

Folge 1 - Wir lernen das Programm kennen

Vorbemerkungen

Ich gehe davon aus, dass Sie das Programm BlueJ auf Ihrem Rechner installiert haben. Zur Original BlueJ-Seite kommen Sie, wenn Sie dem Link www.bluej.org folgen.

Ich lehne mich in meinem Lehrgang anfangs stark an die erste Auflage des hervorragende Buchs "Objects first with Java" von David. J. BARNES und Michael KÖLLING an. Inzwischen ist der Verlag bei der dritten Auflage angekommen, die deutsche Übersetzung heißt "Java lernen mit BlueJ", ist im März 2006 erschienen und kostet 39,95 € (ISBN 3-8273-7152-X).

Vorbereitungen

Wenn Sie erfolgreich an diesem Kurs mitarbeiten wollen, benötigen Sie erstens eine lauffähige Version von **BlueJ**. Wenn Sie BlueJ herunter geladen und entpackt haben, sehen Sie unter anderem einen Ordner "examples". Dieser Ordner enthält einige interessante Beispiel, mit deren Hilfe Sie sich mit BlueJ und Java vertraut machen können.

Bevor Sie BlueJ benutzen können, müssen Sie die **Java-Runtime-Umgebung** bzw. das **Java-SDK** von Sun herunter laden und installieren, wenn Sie einen Windows-Rechner haben. Bei einem Apple-Rechner unter MacOS X können Sie BlueJ direkt einsetzen, da Java hier zum Betriebssystem gehört.

Drittens benötigen Sie **ausreichend freien Speicherplatz** auf einem Laufwerk, auf das Sie Schreibzugriff haben. BlueJ hat Schwierigkeiten beim Kompilieren, wenn der Speicherplatz zu knapp wird.

1. Das BlueJ-Fenster

Starten Sie BlueJ und laden Sie das **Projekt** "Figuren" bzw. "shapes".

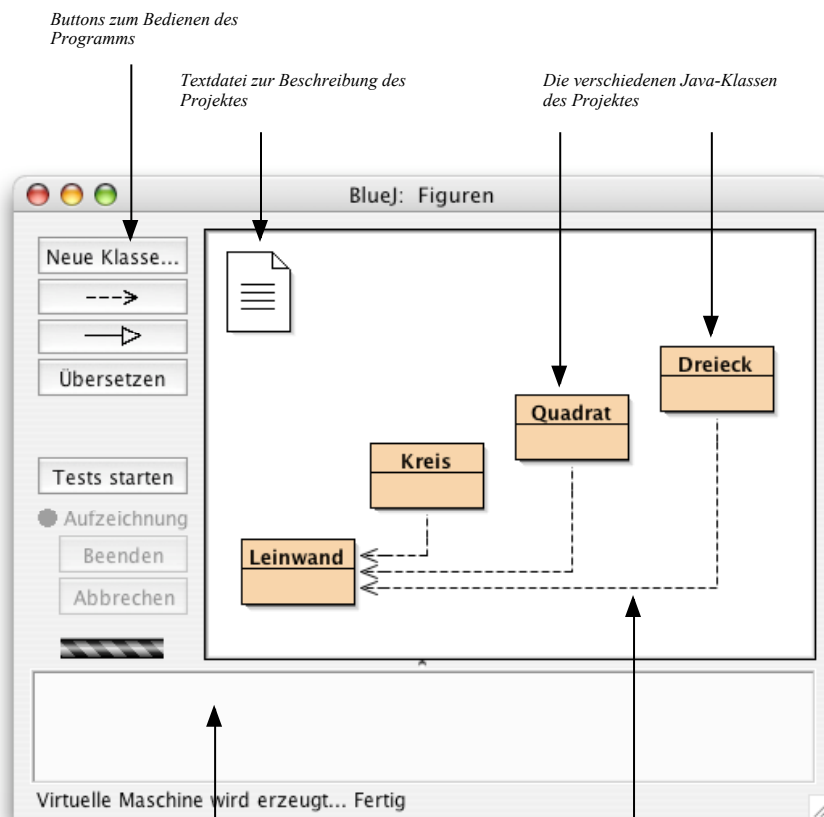
Betrachten Sie zunächst das **BlueJ-Fenster** (Abb. 1). Das Fenster gliedert sich in drei Teilbereiche. Links sehen Sie einen grau unterlegten schmalen Bereich mit einigen Buttons. Mit diesen Buttons steuern Sie die wesentlichen Programmfunktionen. In der Mitte finden Sie den **Hauptarbeitsbereich** des Fensters, in dem die **Klassen** Ihres **Projektes** dargestellt werden. Der untere Bereich schließlich dient zur Aufnahme von verschiedenen **Objekten**.

Im Hauptarbeitsbereich des BlueJ-Fensters aus Abbildung 1 kann man vier Klassen erkennen: **Leinwand**, **Kreis**, **Quadrat** und **Dreieck**.

2. Klassen und Objekte

Klicken Sie mit der rechten Maustaste auf die Klasse **Quadrat**. Ein so genanntes **Kontextmenü** öffnet sich, das Ihnen den Befehl "new Square()" und weitere **Befehle** zur Auswahl stellt.

Falls der Befehl "new Square()" bzw. "neues Quadrat()" im Kontextmenü nicht vorhanden ist, so müssen Sie die Klasse **Square** bzw. **Quadrat** erst **kompilieren**. Rufen Sie dazu den Befehl Kompilieren



1.1 Die BlueJ-Benutzeroberfläche

(engl. Version: compile) auf. Durch das Kompilieren wird der **Java-Quelltext** der Klasse **Quadrat** jetzt in so genannten **Java-Code** übersetzt.

Wenn Sie mit den Begriffen "Quelltext", "Kompilieren" und "Java-Code" noch gar nichts anfangen können, lesen Sie den Anhang: "Programme und Programmiersprachen".

Wenn der Übersetzungsvorgang erfolgreich abgeschlossen wurde, können Sie mit der rechten Maustaste den Befehl "neues Quadrat()" aufrufen. Wenn Sie die englische Version von BlueJ verwenden, heißt der entsprechende Befehl "new square()".

Ich werde mich im Folgenden immer auf die deutsche Version beziehen. Wie Sie selbst BlueJ auf die deutsche Sprache umstellen können, erfahren Sie im Anhang C zu Folge 1.

Mit dem Befehl "neues Quadrat()" erzeugen Sie ein **Objekt** der Klasse **Quadrat**. Diesem Objekt müssen Sie einen Namen geben, z.B. "kasten" oder "haus".

Während die Bezeichner von Klassen immer mit Großbuchstaben beginnen, fangen die Bezeichner von Objekten stets mit Kleinbuchstaben an. Dies ist kein direkter Zwang, sondern eher eine Konvention, an die sich aber alle Java-Programmierer weitgehend halten.

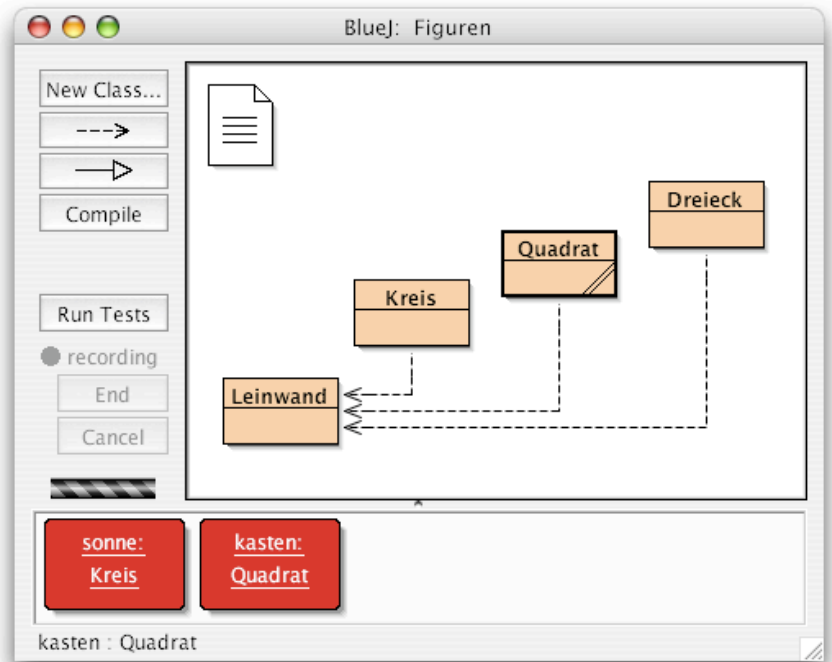
Erzeugen Sie jetzt ein zweites Objekt: Diesmal klicken Sie auf die Klasse **Kreis** und erzeugen ein Objekt namens "sonne". Ihr Projekt müsste jetzt so ähnlich aussehen wie Abbildung 2.

3. Methoden

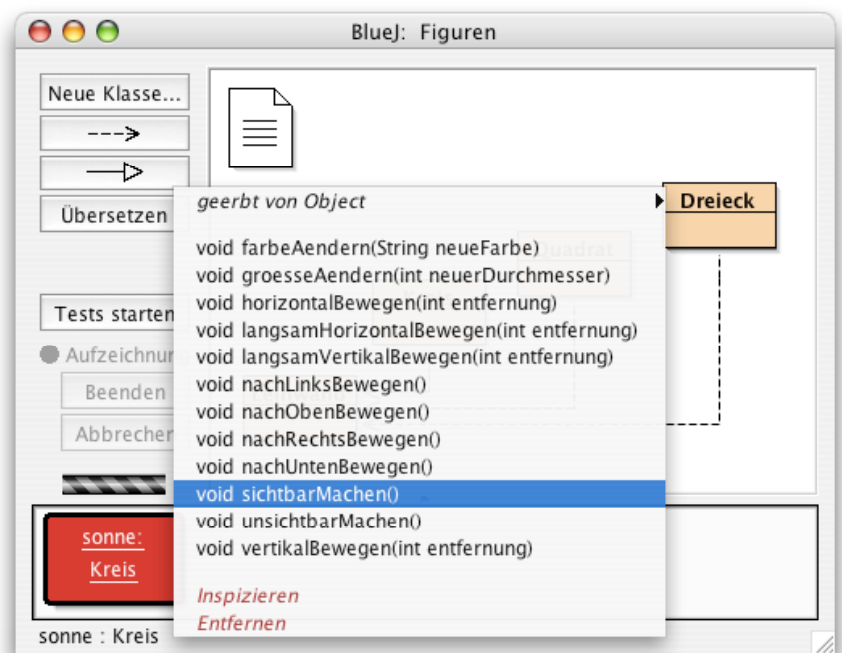
Was kann man jetzt eigentlich mit den erzeugten Objekten **kasten** und **sonne** machen? Klicken Sie mit der rechten Maustaste auf eines der beiden Objekte. Es erscheint ein recht langes **Kontextmenü** mit vielen **Methoden** und den beiden Befehlen "Inspect" und "Remove". Die angezeigten Methoden dienen zum **Manipulieren von Objekten**. Schauen Sie einmal genau hin, welche Methoden einem Kreis-Objekt zur Verfügung stehen.

Mit der Methode **farbeAendern()** kann man z.B. die Farbe eines Kreises ändern, mit der Methode **groesseAendern()** kann die Größe verändert werden und so weiter (siehe Abb. 3).

Mit Hilfe von Methoden kann man also ein Objekt verändern. Genauer gesagt, man kann den Zustand eines Objektes ändern. Den wichtigen Begriff **Zustand** werde ich im Theorieteil zu dieser ersten Folge noch näher erläutern.



1.2 Erzeugen von Objekten einer Klasse

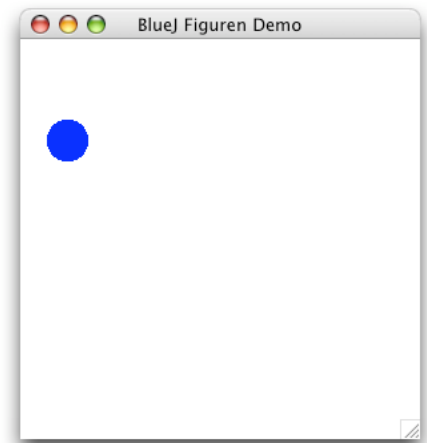


1.3 Die Methoden eines Objektes

4. Verändern der Objekte

Klicken Sie jetzt bitte mit der rechten Maustaste auf das Objekt **sonne** und rufen Sie aus dem Kontextmenü die Methode **sichtbarMachen()** auf. Es erscheint ein neues Fenster mit weißem Hintergrund und einem blauen Kreis (Abb. 4). Sie haben durch den Aufruf dieser Methode den Zustand des Objektes **sonne** verändert: Vorher war das Objekt nicht sichtbar, jetzt ist es in dem Fenster zu sehen. Methoden wie **sichtbarMachen()**, die den Zustand eines Objektes verändern, heißen **verändernde** oder **manipulierende Methoden**.

Wie jedermann weiß, ist unsere Sonne groß und gelb und nicht klein und blau. Sie sollen jetzt aus diesem "blauen Zwerg" eine normale gelbe Sonne machen. Dazu müssen Sie weitere verändernde Methoden aufrufen.

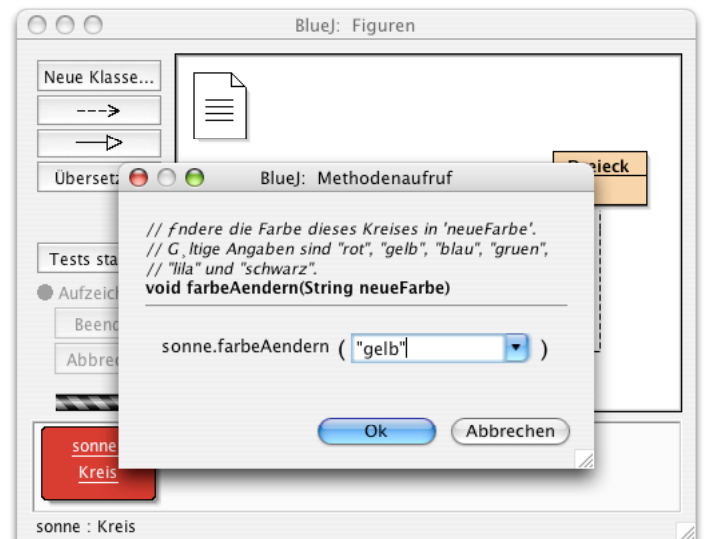


1.4 Ein sichtbar gemachtes Objekt der Klasse Kreis

4.1 Verändern der Farbe

Wenn Sie die Methode **farbeAendern()** aufrufen, erscheint ein Dialogfenster mit dem Titel "BlueJ: Methodenaufruf" (Abb. 5). Im oberen Teil des Fensters werden Sie ausführlich über die Bedeutung der Methode aufgeklärt. Im unteren Teil des Fensters müssen Sie die neue Farbe des Kreises eintippen. Wichtig ist, dass Sie die neue Farbe in Anführungszeichen setzen. Also nicht einfach gelb eingeben, sondern "gelb".

Können Sie den Unterschied zwischen den beiden Methoden **sichtbarMachen()** und **farbeAendern()** mit wenigen Worten zusammenfassen? Die erste Methode konnte *einfach so* aufgerufen werden. Es waren keine weiteren Angaben notwendig. Der Aufruf der Methode machte den vorher unsichtbaren Kreis sichtbar. Die zweite Methode kann *nicht einfach so* aufgerufen werden. Wenn Sie die Farbe eines Kreises ändern wollen, so müssen sie genau angeben, *welche Farbe* der Kreis denn jetzt bitte haben soll. Die Methode benötigt eine zusätzliche Information. In der Mathematik würde man hier von einem **Argument** sprechen, in der Informatik bezeichnet man solche zusätzlichen Informationen, die Methoden benötigen, als **Parameter**. Die Methode **farbeAendern()** benötigt genau *einen* Parameter, nämlich die neue Farbe. Es gibt aber auch Methoden, die *zwei, drei* oder *mehr* Parameter benötigen. Wenn Sie zum Beispiel ein Objekt der Klasse Dreieck anlegen und dann die Größe ändern wollen, so müssen Sie die neue Höhe und die neue Breite des Dreiecks eingeben, also zwei Parameter.



1.5 komfortable Eingabe eines Parameters in eine BlueJ-Methode

4.2 Größe verändern

Sie können sich bestimmt schon denken, wie die Größe der Sonne verändert werden kann. Richtig: Sie müssen die Methode **groesseAendern()** aufrufen. Auch diese Methode benötigt einen Parameter, nämlich die neue Größe, die der Kreis haben soll. Im Gegensatz zu **farbeAendern()** dürfen Sie jetzt keine **Zeichenkette** in das Textfeld des Dialogfensters eintippen, sondern es wird ein **Zahlenwert** erwartet. Tippen Sie jetzt die Zahl 100 ein - ohne Anführungszeichen wohl-gemerkt.

Haben Sie es bemerkt? Es gibt in Java unterschiedliche Typen von Parametern. Bei **farbeAendern()** mussten Sie einen Text in Anführungszeichen eintippen - einen so genannten **String** (deutsch: **Zeichenkette**). Bei **groesseAendern()** dagegen mussten Sie eine ganze Zahl eingeben.

Jetzt hat der Kreis nicht nur eine andere Farbe, sondern ist auch größer geworden. Durch den Aufruf der Methode **farbeAendern()** haben Sie das **Attribut farbe** des Objektes **kreisi** verändert. Vorher hatte das Attribut **farbe** den Attri-

butwert "blau", jetzt hat es den Attributwert "gelb". Durch den Aufruf der Methode **grosseAendern()** wurde entsprechend der Attributwert des Attributes **durchmesser** geändert.

Verändernde Methoden heißen deswegen so, weil sie mindestens einen Attributwert des Objektes verändern.

4.3 Position verändern

Als letzte Übung in diesem Schritt 4 sollen Sie die Position der Sonne verändern, so dass sie ungefähr in der Mitte des Fensters zu sehen ist (vergleiche Bild 7). Dazu eignen sich die Methoden **nachRechtsBewegen()** und **nachUntenBewegen()** hervorragend.

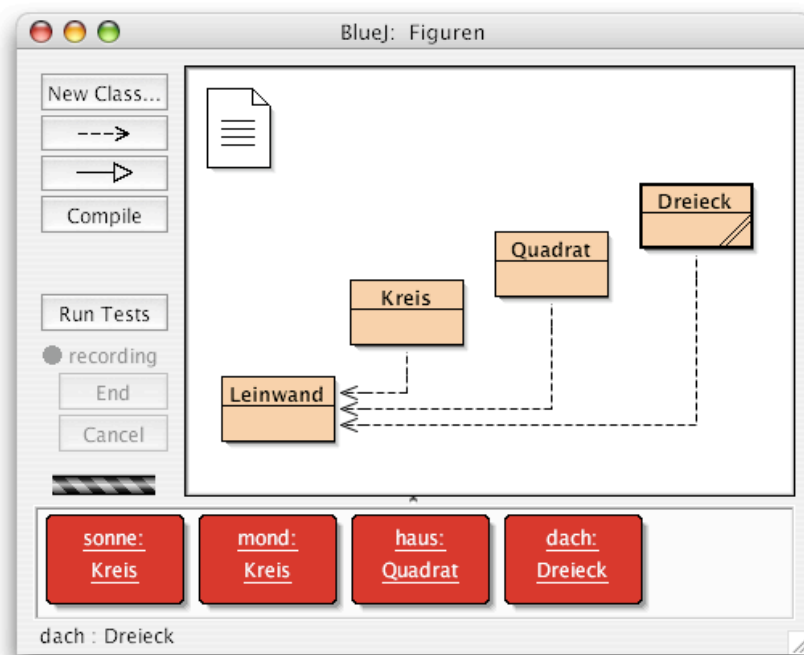
5. Ein tolles Bild zeichnen

Kontrollieren Sie bitte, ob Sie in Ihrem BlueJ-Projekt folgende Objekte erzeugt haben: Einen Kreis namens **sonne**, einen zweiten Kreis namens **mond**, ein Quadrat namens **haus** und ein Dreieck namens **dach**.

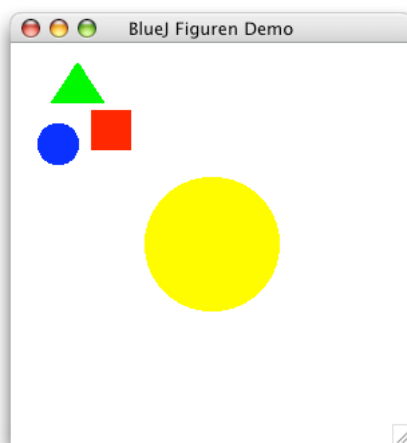
Rufen Sie dann für jedes der Objekte die Methode **sichtbarMachen()** auf. Ihr "Gemälde" sollte dann ungefähr so aussehen wie in Abbildung 8, falls Sie den Kreis **sonne** bereits wie oben besprochen verändert haben. Wenn nicht, sehen Sie nur *einen* blauen Kreis, der in Wirklichkeit aber aus *zwei* Kreisen **sonne** und **mond** besteht, die jedoch die gleichen Koordinaten haben und daher übereinander liegen.

6. Übung 1.1

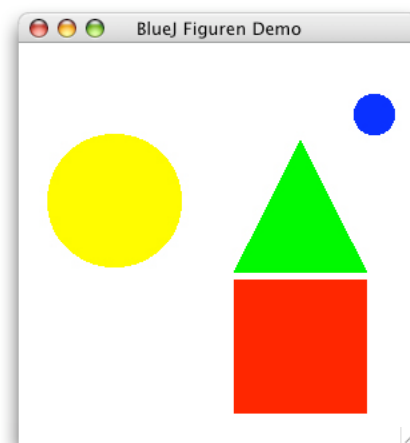
Versuchen Sie jetzt durch geschickten Aufruf geeigneter Methoden ein Bild zu malen, wie es in der Abbildung 8 dargestellt ist.



1.6 Die vier Objekte, die Sie für Ihr Bild benötigen.



1.7 So sieht ihr "Bild" zunächst aus.



1.8 Dieses Bild sollen Sie zeichnen.