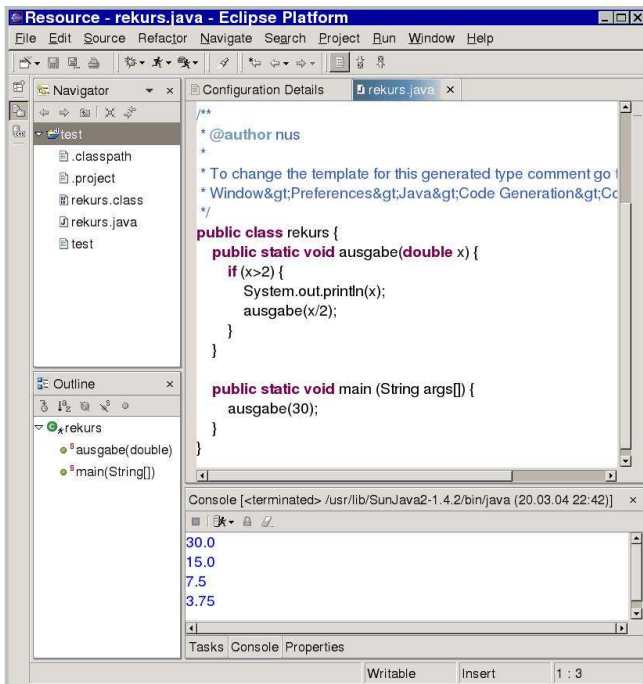


Eclipse

Eclipse wurde ursprünglich (1999) als JAVA-Entwicklungs-umgebung entworfen. IBM hat nach Erscheinen der Version 1.0 den gesamten Code als Open Source frei gegeben; zur Zeit ist die Version 3.0 erhältlich¹. Im Gegensatz zu anderen IDEs (z.B. SUNs NetBeans) besteht Eclipse aus einem relativ schlanken Kern, der um zahlreiche Plug-ins erweitert wird. Grundsätzlich kann Eclipse für die Projektentwicklung in verschiedenen Programmiersprachen verwendet werden – in diesem Text sprechen wir ausschließlich über die Verwendung in Zusammenhang mit JAVA-Klassen, Applikationen und Applets.

Nach dem ersten Start liefert die so genannte „Eclipse-Workbench“ ein „Welcome“-Dokument und einen CVS (*Concurrent Versions System*)-Dialog. Wir wählen die Ansichten des Dateimanagers, der JAVA-Objekte, den Editor und das Ausgabefenster:



Das Online-Hilfesystem

Eclipse wird mit einer umfangreichen Online-Hilfe ausgeliefert; und zwar läuft unter dem Port 32870 ein interner WebServer mit der Adresse <http://127.0.0.1>, der zahlreiche Hilfedokumente zur Verfügung stellt. Das Startdokument „help/index.jsp“ wird im Browser geöffnet, sobald man die aus dem Menüpunkt für die Hilfe den Eintrag „Help Contents“ auswählt:

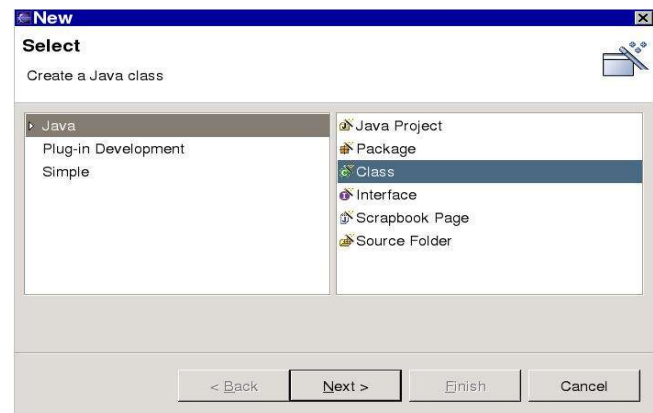


Die einzelnen Abschnitte behandeln detailliert verschiedene Arbeitsbereiche:

- Anleitung über das Arbeiten mit Eclipse
- Leitfaden zum Erstellen von JAVA-Anwendungen
- Handbuch über die Verwendung von Plug-ins
- Beschreibung der Java-Entwicklungs-Werkzeuge JDT
- Plug-in Entwicklungsumgebung PDE
- Leitfaden für FTP und WebDAV

