



Java 2 SDK und JBuilder

Mit Milch und Zucker



Java schickt sich als plattformunabhängige Programmiersprache an. Überzeugen Sie sich selbst mit dem Java 2 SDK und JBuilder für Linux und Windows.

OLIVER MÜLLER

Java ist bekanntlich eine plattformunabhängige Programmiersprache und Laufzeitumgebung.

Wohingegen früher Java-Entwicklungsumgebungen und -Tools nur als native Anwendungen verfügbar waren, geht inzwischen der Trend in Richtung Java-Tools als Java-Anwendung.

Programme wie JBuilder oder Together/J werden heute selbst (oder unter anderem) als Java-Anwendungen

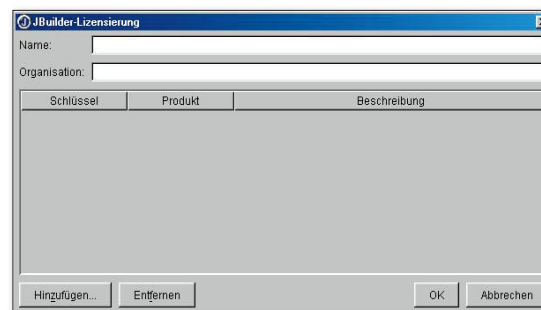
QUICK INDEX

- **Frag nach Sonnenschein**
Der Beginn aller Java-Aktivitäten ist das Java 2 SDK von Sun.
- **Baumeister**
JBuilder ist einfach durch ein grafisches Setup zu installieren.
- **Pilotprojekt**
Das Anlegen eines neuen Projekts in JBuilder geht per Point&Click.
- **Neues braucht das Projekt**
Neue Bestandteile, wie Fenster oder Dialoge, für das Projekt lassen sich automatisch erzeugen.
- **Oberflächlichkeiten**
Als ersten Schritt erstellen Sie die Oberfläche Ihres Programms.
- **Mit Leben erfüllt**
Im zweiten Schritt füllen Sie die Oberfläche mit Aktionen.
- **Hinein ins Getümmel**
Das Programm ist geschrieben. Einem Start steht nichts mehr im Wege.

ausgeliefert. Das hat den Vorteil, dass die Software quasi auf allen Plattformen läuft, für die eine Java-Umgebung zur Verfügung steht. So einfach ist es zwar nicht, da die Software selbst an das System, auf dem sie laufen soll, angepasst werden muss. Denken Sie in diesem Zusammenhang nur an die unterschiedlichen Dateisysteme.

Für den Einsteiger ist ein Vorteil noch hervorzuheben. Viele Java-Entwicklungstools werden in abgespeckten Versionen zum freien Download zur Verfügung gestellt. Um sich in Java und Objektmodellierung in der Praxis einzuarbeiten, muss man heute nicht sofort Hunderte oder Tausende Mark auf den Ladentisch blättern, sondern kann erst einmal in Ruhe zum Nulltarif üben.

Später kann man immer noch eine der "großen" Versionen erwerben, wenn man an die Grenzen der freien Versionen stößt.



JBUILDER MÜSSEN SIE durch einen Aktivierungsschlüssel freischalten, den Sie bei Inprise erhalten.

■ Frag nach Sonnenschein

Der Standard schlechthin im Java-Bereich ist das JDK bzw. Java 2 SDK (J2SDK) von Sun. Die im JDK enthaltene JVM (Java Virtual Machine) ist mehr oder minder der Dreh- und Angelpunkt für Java. Auf der CD-ROM finden Sie im Verzeichnis JDK/1.2 das Java 2 SDK für Linux und Windows in der Version 1.2.

Java für Linux

Die Installation von JDK unter Linux ist etwas Tippakrobatik. Mit anderen Worten: Sie müssen mit der Shell arbeiten und Kommandos eingeben.

Loggen Sie sich zunächst als "root" ein. Falls Sie unter X11 arbeiten, öffnen Sie ein Shell-Fenster. Nun müssen Sie das .tar.gz-Archiv von CD entpacken. Im folgenden wird JDK im Verzeichnis `/usr/local/jdk1.2.2` installiert.

Mounten Sie zunächst die CD-ROM. Dies können Sie entweder durch ein entsprechendes Symbol auf Ihrem GNOME- oder KDE-Desktop bewerkstelligen, oder Sie geben das folgende Kommando ein:

```
mount /mnt/cdrom
```

Bitte beachten Sie bei diesem Kommando, dass SuSE Linux hier einen eigenen Weg geht. CD-ROMs werden hier unter `/cdrom` gemountet, so dass Sie unter diesem System folgenden Befehl verwenden müssen:

```
mount /cdrom
```

Die folgenden Pfadangaben



müssen Sie jeweils anpassen. Statt /mnt/cdrom müssen Sie /cdrom einsetzen.

