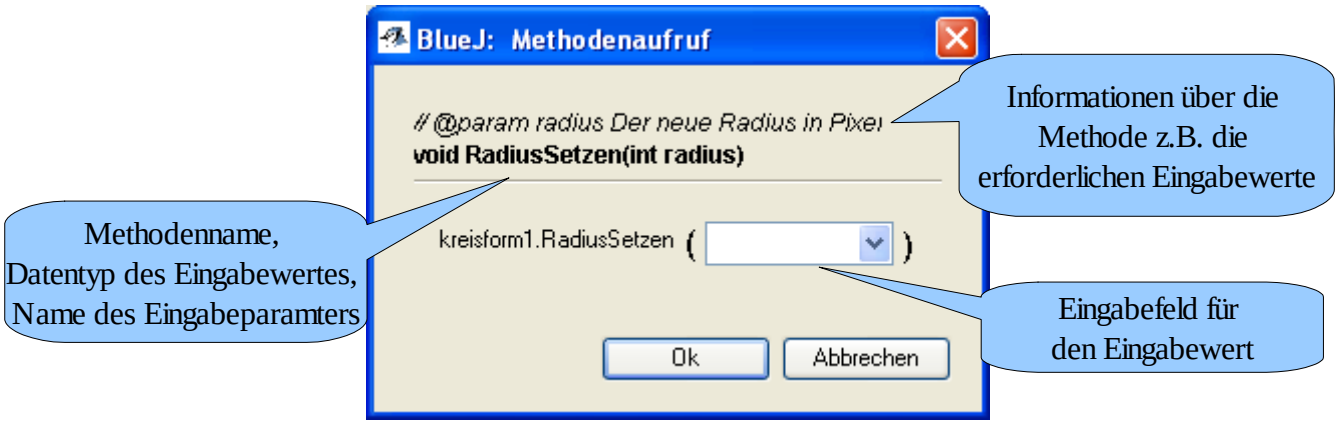

Abbildung 8: Eingabefenster beim Aufruf der Methode RadiusSetzen: Neben Informationen zur Methode in Textform, wird der Methodenname, der Datentyp des Eingabewertes und der Name des Eingabeparameters angezeigt. Im Eingabefeld kann dann ein konkreter Wert eingegeben werden, mit dem dann die Methode ausgeführt wird.

Diesen Eingabewert kann man in ein Eingabefeld eintragen. Der Ersteller der Klasse, zu der das Objekt *kreisform1* gehört hat dabei festgelegt, von welchem Typ dieser Eingabewert sein muss. Da der Radius in Pixel angegeben wird, muss der Wert als ganze Zahl angegeben werden. Die Festlegung des Programmierers kannst du an der Angabe des Datentyps `int` innerhalb der Klammer erkennen. Andere grundlegende Datentypen kannst du der Tabelle in der Abbildung 8 entnehmen.

Datentyp	Bedeutung	Beispiel	Anmerkung
int	ganze Zahl	34	
double	Dezimalzahl	3.14d	Schreibweise mit einem d für double
char	Zeichen	'N'	Hochkommas erforderlich
String	Zeichenkette	"gelb"	Anführungszeichen erforderlich
boolean	Wahrheitswert	true oder false	

Abbildung 9: Datentypen und ihre Bedeutung

Aufgabe 2.3
Öffne vom Objekt kreisform1 den Objektinspektor. Positioniere ihn so auf deinen Bildschirm, dass er nicht das BlueJ Hauptfenster (Abbildung 1) und das Anzeigefenster (Abbildung 3) überdeckt. Ordne deine Fenster in etwa in folgender Form an:

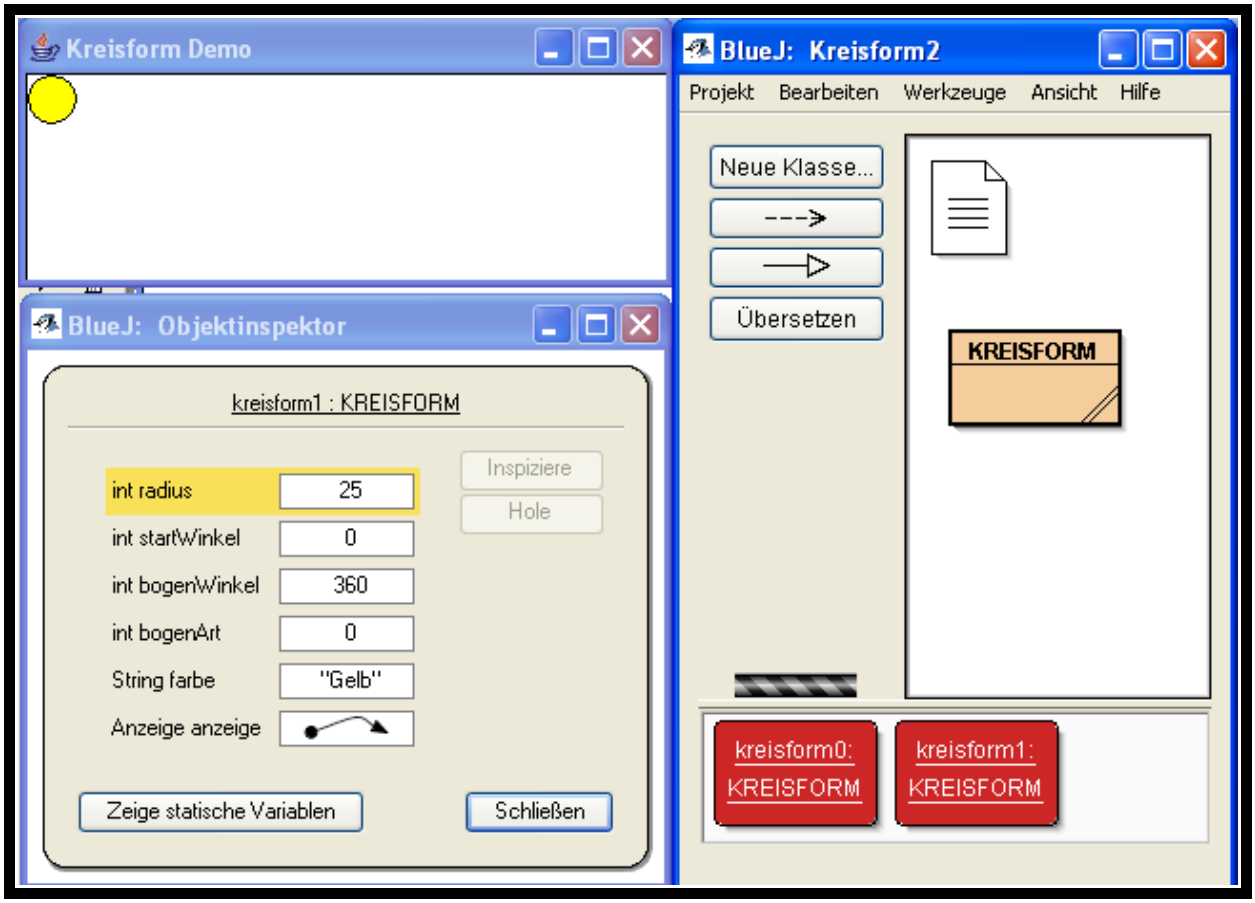

Abbildung 10: Sinnvolle Anordnung der an der Entwicklung beteiligten Fenster: Anzeigefenster (links oben), BlueJ-Projekt-Fenster (rechts) und Objektinspektor des Objekts .kreisform1 (links unten)

Rufe nun die Methode *RadiusSetzen* auf und gib einen Zahlenwert ein. Beobachte nach dem Aufruf, den Wert für den Radius im Objektinspektor.

Aufgabe 2.4
Rufe beim gleichen Objekt die Methode *FarbeSetzen* auf. Beachte, dass hier der Eingabewert eine Zeichenkette sein muss. Zeichenketten in Java werden immer durch umrandete Anführungszeichen gekennzeichnet, z. B. “rot“.

Aufgabe 2.5
Experimentiere nun mit den Methoden *BogenArtSetzen*, *BogenWinkelSetzen* und *StartWinkelSetzen*. Hilfreich sind die Informationstexte in den Fenstern mit dem Feld für den Eingabewert wie in der Abbildung rechts.