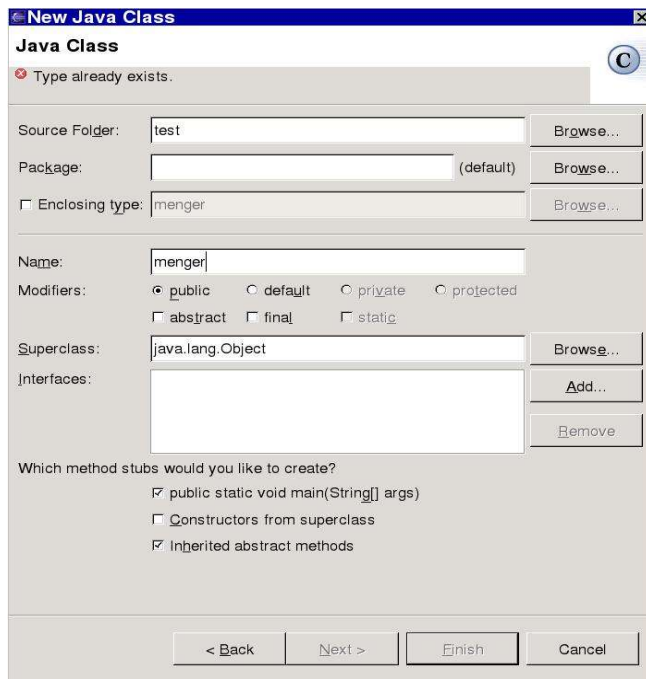
Eine neue Datei öffnen

Wählt man das Erstellen einer neuen Datei aus, so werden die Einträge „Project“ - „Folder“ - „File“ - „Other“ angezeigt. Entscheidet man sich für den Menüpunkt „Other“, so erhält den folgenden Dialog:



Wir wählen beispielsweise das Erstellen einer Java-Klasse aus, legen den Namen der Datei und die defaultmäßig erzeugten Methoden, z.B. die main() - Methode, fest:

¹ In diesem Dokument wird eine Eclipse-Version verwendet, die mit der aktuellen Linux-Distribution von SuSE ausgeliefert wird (Version 2.1.1, Build Juni 2003).



Nach dem Abschließen aller notwendiger Konfigurationen erhält man ein syntaktisches Gerüst für eine JAVA-Klasse, beispielsweise:

```

/*
 * Created on 20.03.2004
 *
 * To change the template for this generated file
go to
 * Window>>Preferences>>Java>>Code
Generation>>Code and Comments
 */

/**
 * @author nus
 *
 * To change the template for this generated type
comment go to
 * Window>>Preferences>>Java>>Code
Generation>>Code and Comments
 */
public class menger {

    public static void main(String[] args) {

    }

}

```

Ist der Programmcode vollständig, so wird die Datei gespeichert, kompiliert und ausgeführt.

Beispiel: Applets entwickeln

Im folgenden Beispiel soll ein so genanntes Fraktal in einem Applet ausgegeben werden. Dazu verwenden wir zunächst die Klasse `Turtle`, die zwei aus der LOGO-Programmierung bekannte Methoden zur Verfügung stellt: `fd(double strecke)` stellt den Zeichenstift in der aktuellen Richtung um die angegebene Anzahl von Pixel weiter, `rt(double winkel)` dreht die Zeichenrichtung um die angegebenen Winkelgrade:

Quelltext Turtle.java:

```

import java.awt.*;

class Turtle {
    double x;
    double y;
    double alpha;
    Container c;
    Graphics g;

    Turtle(Container ct, double x, double y) {
        c = ct;
        g = c.getGraphics();
        g.setColor(Color.black);
        this.x = x;
        this.y = y;
        alpha = 0;
    }

    public void fd (double strecke) {
        double aa = alpha * Math.PI / 180;
        double dx = strecke * Math.cos(aa);
        double dy = strecke * Math.sin(aa);
        g.drawLine((int) x, (int) y, (int) (x+dx),
(int) (y+dy));
        x = x+dx;
        y = y+dy;
    }

    public void rt (double winkel) {
        alpha = alpha - winkel;
    }
}

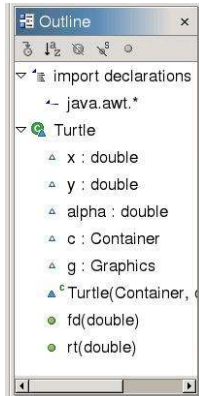
```

Der Konstruktor `Turtle(Container ct, double x, double y)` übernimmt die aktuelle Position (x/y) des Zeichenstifts und das Container-Objekt `ct`, das die gesamte Grafik enthält. Anschließend weist er der Klassenvariablen diese Grafik zu und übergibt die Stiftposition an die Klassenvariablen `x` und `y`. Der Winkel `alpha` erhält den Wert 0; die aktuelle Zeichenrichtung ist somit die Richtung der positiven x -Achse.