Wechseln Sie in das Verzeichnis /usr/local, durch den Befehl

```
cd /usr/local
```

Anschließend entpacken Sie das .tar.gz-Archiv durch Eingabe von

```
tar xzf
/mnt/cdrom/JDK/1.2/jdk1_2_2-linux
i386.tar.gz
```

Nach diesem Prozedere befindet sich das JDK im Verzeichnis /usr/local/jdk1.2.2. Die einzige Einstellung, die Sie vornehmen müssen ist nun die Aufnahme des Pfades /usr/local/jdk1.2.2 in Ihre PATH-Variable. Auf diese Weise können Sie Java durch die Eingabe der Kommandos der Pfade auf einfachste Weise nutzen.

TIPP Für das Basissystem müssen Sie keine weiteren Angaben über CLASSPATH oder JAVA_HOME machen, damit Ihr System läuft. Das Anpassen von PATH reicht unter Linux völlig aus, wenn Sie keine zusätzlichen Libraries installieren.

Für einen ersten Test genügt es, wenn Sie folgendes Kommando eingeben:

```
export
PATH="$PATH:/usr/local/jdk1.2.2/
bin"
```

Anschließend können Sie eines der mitgelieferten Demos starten. Geben Sie hierzu folgendes Kommando ein:

```
appletviewer
/usr/local/jdk1.2.2/demo/jfc/
Java2D/Java2Demo.html
```

oder

```
java -jar /usr/local/jdk1.2.2/
demo/jfc/Notepad/Notepad.jar
```

TIPP Sie müssen unter X11 arbeiten, damit Sie diese Beispiele starten können.

Damit Sie nicht jedes Mal die PATH-Anweisung von Hand anpassen müssen, lohnt es sich, die oben genannte Zeile in die Datei ~/.bashrc oder ~/.bash_profile aufzunehmen. Fügen Sie hierzu einfach die "export PATH"-Anweisung von vorhin ans Ende einer dieser Dateien an. Bitte bedenken Sie jedoch, dass Sie diese Änderung für jeden Benutzer vornehmen müssen, der Zugriff auf Java haben soll.

Wenn Sie wollen, dass alle Anwender diesen einfachen Zugriff auf das JDK verwenden sollen, können Sie als "root"

auch die Datei /etc/profile um die betreffende Zeile erweitern. Damit wirkt sich die Anpassung der PATH-Variablen auf alle Benutzer aus.

TIPP Die Einstellungen in ~/.bashrc, ~/.bash_profile und /etc/profile werden erst nach dem nächsten Login aktiv.

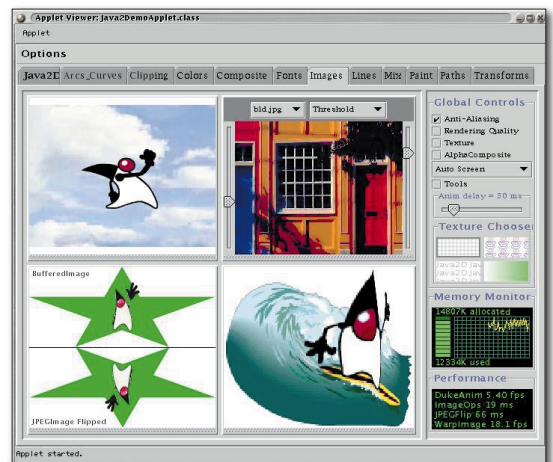
Java unter Windows

Die Installation unter Windows ist weniger aufwendig. Hier starten Sie zunächst einen grafischen Installationsassistenten. Diesen finden Sie auf der CD-ROM im Verzeichnis JDK\1.2. Starten Sie einfach das Programm jdk1_2_2-001-win.exe.

Das Installationsprogramm ist ein typisches Setup unter Windows. Sie können über die Schaltflächen Next und Previous vor bzw. zurückblättern und so alle Daten für die Installation zusammentragen.

Nach dem Begrüßungsdialog erscheint eine Lizenzvereinbarung. Diese müssen Sie mit Yes akzeptieren, um das Java 2 SDK überhaupt installieren zu können. Wenn Sie mit den Vereinbarungen nicht leben können, klicken Sie auf No. In diesem Fall findet Ihr Vorhaben das Java 2 SDK zu installieren ein jähes Ende.

Sollten Sie sich für die Lizenz entschieden haben, können Sie jetzt einen Zielordner auswählen. Dies ist das Ver-

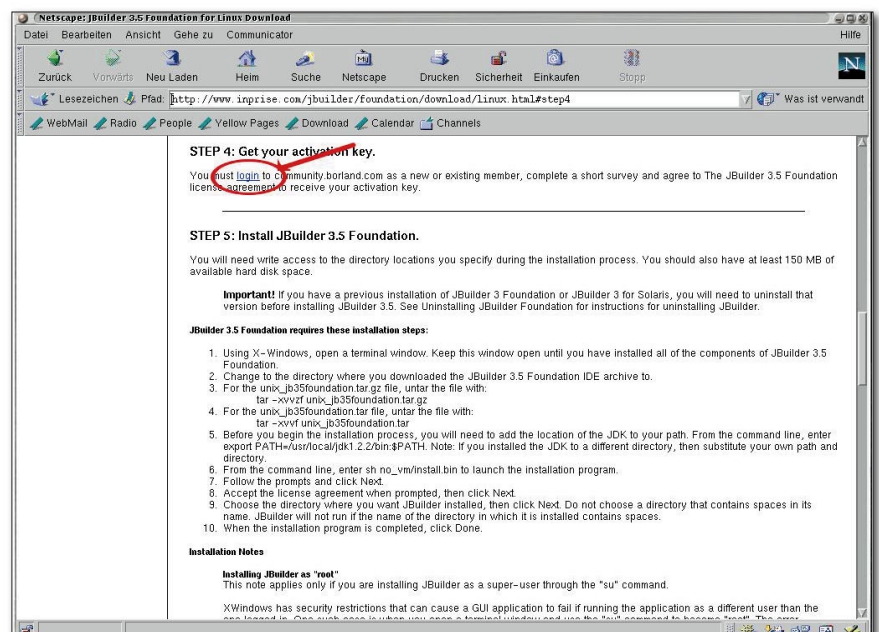


JAVA2DEMOAPPLET.CLASS – JDK in Aktion.

zeichnis, in dem das J2SDK installiert wird. Per Default ist hier C:\jdk1.2.2 angegeben. Im Normalfall können Sie dieses Verzeichnis unbesehen übernehmen. Wollen Sie das JDK aus irgendeinem Grund in ein anderes Verzeichnis installieren, klicken Sie auf die Schaltfläche links neben der Angabe C:\jdk1.2.2 und wählen einen anderen Zielordner.

Im nächsten Dialogfeld wählen Sie den Umfang der Installation. Sie können durch Auswahlfelder festlegen, welche Komponenten (Programme, Beispiele etc.) installiert werden sollen. Im Zweifelsfall übernehmen Sie die Voreinstellung, bei der alles installiert wird.

Zur Konfusion von J2SDK-Neulingen zielt wieder eine Lizenzvereinbarung den Bildschirm. Hierbei handelt es



EINEN AKTIVIERUNGSSCHLÜSSEL für JBuilder erhalten Sie im Web.



DER HANOI-ALGORITHMUS

```
void hanoi(String a, String b, String c, long n) {
    if(n > 1) {
        hanoi(a, c, b, n-1);
        textArea1.append("Scheibe von " + a + " nach " + b + "\n");
        hanoi(c, b, a, n-1);
    } else {
        textArea1.append("Scheibe von " + a + " nach " + b + "\n");
    }
}
```

sich um die Lizenz für das "Java Runtime Environment" (JRE). Dies ist die reine Laufzeitumgebung von Java 2. Damit können keine Programme entwickelt werden, sondern nur gestartet, also ausgeführt werden.

Es ist sinnvoll diese Umgebung ebenfalls zu installieren. Gerade wenn Sie später andere Java-Programme, zum Beispiel neue Tools, installieren wollen, sollten Sie JRE installieren. Häufig verwenden Java-Anwendungen nicht den Java-Interpreter *java* von J2SDK, sondern das Programm *jre* der Laufzeitumgebung. Damit Sie nicht womöglich unzählige Batch-Dateien per Hand editieren müssen, um von *jre* auf *java* umzustellen, ist es sehr zu empfehlen das JRE zu installieren. Akzeptieren Sie am besten auch hier die Lizenzvereinbarung mit *Yes*. Last but not least können Sie einen Zielordner für das JRE auswählen. Die Standardvorgabe *C:\Programme\JavaSoft\JRE1.2* können Sie normalerweise übernehmen, es sei denn Sie haben andere Pläne (zum Beispiel zu wenig Speicher auf Laufwerk C:).

Nach einem letzten Klick auf *Next* startet die Installation.

Ist die Installation zu einem erfolgreichen Ende gelangt, müssen Sie noch die PATH-Variable anpassen. Bearbeiten Sie zu diesem Zweck die Datei *C:\AUTOEXEC.BAT*. Fügen Sie das Kommando

```
set PATH=%PATH%;C:\jdk1.2.2\bin
```

an.

TIPP Unter Windows NT/2000 verwenden Sie zum Anpassen der PATH-Variablen die Systemsteuerung. Über das Symbol *System* erhalten Sie den Dialog *Systemeigenschaften*. Klicken Sie darin auf den Reiter *Umgebung* und passen Sie die PATH-Variable dort an.

Beachten Sie, dass diese Änderungen erst nach dem nächsten Login bzw. Start des Systems wirksam werden.

Die CLASSPATH-Variablen müssen Sie unter J2SDK nicht setzen. Sollten Sie von einer früheren JDK-Version diese

Variable gesetzt haben, wird Java 2 solange ohne Probleme funktionieren, wie das aktuelle Verzeichnis . darin enthalten ist. Einen ersten Test, ob das Java 2 SDK korrekt installiert wurde, können Sie mit den Beispielen durchführen. Geben Sie die Befehle auf der MS-DOS-Eingabeaufforderung ein:

```
appletviewer
C:\jdk1.2.2\demo\jfc\Java2D\Java2Demo.html
```

oder

```
java -jar C:\jdk1.2.2\demo\jfc\
Notepad\Notepad.jar
```

Baumeister

Im Internet steht JBuilder 4 Foundation unter <http://www.borland.com/jbuilder/foundation/> zum freien Download zur Verfügung. Alternativ können Sie dort die freie Version der preisgekrönten Java-Entwicklungsumgebung auf CD-ROM bestellen.

TIPP JBuilder 4 ist in der "Foundation-Version" für Windows auf PC Magazin 2/2000 enthalten.

Bevor Sie sich die Mühe machen und JBuilder installieren, sollten Sie sich für die Foundation einen Aktivierungsschlüssel bei Inprise besorgen. Damit

schalten Sie JBuilder frei, damit Sie überhaupt mit diesem Produkt arbeiten können. Diesen Schlüssel erhalten Sie ebenfalls unter der oben aufgeführten URL.

Nachdem es JBuilder in verschiedenen Versionen gibt, schalten die einzelnen Aktivierungsschlüssel nur einen bestimmten Funktionsumfang frei. Bei dem kostenlosen Aktivierungsschlüssel erhalten Sie nicht das gesamte Spektrum der JBuilder-Funktionen. Allerdings können Sie schon einfache Java-Programme und -Applets mit JBuilder Foundation erstellen.

TIPP Wichtig ist, dass Sie unter Linux auf jeden Fall das Java 2 SDK installieren, sonst können Sie das Setup-Programm von JBuilder gar nicht erst starten.

Auf die Platte, fertig, los!

Unter Windows starten Sie das Programm *FND_INSTALL.EXE*. Unter Linux sind ein paar Schritte mehr notwendig.

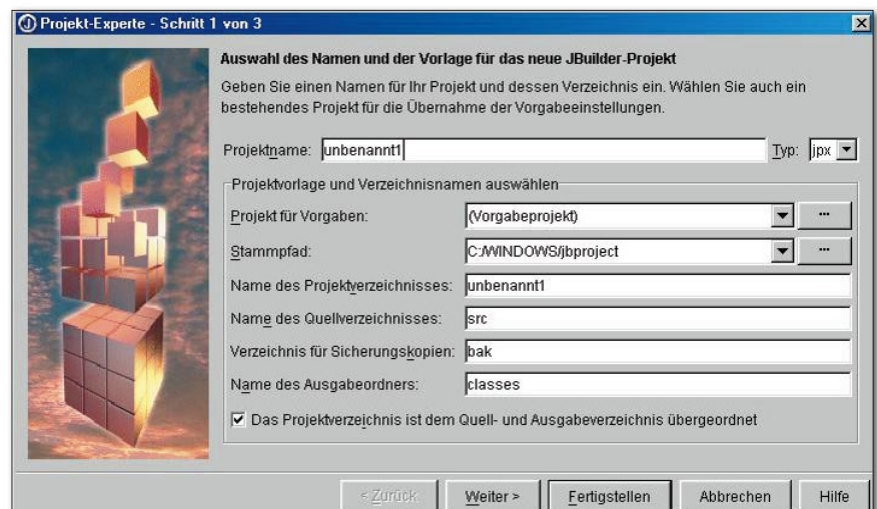
Logen Sie sich auch zur Installation von JBuilder wieder als "root" ein, und starten Sie X11, sofern es nicht schon automatisch gestartet wird. Anschließend öffnen Sie ein Shell-Fenster.

Zunächst wechseln Sie in das Verzeichnis, in dem Sie die Installationsdateien von JBuilder gespeichert bzw. entpackt haben. Angenommen Sie haben JBuilder unter */tmp/jbuilder* entpackt, dann geben Sie

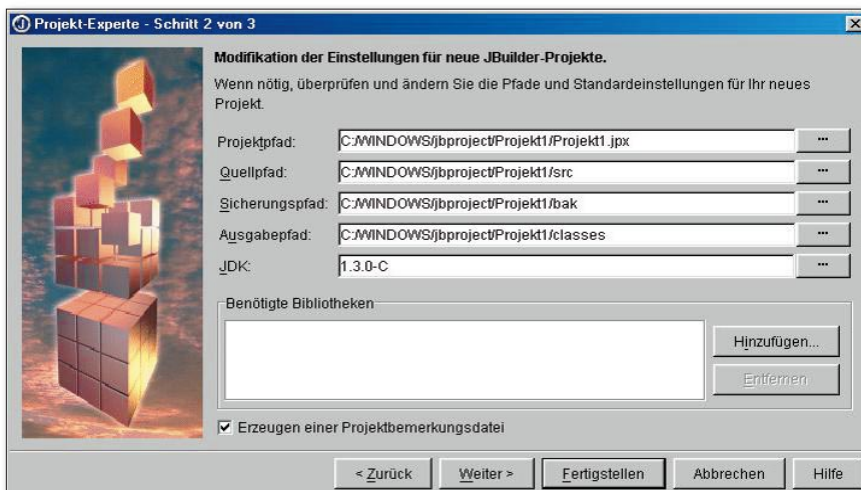
```
cd /tmp/jbuilder
```

ein. Anschließend starten Sie das Installationsprogramm durch

```
./install_linux
```



ANLEGEN EINES NEUEN Projekts im JBuilder ist einfach.



IN DIESEM DIALOG können Sie weitere Bibliotheken hinzufügen.

Jetzt startet der grafische Installer. Der erste Dialog lässt Sie die Sprache wählen. Zur Auswahl stehen deutsch, englisch, spanisch und französisch. In aller Regel, sofern Ihr Linux-System korrekt auf "deutsch" eingestellt ist, ist die Default-Auswahl "deutsch". Wählen Sie die von Ihnen bevorzugte Sprache, und klicken Sie auf "OK". Daraufhin öffnet sich ein Installationsassistent, wie Sie ihn von Windows her kennen. Die Bedienung funktioniert wie gewohnt. Schaltfläche "Weiter", um den nächsten Dialog anzuzeigen, und "Zurück", um eine Stufe in der Installation zurückzugehen. Nach dem Begrüßungsschirm, erscheint die Lizenzvereinbarung von JBuilder. Wenn Sie diese akzeptieren, klicken Sie auf die entsprechende Optionsschaltfläche. Die Installation können Sie nur nach dem Akzeptieren der Lizenz fortsetzen.

Als nächstes müssen Sie das Verzeichnis angeben, in das JBuilder installiert werden soll. Sie können hier die Vorgabe übernehmen. Wollen Sie das nicht, geben Sie über die über die Schaltfläche "Auswählen" ein anderes Verzeichnis an. Nach einem Klick auf "Weiter" startet die Installation. Nach einem kurzem Augenblick erscheint der Hinweis, dass die Installation abgeschlossen ist, und Sie können durch einen Klick auf "Fertig" das Setup beenden.

TIPP Sofern Sie die Standardvorgabe für das Zielverzeichnis übernommen haben, können Sie JBuilder unter Linux durch Eingabe des Kommandos

`/usr/local/jbuilder35/bin/jbuilder` starten.

Unter Windows verwenden Sie den neu hinzugekommen Eintrag im Menü der Start-Schaltfläche.

Nach dem ersten Start setzt Ihnen JBuilder einen Dialog vor, in dem Sie Ihren Namen und ggf. den Ihrer Firma eingeben sollen. Des weiteren ist es notwendig, dass Sie einen Aktivierungsschlüssel zum Freischalten von JBuilder eingeben. Klicken Sie auf "Add", und geben Sie anschließend Ihre Seriennummer und Ihren Authorisierungsschlüssel ein, die Sie über die Web-Seite von Inprise angefordert haben. Danach können Sie JBuilder starten. Bei den nächsten Starts wird nicht mehr nachgefragt. Die einzige Ausnahme ist, wenn Sie sich unter einer anderen User-Kennung einloggen und JBuilder starten. Hier wird wieder nach dem Aktivierungsschlüssel gefragt, da die Aktivierung auf Benutzerebene stattfindet.

Der Versuch JBuilder ohne Aktivierungsschlüssel zu verwenden ist zum Scheitern verurteilt. Sie müssen sich also den kostenlosen Schlüssel von der Web-Seite von Inprise holen.

■ Pilotprojekt

Um ein neues Projekt anzulegen, klicken Sie auf den Menüpunkt Datei / Neues Projekt. Im ersten Dialog legen Sie den Namen des Projekts und der verwendeten Verzeichnisse fest. Das "Stammverzeichnis" ist dabei das übergeordnete Verzeichnis, in dem das "Projektverzeichnis" angelegt wird. Die anderen Ordner für Quelltexte, Sicherungskopien und Ausgabe liegen im sogenannten Projektverzeichnis.

TIPP Wenn Sie häufiger ähnlich gestaltete Projekte anlegen, können Sie unter **Projekt für Vorgaben** ein anderes, bereits existierendes Projekt angeben. Dieses Projekt wird als Vorlage oder Schablone für das neue verwendet.

Wenn Sie Ihr Projekt benannt und die Verzeichnisnamen ggf. angepasst haben, klicken Sie auf *Weiter*. Jetzt haben Sie die Gelegenheit die Pfade als absolute Angaben zu bearbeiten. Außerdem können Sie zum Projekt weitere Bibliotheken (zum Beispiel JacORB für die CORBA-Unterstützung) hinzufügen.

Im dritten Schritt machen Sie weitere Angaben über das Projekt. Daraus wird eine Projektinformationsdatei als Beschreibung generiert. Außerdem können diese Informationen in neue von JBuilder automatisch erzeugte Quelltexte als Kommentare übernommen werden.

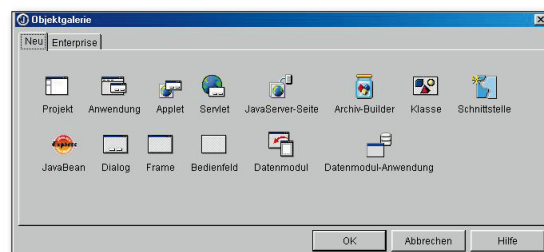
Ein Klick auf *Fertigstellen* und das neue Projekt wird erzeugt.

■ Neues braucht das Projekt

Neue Bestandteile, wie Anwendungen, Dialoge, Klassen oder Fenster, fügen Sie in das Projekt über Datei / Neu ein. Sie können auf diese Weise Rahmencode für fast jede Lebenslage erstellen lassen. Statt das grundlegende Gerüst für ein Applet oder eine Anwendung selbst einzutippen, verwenden Sie diese "Objektgalerie".

Erstellen Sie auf diese Weise eine neue Anwendung. Der Rahmencode, der für die Anwendung generiert wird, enthält bereits einen Frame. Ein Frame ist die Grundlage eines neuen Fensters. Sie erhalten durch die "Anwendung" nicht nur den Rahmencode zur Initialisierung Ihres Programms, sondern gleich ein Hauptfenster zur weiteren Ausgestaltung nach Ihren Wünschen mitgeliefert.

Sie können im folgenden Dialogfeld eine ganze Reihe von Eigenschaften angeben. Sie bestimmen in welches Pack-



NEUE OBJEKTE INS Projekt aufnehmen.



DIE TÜRME VON HANOI in alphanumerischer Aktion.

age die neue Klasse aufgenommen wird und wie sie heißen soll.

Im zweiten Schritt bestimmen Sie weiter, welche Elemente das Hauptfenster haben soll. Sie können bereits an dieser Stelle eine Menüzeile oder eine Statuszeile aufnehmen. Für das folgende kleine Beispiel ist nichts dergleichen wichtig.

Nach dem Erzeugen der Anwendung, sehen Sie den generierten Rahmencode. Am oberen Rand des Projektfensters befindet sich ein Register. Über die Reiter können Sie einzelne Dateien auswählen, um so den Frame oder die Anwendung selbst bearbeiten zu können. Wählen Sie den Frame.

Am unteren Rand des Fensters befindet sich ein Register, in dem Sie *Design* wählen können. Ein Klick auf diesen Reiter und Sie wechseln in den Designer von JBuilder. In dieser Ansichtsform können Sie Oberflächenelemente wie Eingabezeilen oder Schaltflächen per Point & Click aus der Objektleiste über dem (noch) leeren Frame einfügen. Zum Quelltext wechseln Sie über den Reiter *Quelle* zurück.

JBuilder arbeitet in beide Richtungen. Änderungen, die Sie im Quelltext vornehmen, werden automatisch in die Design-Ansicht übernommen und umgekehrt das gleiche. Auf diese Weise können Sie sehr flexibel sowohl mit dem Java-Code, als auch einer komfortablen Visualisierung arbeiten.

Wenn Sie schon einmal mit einem RAD-Tool (RAD = Rapid Application Development), wie Delphi oder Visual Basic, gearbeitet haben, werden Sie das Verhalten des Designers seltsam finden. Versuchen Sie eine Schaltfläche aus *Swing* oder *AWT* einzufügen, wird diese Schaltfläche nicht dort angezeigt, wo Sie hingeklickt haben.

Das liegt daran, dass in Java jeder Frame ein "Layout" erhalten kann. Ein Layout legt fest, wie die Oberflächenelemente in den Frame eingefügt werden. Per Standard wird für einen Frame von JBuilder *standard-layout* oder *BorderLayout* festgelegt.

Klicken Sie auf den Frame, und Sie sehen seine Eigenschaften in der Darstellung am rechten Rand. Suchen Sie dort die Eigenschaft *layout*. Wenn Sie auf den nach unten weisenden Pfeil am rechten Rand des Eingabefeldes klicken, wird eine Auswahl der verfügbaren Layouts dargestellt. Wenn Sie *XYLayout* wählen, können Sie die Oberflächenelemente im Frame frei positionieren.

Am linken Rand finden Sie eine Übersicht über das Projekt. Im oberen Bereich sind die Dateien aufgeführt. Im unteren Teil befinden sich die Bestandteile des gegenwärtig angezeigten Objekts. Nachdem der neue Frame ausgewählt ist, können Sie in diesem unteren Teil die einzelnen Komponenten auflisten.

■ Oberflächlichkeiten

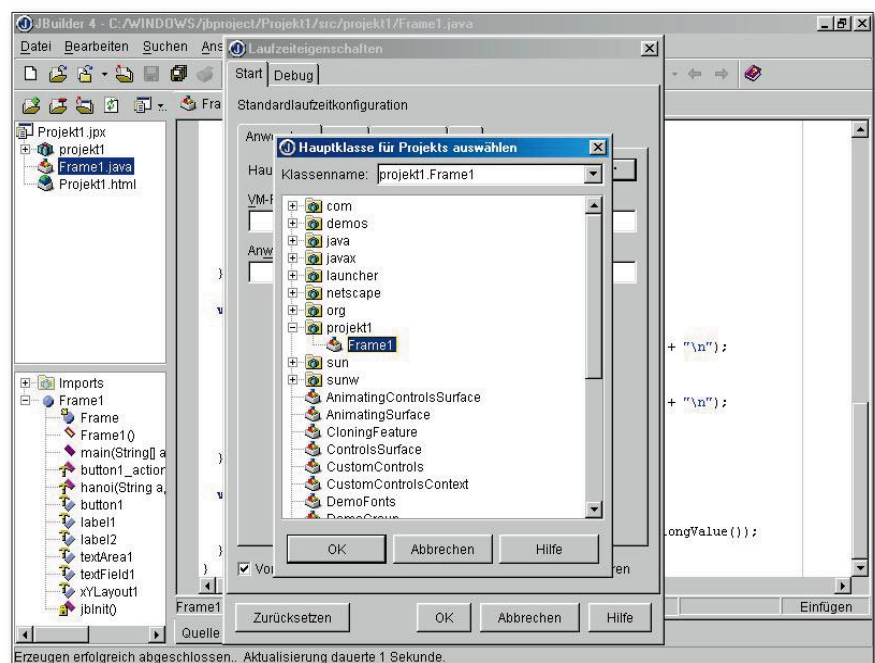
Als kleines Beispiel wollen wir die "Türme von Hanoi" implementieren. Näheres zu diesem klassischen Problem, mit dem sich jeder Informatiker einmal konfrontiert sieht, finden Sie im

gleichnamigen Kasten. Unser kleines Programm soll die Eingabe der Scheibenanzahl ermöglichen und beim Klick auf eine Schaltfläche die Schiebевorgänge berechnen. Die Ergebnisse sollen dabei in einem scrollbaren Feld als Anweisungen der Art "Scheibe von X nach Y" ausgegeben werden.

Die Komponenten, die wir zum Aufbau der Oberfläche verwendet wollen, stammen aus dem Reiter *AWT* des Designers. Schlagen Sie dieses Register im Designer auf. Fügen Sie zunächst zwei Label in den Frame ein. Eines soll die Beschriftung für das Eingabefeld und das andere für die Ergebnisliste aufnehmen. Die Texte der Label ändern Sie über den Eigenschaftendialog am rechten Bildschirmrand. Die Eigenschaft, die hierfür zuständig ist, heißt *text*.

Anschließend fügen Sie eine Schaltfläche (*Button*) ein. Diesen beschriften Sie über die Eigenschaft *label* mit "Starten". Dieser wird den Berechnungsvorgang starten.

Last but not least benötigen wir ein geeignetes Feld zur Ausgabe der Ergebnisliste. Sehr gut geeignet ist ein *TextArea*, das zur Eingabe von mehreren Zeilen dient. Fügen Sie dieses in den Frame ein. Die Eigenschaft *text* setzen Sie auf einen leeren String, denn sinnvolle Angaben lassen sich hier zwangsläufig noch nicht machen. Damit im Programm später die Ergebnisliste nicht verändert werden kann, setzen Sie außerdem die Eigenschaft *editable* auf *false*.



BEIM STARTEN DER ANWENDUNG müssen Sie die Hauptklasse angeben.



Mit Leben erfüllt

Nun steht die Oberfläche unseres Programms bereits. Es ist bislang nur eine leere Hülle ohne Leben. Es ist an der Zeit den Designer zu verlassen und auf die Quelltextebene zu wechseln. Klicken

Sie auf den Reiter **Quelle**. Zunächst benötigen Sie den Hanoi-Algorithmus.

Diesen implementieren Sie am einfachsten durch eine Methode, die sich rekursiv aufruft. Die Implementierung finden Sie in Kasten "Der Hanoi-Algorithmus". Tragen Sie diese Zeilen in die

Klassendefinition des Frames ein. Jetzt müssen Sie dem Programm noch begreiflich machen, dass es den Algorithmus starten soll, wenn auf die Schaltfläche geklickt wird. Wechseln Sie hierzu noch einmal in den Designer. Dort angekommen klicken Sie doppelt auf die Schaltfläche im Frame. JBuilder springt zurück in den Quelltexteditor und hat bereits den Rahmencode für die Methode zur Ereignisbehandlung generiert.

In diese Methode schreiben Sie zwei Anweisungen hinein:

```
textArea1.setText("");
hanoi("A", "B", "C",
(new Integer(textField1.
getText()).longValue()));
```

Die erste Anweisung löscht den Textinhalt des TextArea bzw. setzt ihn auf einen leeren String. Danach wird die Methode `hanoi()` aufgerufen. Die ersten drei Parameter sind einfach Strings, welche die Stäbe bezeichnen. Sie könnten hier auch "Nr.1" bis "Nr.3" oder "Stab A" bis "Stab C" oder was immer Ihnen gefällt eintragen. Diese Angaben werden statt der "X" und "Y" in der Meldung "Scheibe von X nach Y" eingesetzt.

Der vierte und letzte Parameter des Methodenaufrufs ist die Scheibenanzahl, die aus dem Textfeld entnommen wird. `textField1.getText()` liefert zunächst den Inhalt des Eingabefeldes als String zurück. Dieser String wird dem Konstruktor der Klasse `Integer` übergeben, woraus direkt die Konvertierung in einen Integer-Wert erfolgt. Dieses neue Integer-Objekt liefert über die Methode `longValue()` den Wert als `long` zurück, wie ihn die Methode `hanoi()` erwartet.

Hinein ins Getümmel

Jetzt ist der große Moment gekommen. Es heißt gespannt warten, was beim Starten des Programms passiert.

Speichern Sie zunächst das Projekt über **Datei / Alles speichern** ab. Um das Programm zu starten, verwenden Sie **Start / Projekt ausführen**.

Wenn das Programm läuft, geben Sie eine Scheibenanzahl ein, und klicken Sie auf **Starten**. Ist Ihnen kein Tippfehler unterlaufen, erhalten Sie jetzt eine Anweisungsliste, wie Sie das Rätsel lösen.



Achtung: Je höhere Zahlen Sie eingeben, desto länger dauert es, bis Sie ein Ergebnis erhalten.

DIE TÜRME VON HANOI

Das Problem der "Türme von Hanoi" beruht auf einer alten Sage. Diese besagt, dass die Mönche in einem Brahma-Tempel in der Nähe von Hanoi an der Lösung eines Puzzles arbeiten. Die Lösung des Rätsels ist direkt an die Existenz der Welt geknüpft. Sowie es nämlich gelöst wird, sei das "Ende aller Dinge", also der "Jüngste Tag" gekommen. Das Rätsel besteht aus einer Grundplatte, auf die drei Stäbe – A, B und C – senkrecht aufgesteckt sind. Auf diese Stäbe werden in der Mitte durchbohrte runde Platten in unterschiedlichen Größen gesteckt. Jede Größe kommt dabei nur einmal vor. Die Anzahl der runden Platten ist mit einer festen Zahl n bestimmt.

In der Startsituation sind alle Platten auf Stab A aufgesteckt, so dass immer eine kleinere auf einer größeren liegt. Es entsteht ein Stufenkegel. Das Ziel ist es, die Scheiben von Stab A nach Stab C mit der geringsten Anzahl von Schritten zu transportieren. Unter Zuhilfenahme eines "Zwischenlagers" auf Stab B kann in jedem Zug eine Scheibe verlagert werden. Die Einschränkung ist jedoch, dass immer nur eine kleinere auf einer größeren liegen darf.

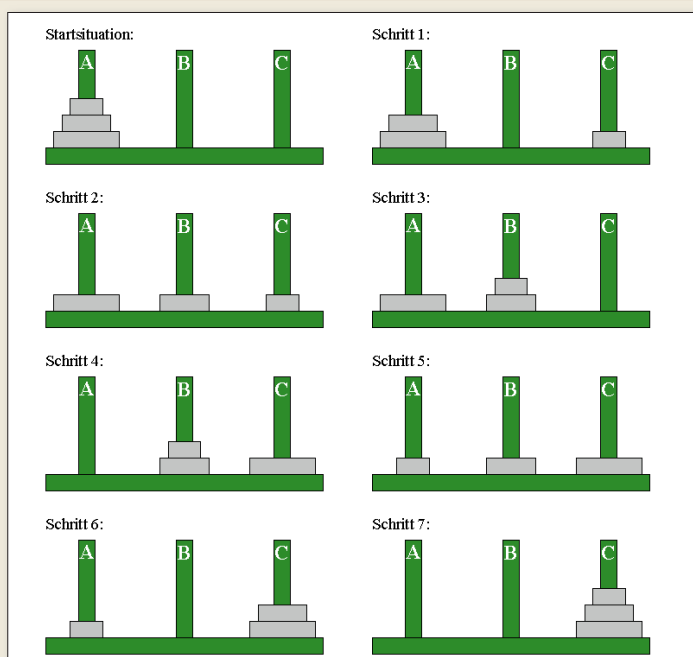
Das Prinzip beschreibt am besten ein Beispiel. Das Bild unten gibt die Züge für drei Scheiben wieder. Dieses Problem lässt sich rekursiv lösen. Der rekursive Algorithmus für n Scheiben lautet:

1. Schritt: Wenn n nicht eins ist: Versetze Stapel $n-1$ durch diesen Algorithmus vom Startstab auf den Zwischenstab.

2. Schritt: Versetze nun die frei liegende Scheibe vom Startstab auf den Zielstab.

3. Schritt: Versetze den Stapel $n-1$ durch diesen Algorithmus vom Zwischenstab auf den Zielstab.

Die geringste Anzahl an Schritten berechnet sich im übrigen nach der Formel $2^n - 1$. Falls Sie die Befürchtung hegen, dass die Mönche in Ihrem Kloster das Problem bald gelöst haben und die Welt untergeht, bevor Sie dieses Heft durchgelesen haben, können wir Sie beruhigen. Die Mönche arbeiten mit 64 Scheiben. Selbst wenn Sie pro Sekunde eine Scheibe bewegen könnten, würden Sie $2^{64} - 1$ Schritte bzw. Sekunden benötigen. Dies entspricht einem Zeitaufwand von $1,8447 \cdot 10^{19}$ Sekunden bzw. ca. $5,8494 \cdot 10^{11}$ Jahre. Oder anders gesagt: 585 Milliarden Jahre!



DIE TÜRME VON Hanoi, ein Musterbeispiel für Rekursion