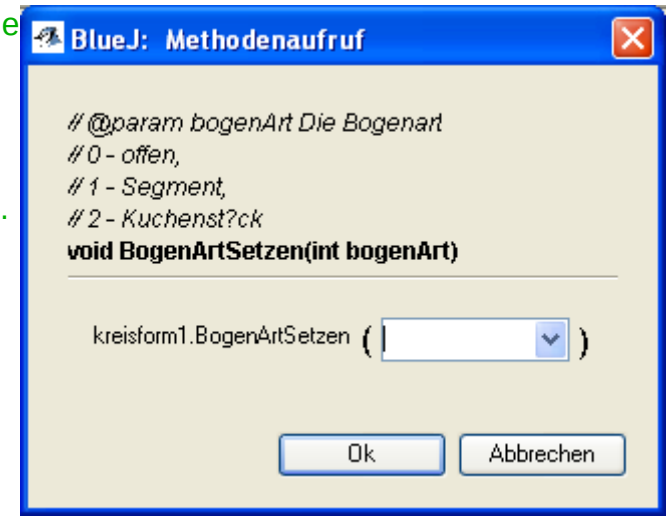

Abbildung 11: Eingabefenster beim Aufruf der Methode BogenArtSetzen; hilfreich sind hier die Informationen zum Eingabewert

Versuche dadurch die Bedeutung der Attribute zu verstehen, die durch die genannten Methoden verändert werden. Erkläre in deinem Heft an Hand einer Skizze die Bedeutung der Attribute Startwinkel und Bogenwinkel.

2.4 Methoden und Methodenaufrufe

Du hast zur Lösung der letzten Aufgaben viele Methoden per Mausklick aufgerufen. Mit jedem Mausklick hast du einem Objekt der Klasse KREISFORM eine Botschaft geschickt, etwas Konkretes zu tun. In Abbildung 12 ist dies veranschaulicht.

Aufgabe 2.6
Erkläre den Unterschied zwischen einer Methode und einem Methodenaufruf. Als Hilfe soll dir folgende Abbildung dienen, die einen Methodenaufruf veranschaulicht.

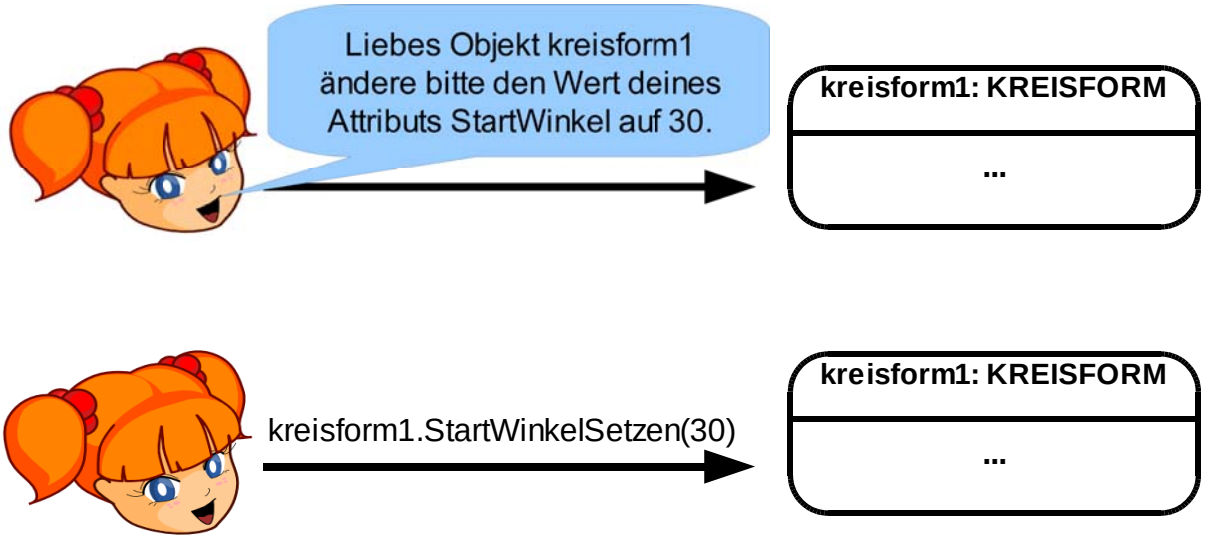

Abbildung 12: Veranschaulichung eines Methodenaufrufs durch eine externe Aufruferin – die Schülerin fordert das Objekt kreisform1 auf, den Wert seines Attributs StartWinkel auf 30 zu setzen.

Quelle des Cliparts in der Abbildung: gopher at http://openclipart.org

Aufgabe 2.7
Im unteren Teil der Abbildung 12 ist der Pfeil mit einem Methodenaufruf in Punktnotation beschriftet. Erkläre die Bestandteile dieser Punktnotation.

2.5 Pacman mit einer Kreisform darstellen

Aufgabe 2.8
Erstelle vier verschiedene Objekte der Klasse KREISFORM mit den Namen pacmanOst, pacmanSued, pacmanWest, pacmanNord. Sie sollen jeweils einen Pacman mit unterschiedlichen Blickrichtungen darstellen. Notiere dir dabei alle Methodenaufrufe mit wichtigen Eingabewerten , du wirst sie an späterer Stelle benötigen.

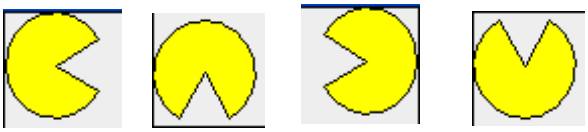

Abbildung 13: Pacman dargestellt durch ein Objekt der Klasse KREISFORM

2.6 BlueJ installieren

Möchtest du zu Hause auch BlueJ installieren findest du die Installationsquellen unter www.digitale-schule-bayern.de --> Informatik --> Werkzeuge --> Entwicklungsumgebungen

Beachte:
- **Vor** der Installation von BlueJ muss Java (z.B. jdk 6) installiert sein!!!
- Installiere auch die UML-Erweiterung. Installationsquelle und -hinweis findest du ebenfalls auf der digitalen schule.

2.7 Zusammenfassung

Aufgabe 2.9
In diesem Kapitel wurden wichtige Grundbegriffe aus den früheren Jahrgangsstufen wiederholt bzw. neu vorgestellt. Erkläre diese knapp. Die Erklärung sollte allgemein sein, jedoch auch ein kleines Beispiel zur Veranschaulichung beinhalten.
- Klassendiagramm
- Methodenaufruf versus Methode
- Eingabeparameter bzw. Eingabewert einer Methode
- Datentyp

zusätzliche Informationen

Compiler

Informatik II: Programme erstellen S. 6f

Namens-konvention

Namenskonvention

Softwareentwickler halten sich an Vereinbarungen (engl. convention) hinsichtlich Layout und Namensgebung. Dadurch werden Programme leichter lesbar und damit besser verständlich. Abhängig von Programmiersprache und Entwicklergruppe können unterschiedliche Konventionen vorliegen. Als Beispiele für Namenskonventionen sind in der folgenden Tabelle die Vereinbarungen für dieses Unterrichtskonzept und für internationale Java Programmierer aufgelistet.

Typ	allgemein	Schreibweise Krümel & Monster Informatik I und II	Schreibweise internationaler Java Programmierer
Klassen und Schnittstellen	Klassennamen sollten Substantive sein	Nur Großbuchstaben z. B. DATENELEMENT, KNOTEN	Beginnt mit einem Großbuchstaben z. B. TextField, BigInteger
Methoden	Methodennamen sollten Verben sein bzw. enthalten; haben immer ein Klammerpaar	Beginnt mit einem Großbuchstaben z. B. InforationAusgeben(), Einfuegen(DATENELEMENT datenNeu)	Beginnt mit einem Kleinbuchstaben z.B. getminimumSize(), compareTo(String otherString)
Attribute und Objekte	Attribut- und Objektnamen sollten aussagekräftig, aber kurz sein	Beginnt mit einem Kleinbuchstaben z.B. wurzel, rechterNachfolger	Beginnt mit einem Kleinbuchstaben z.B. root, myTextField