Die Methode `rt(double winkel)` aktualisiert gegebenenfalls den Winkel für die Zeichenrichtung; weil auf der Zeichenfläche die y -Achse nach unten weist, wird der angegebene Winkel subtrahiert um den Winkel entgegen dem Uhrzeigersinn zu addieren. Die Methode `fd(double strecke)` zeichnet eine Linie in dieser Richtung.

eclipse

Eclipse stellt die Klassenvariablen und Methoden im Outline-Fenster folgendermaßen dar:



Wir verwenden die Klasse `Turtle` im nächsten Beispiel `baum.java`, das einen fraktalen Baum darstellt. Ist die Klasse `Turtle` im gleichen Verzeichnis wie die Klasse `baum` gespeichert, kann der `import`-Befehl entfallen.

Quelltext `baum.java`:

```
import java.awt.*;
import java.applet.*;
// import Turtle.*;

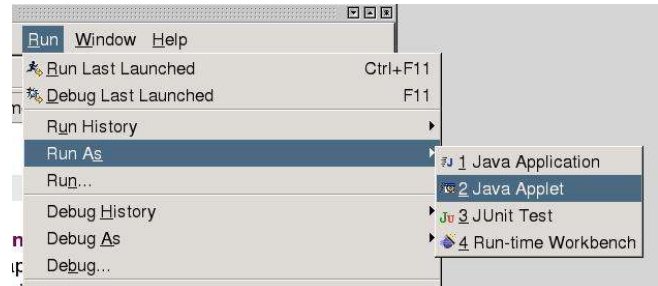
public class baum extends Applet {
    public void paint (Graphics g) {
        Turtle t = new Turtle(this, 380,450);
        t.rt(80);
        baumz(t,200,52,15);
    }

    public void baumz(Turtle t, double stamm,
                    double winkel, int ebene) {
        if (ebene > 0) {
            t.fd(stamm);
            t.rt(winkel+15);
            baumz(t, stamm/1.3, winkel, ebene - 1);
            t.rt(-2*winkel-15);
            baumz(t, stamm/2, winkel, ebene - 1);
            t.rt(winkel);
            t.fd(-stamm);
        }
    }
}
```

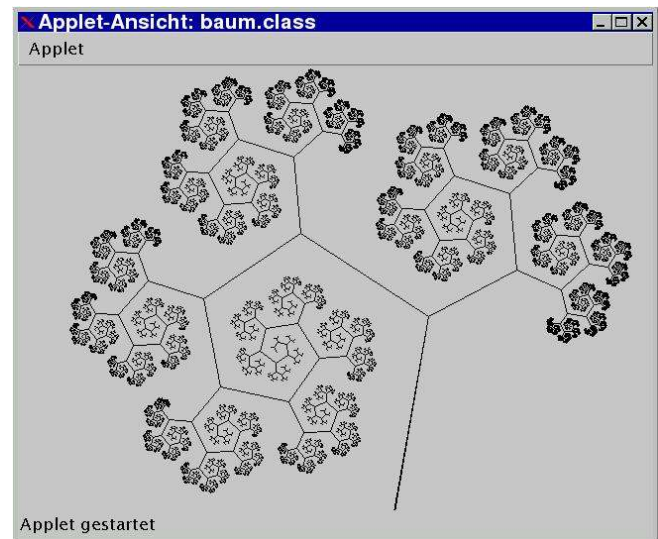
Die `paint()` - Methode erzeugt eine neue Instanz der Klasse `Turtle`, legt den Anfangswinkel für den Zeichenstift fest und ruft die rekursive Methode `baumz()` auf. In dieser ist das Wegfraktal formuliert: So lange die Zahl der Rekursionstiefe `ebene` größer als 0 ist wird zunächst ein kleines Stammstück gezeichnet, dann der Winkel für den nächsten Ast gesetzt und `baumz()` rekursiv aufgerufen. Anschließend wird der Winkel 2-fach verringert und `baumz()` für einen zweiten Zweig rekursiv aufgerufen. Nach dem Zurücksetzen des Winkels geht der Zeichenstift den ursprünglich gezeichneten Stamm

zurück – dies ist wichtig, damit der Zeichenstift nach dem (rekursiven) Zeichnen eines Astes wieder zum Verzweigungspunkt zurückgeschickt wird.

Das Applet wird aus dem „Run“-Menü gestartet:



Eclipse startet daraufhin einen Applet-Viewer, in dem das Applet angezeigt wird; im Allgemeinen wird man die Ausmaße dieses Fensters der Anwendung entsprechend „aufziehen“:



Aufgabe:

Erstelle ein ähnliches Fraktal, das das folgende Bild eines fraktalen Farnes ergibt:

