NetBeans Konfiguration und Verwendung
unter Windows XP/Vista
Version 2.0, April 2009

1 Distribution

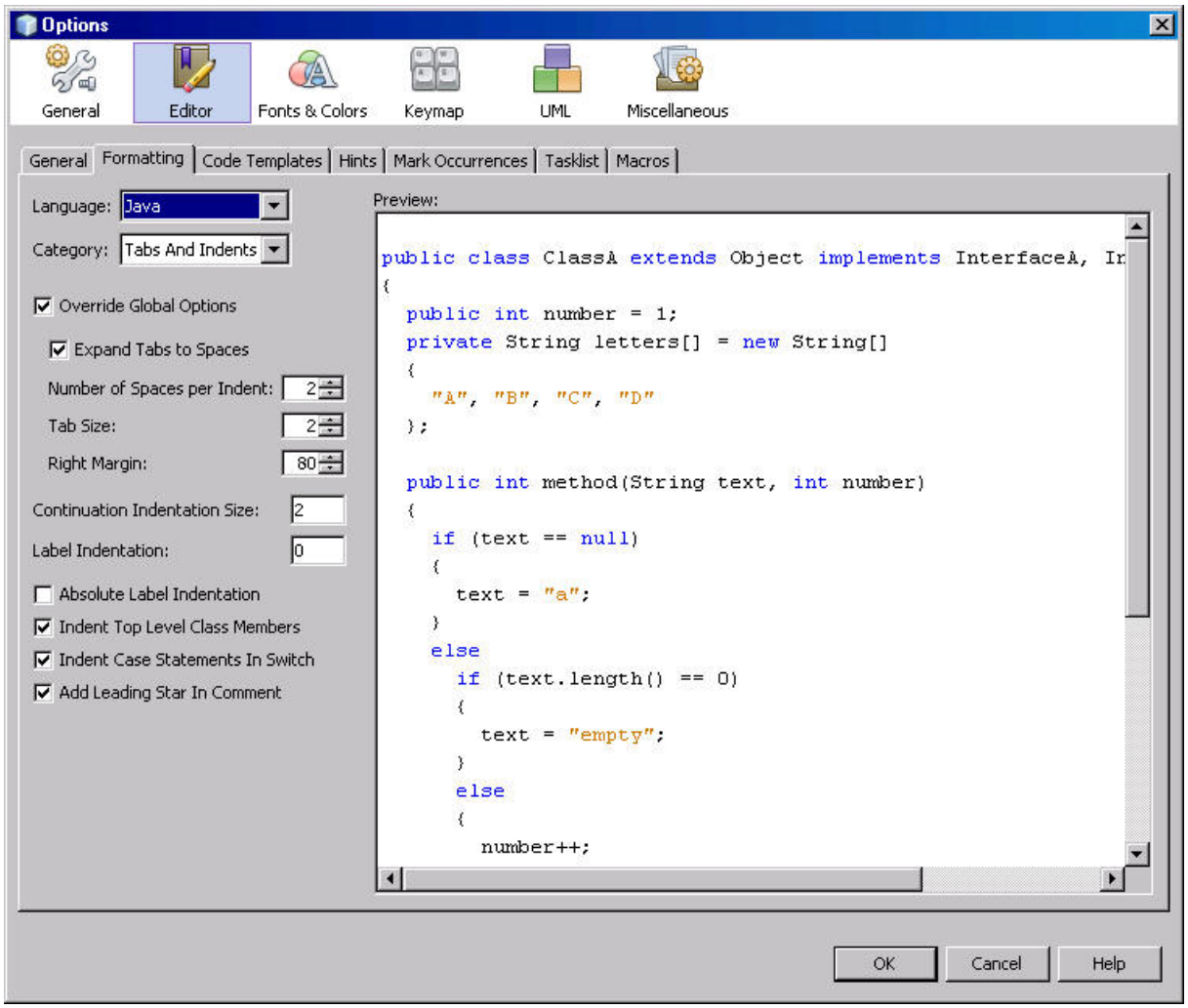
Download und Installation des JDK 6 von *http://java.sun.com/javase/downloads*. Download der dazugehörenden Java SE 6 Documentation von derselben Website. Download und Installation von Netbeans von *http://www.netbeans.org/downloads* (Bundle All).

2 Konfiguration

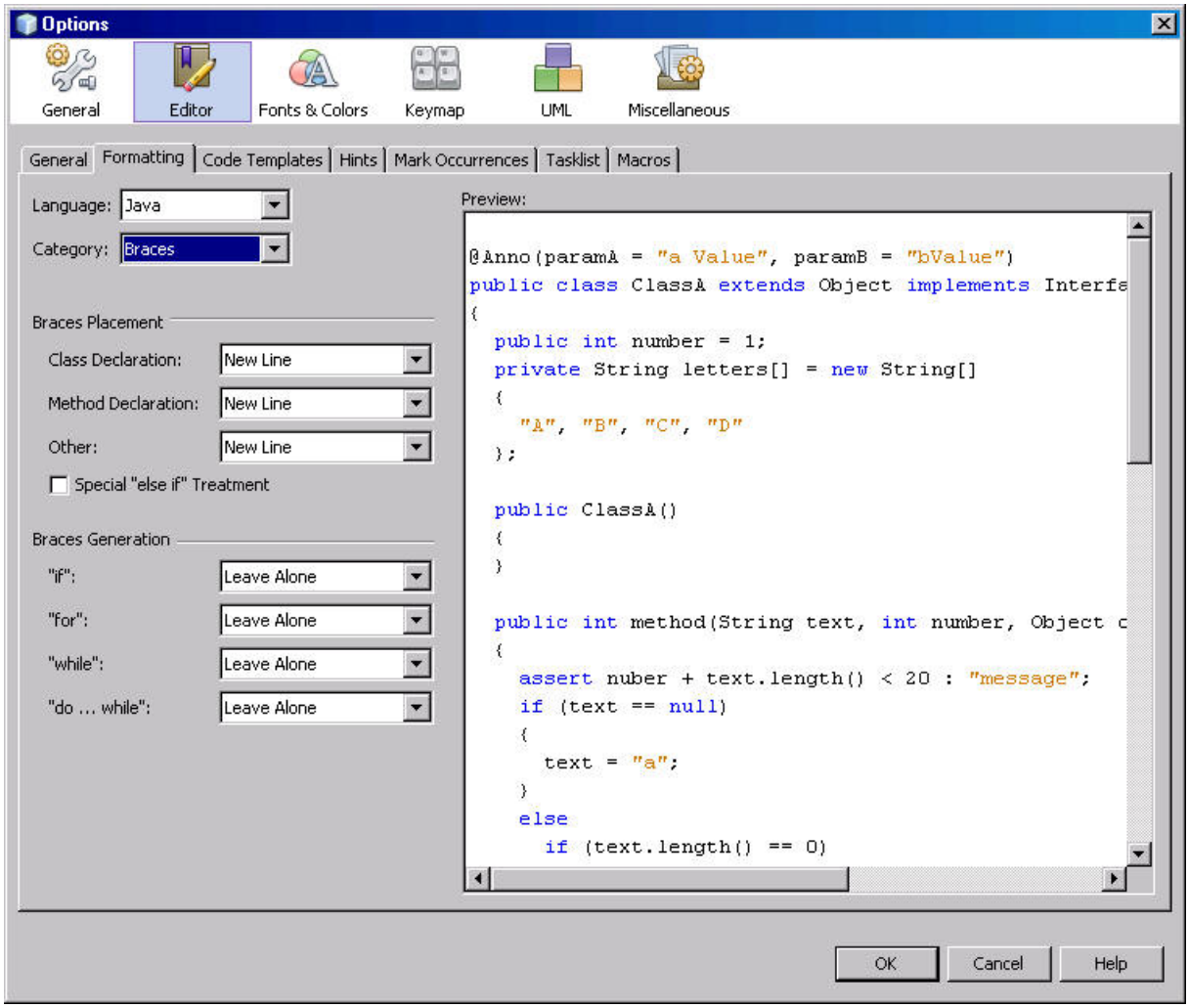
2.1 Die Format-Optionen setzen

Die folgenden Angaben beziehen sich auf die Standardformatierung auf www.java-online.ch und www.aplu.ch. Sie können individuell angepasst werden.

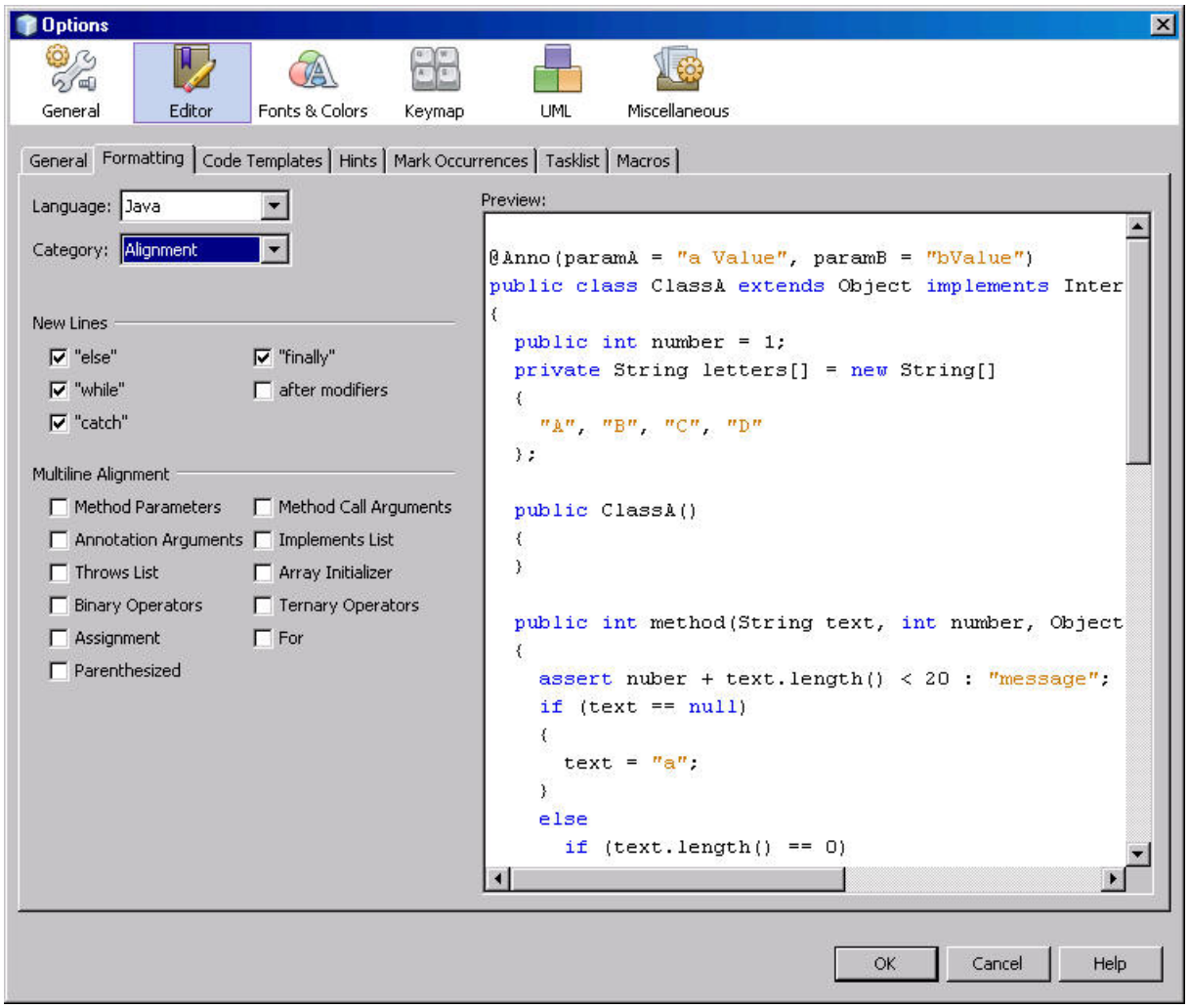
Tools | *Optionen* wählen, auf die Ikone *Editor* klicken und Register *Formatting* wählen. Unter Language *Java* auswählen.



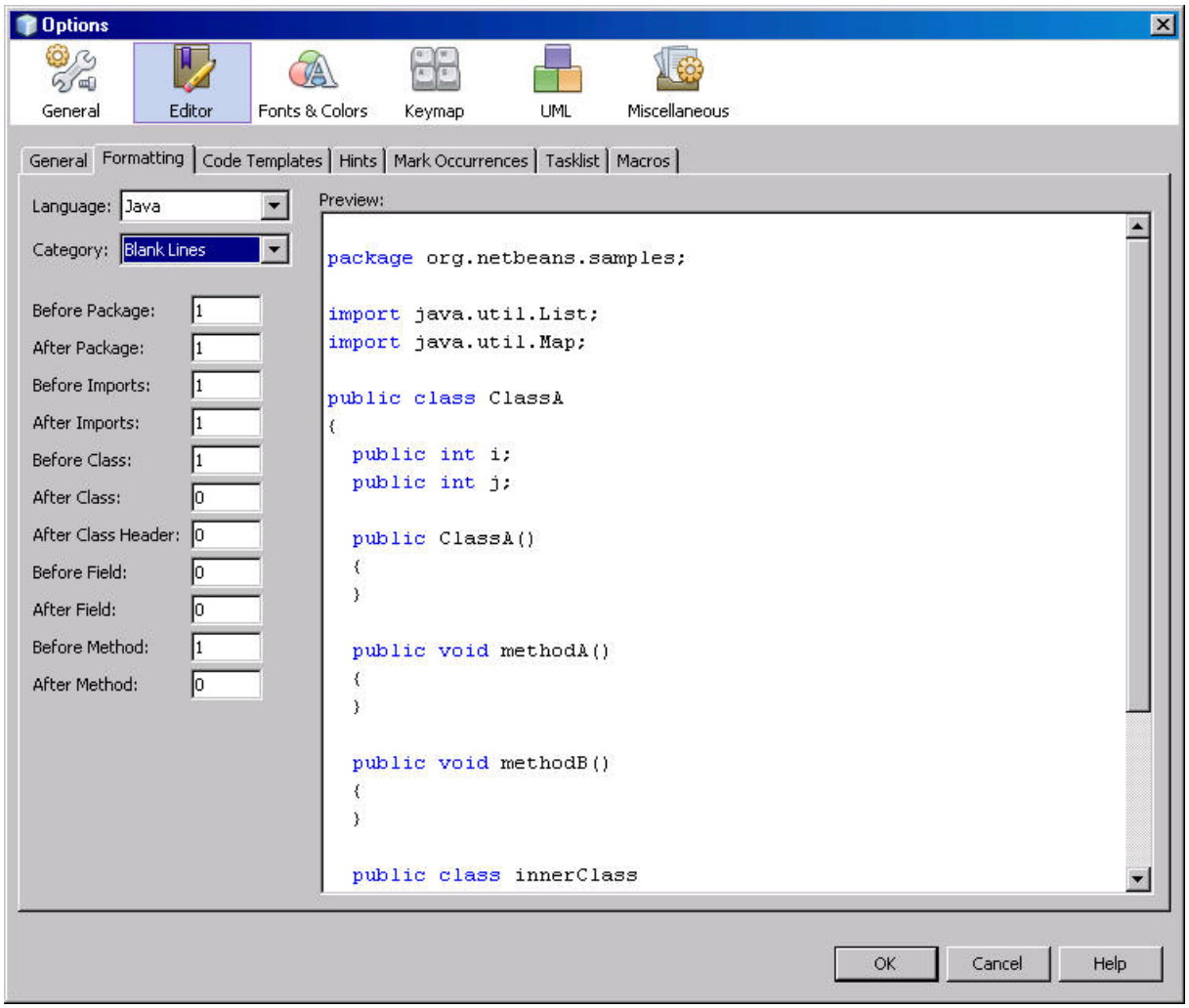
In Category *Tabs And Indents* den Eintrag *Number of Spaces per Idents*: 2 und T*abSize* 2 wählen. Unter *Braces* folgende Einstellungen wählen:



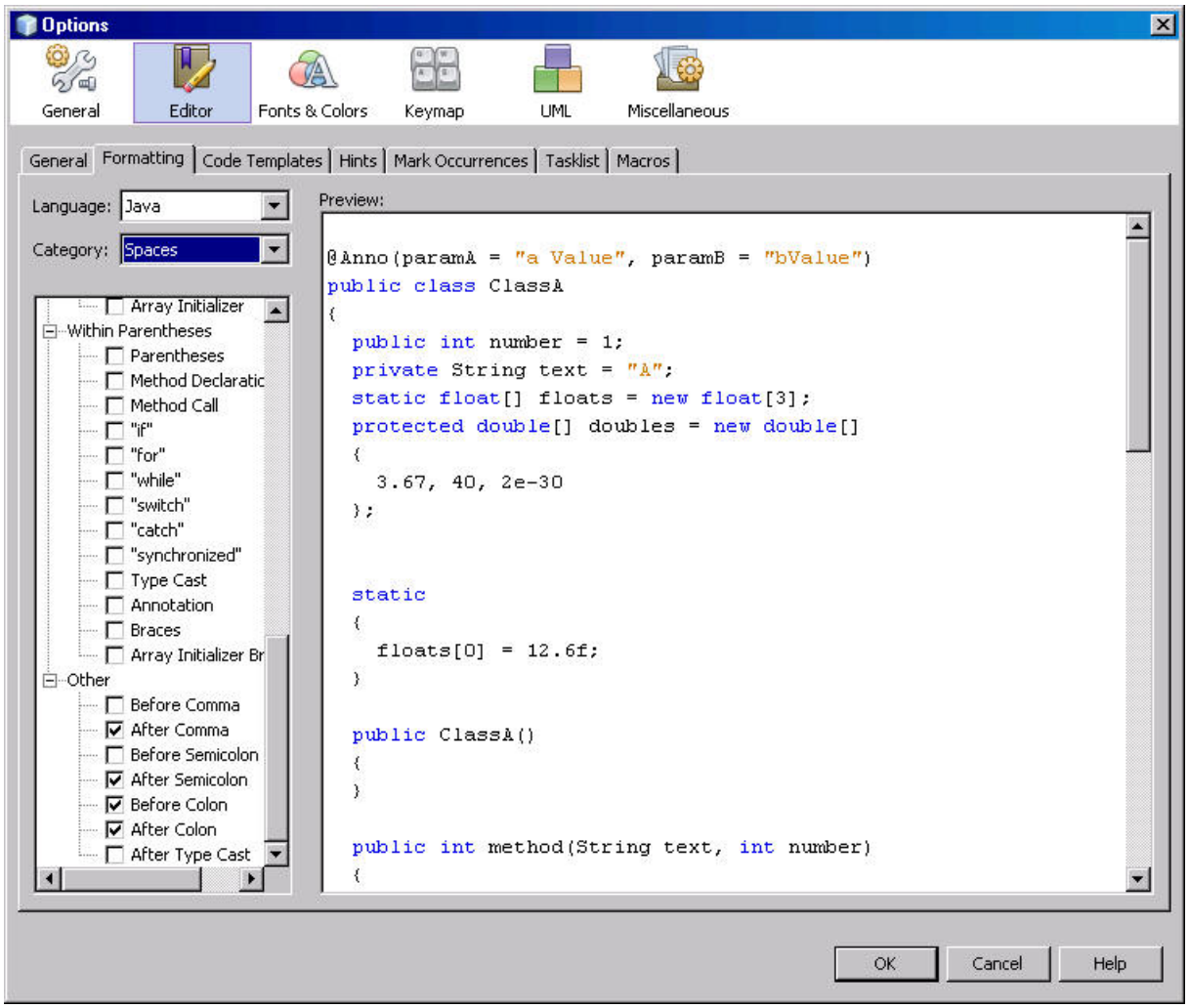
Unter *Alignments* folgende Einstellungen wählen:



Unter *Blank Lines* folgende Einstellungen wählen:



Unter *Spaces* nur die Einstellung *After Type Case* ändern:

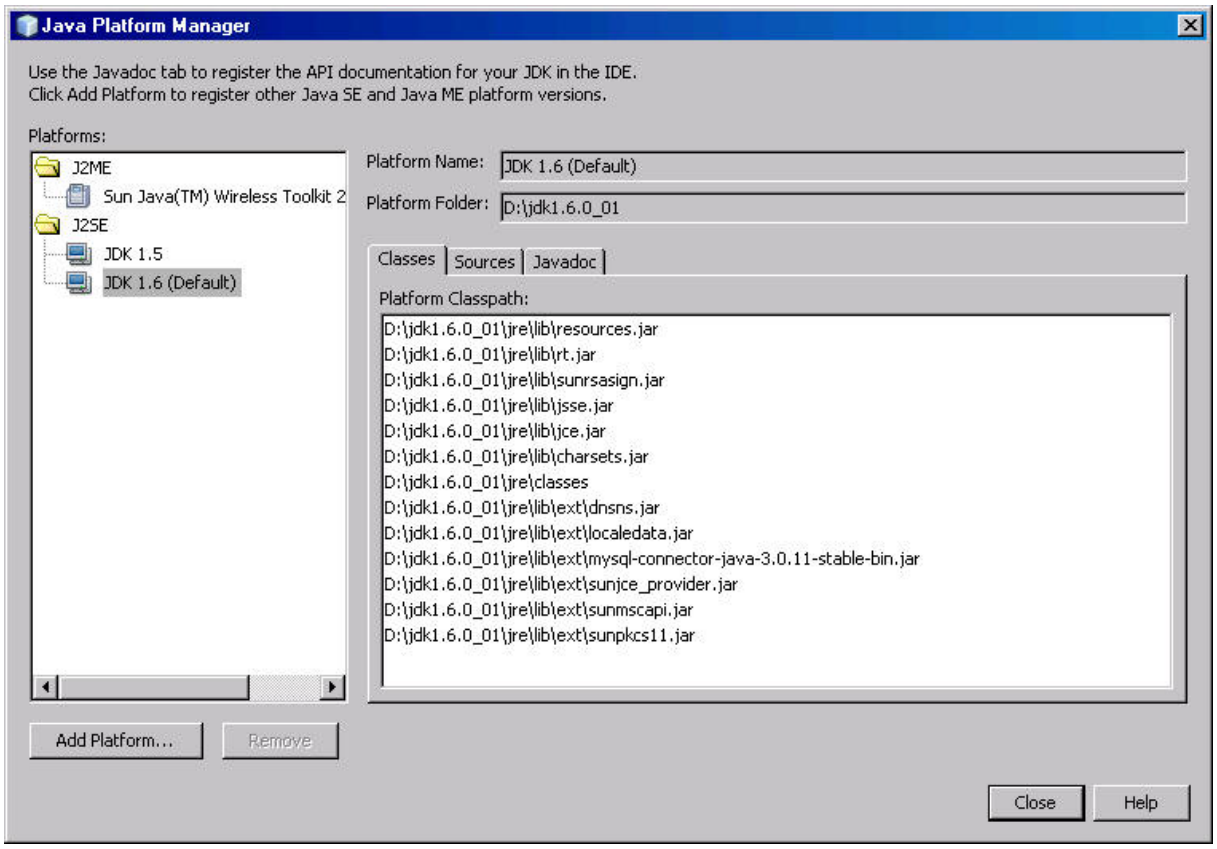


Mit OK bestätigen.

2.2. Installation der kontext-sensitiven Hilfe

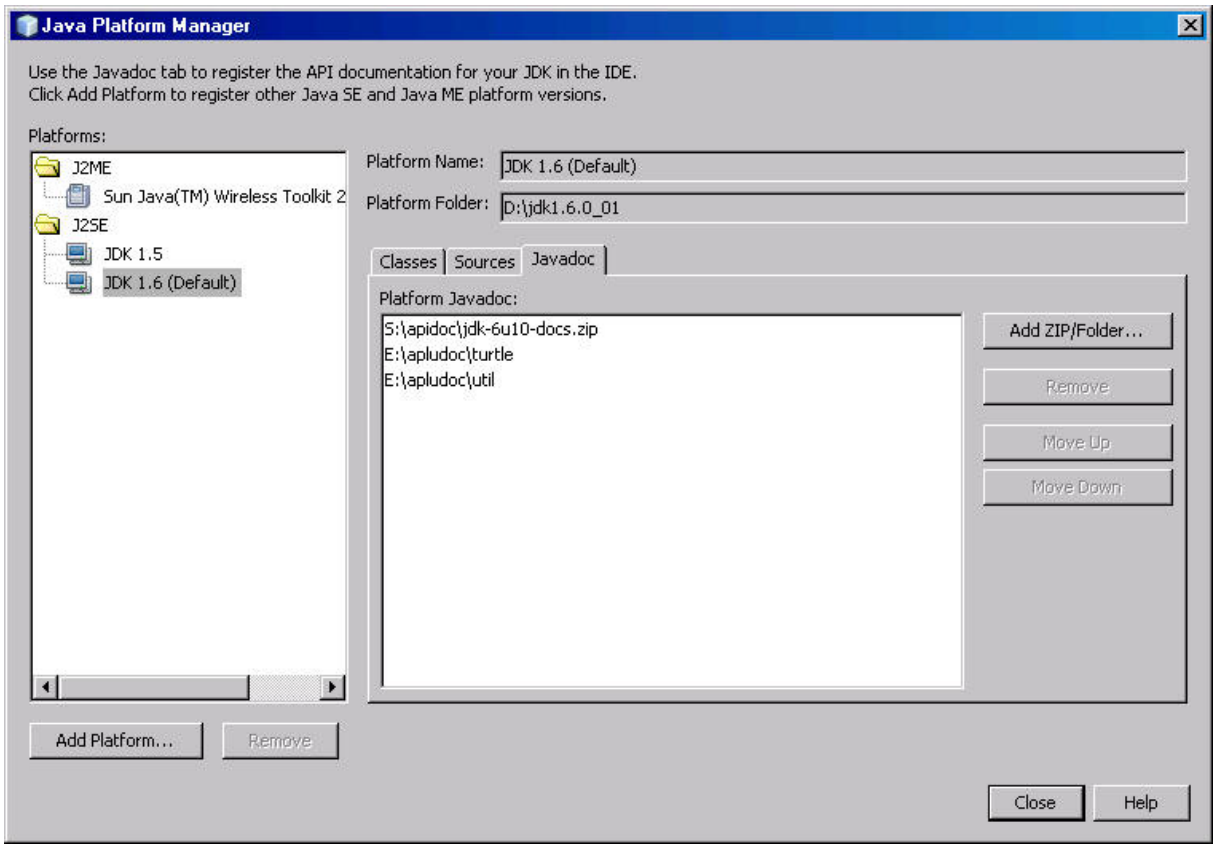
Von *www.aplu.ch/download* die Javadoc des Packages *ch.aplu.util* und *ch.aplu.turtle* downloaden. Je in ein eigenes Verzeichnis auspacken.

Tools | *Java Plattform Manager* wählen:



Auf Register *Javadoc* klicken. Auf *Add ZIP/Folder*... klicken und die ZIP-Datei der API-Documentation, sowie den Verzeichnisnamen aller zusätzlichen Javadocs hinzufügen, beispielsweise:

s:\apidoc\jdk-6u10-docs.zip: Java API doc
E:\apludoc\turtle : JavaDoc für das Package ch.aplu.turtle
E:\apludoc\util : JavaDoc für das Package ch.aplu.util



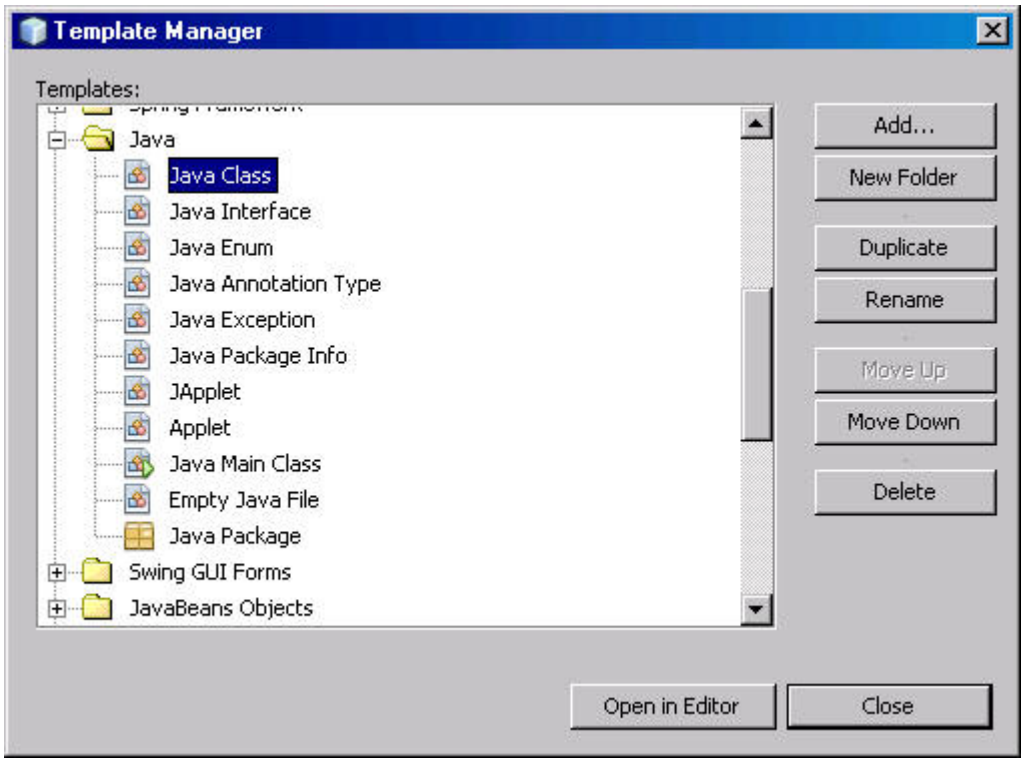
Close klicken.

Damit wird die kontext-sensitive Hilfe im Source-Editor eingerichtet: Setzt man den Cursor auf einen Klassennamen oder eine public Methode und drückt *Shift-F1*, so wird die Javadoc geöffnet.

2.3 Templates einrichten

Die Templates können den individuellen Bedürfnissen angepasst werden. Für Anfängerkurse werden die folgenden Templates empfohlen:

Tools | *Templates* wählen. Dort *Java* öffnen:



Button *Duplicate* drücken. Dann den Namen durch *GPanel* ersetzen und auf Button *Rename* klicken. Jetzt *Open in Editor* klicken und den Text wie folgt ändern:

```
// ${name}.java

<#if package?? && package != "">
package ${package};

</#if>
import ch.aplu.util.*;

public class ${name}
{
  private GPanel p = new GPanel();

  public ${name}()
  {

  }

  public static void main(String[] args)
  {
    new ${name}();
  }
}
```

Mit *Control-S* speichern. Dasselbe mit dem Template *Turtle* durchführen. Das folgende Template verwenden:

```
// ${name}.java

<#if package?? && package != "">
package ${package};

</#if>
import ch.aplu.turtle.*;

public class ${name}
{
  private Turtle t = new Turtle();

  public ${name}()
  {

  }

  public static void main(String[] args)
  {
    new ${name}();
  }
}
```

Dasselbe mit dem Template *Gidlet* unter *MIDP* durchführen. Das folgende Template verwenden:

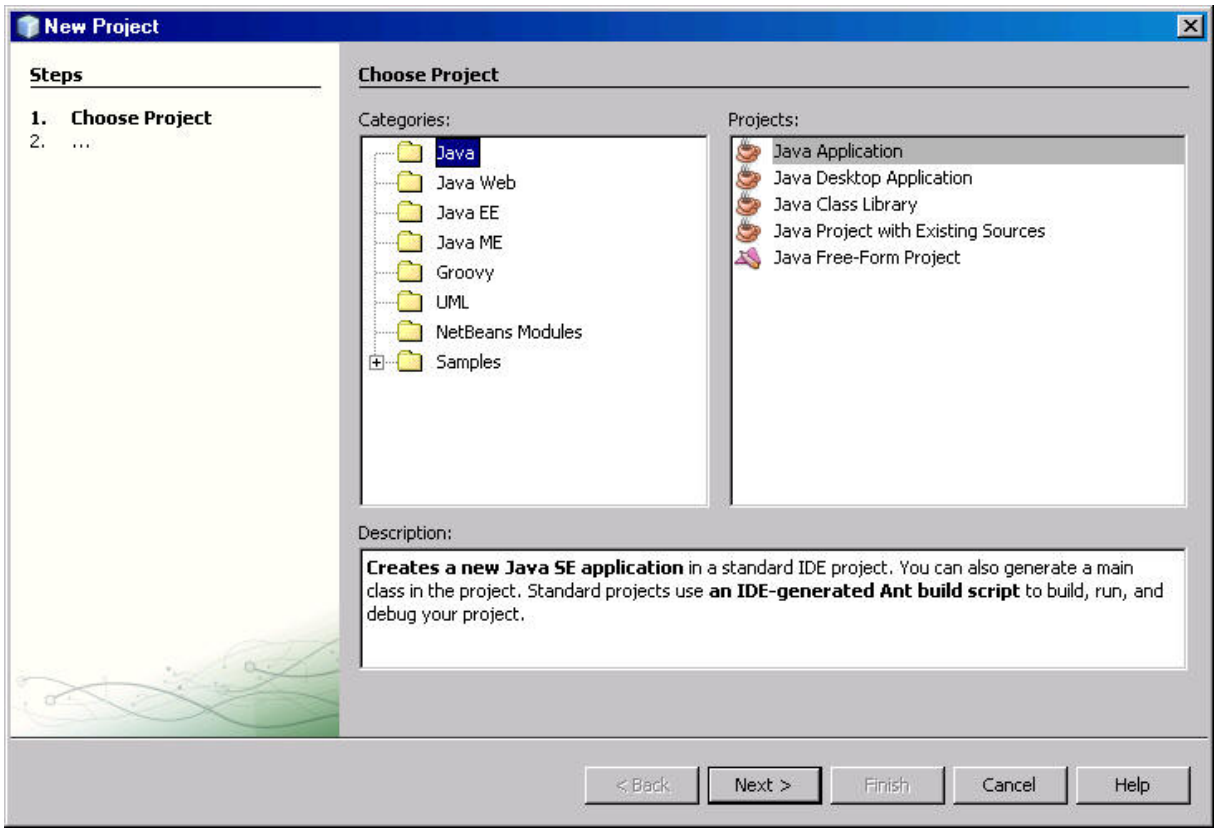
```
// ${name}.java

<#if package?? && package != "">
package ${package};

</#if>
import ch.aplu.gidlet.*;

public class ${name} extends Gidlet
{
  public void main()
  {

  }
}
```

3 Verwendung

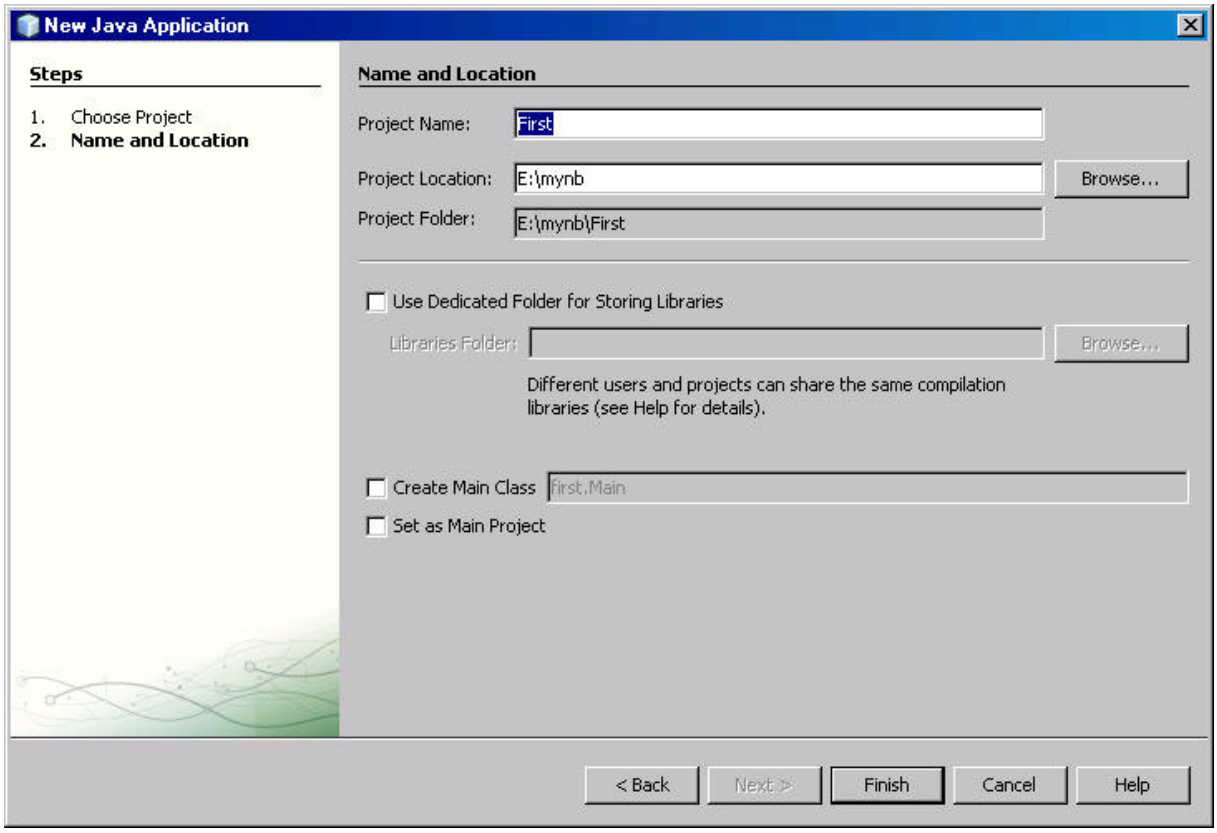
3.1 Applikation für J2SE

Um ein Programm für J2SE zu editieren, kompilieren und auszuführen, wie folgt vorgehen:

File | *New Project* und unter *Categories Java* und unter *Projects Java Application* wählen.



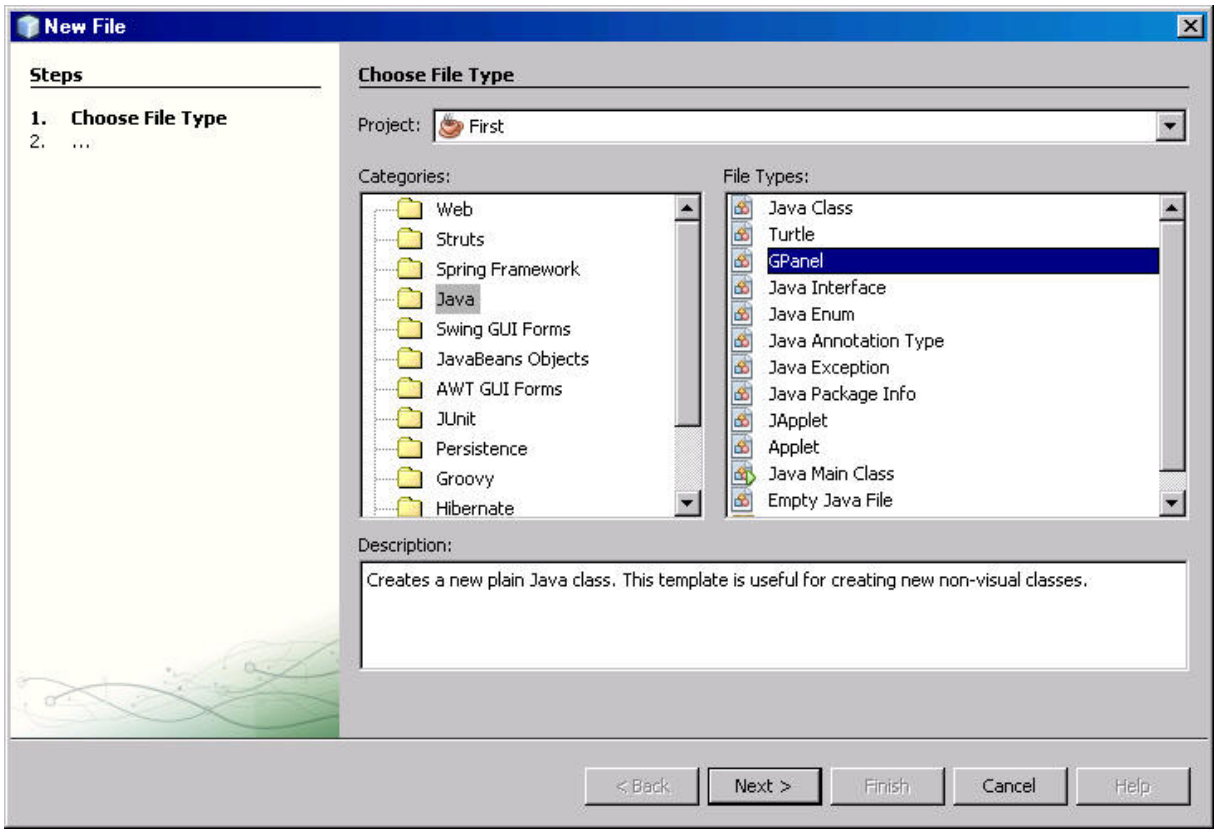
Auf *Next* klicken. Den Projektnamen, z.B. *First* und den Projektort, z.B. *E:\mynb*, eingeben.



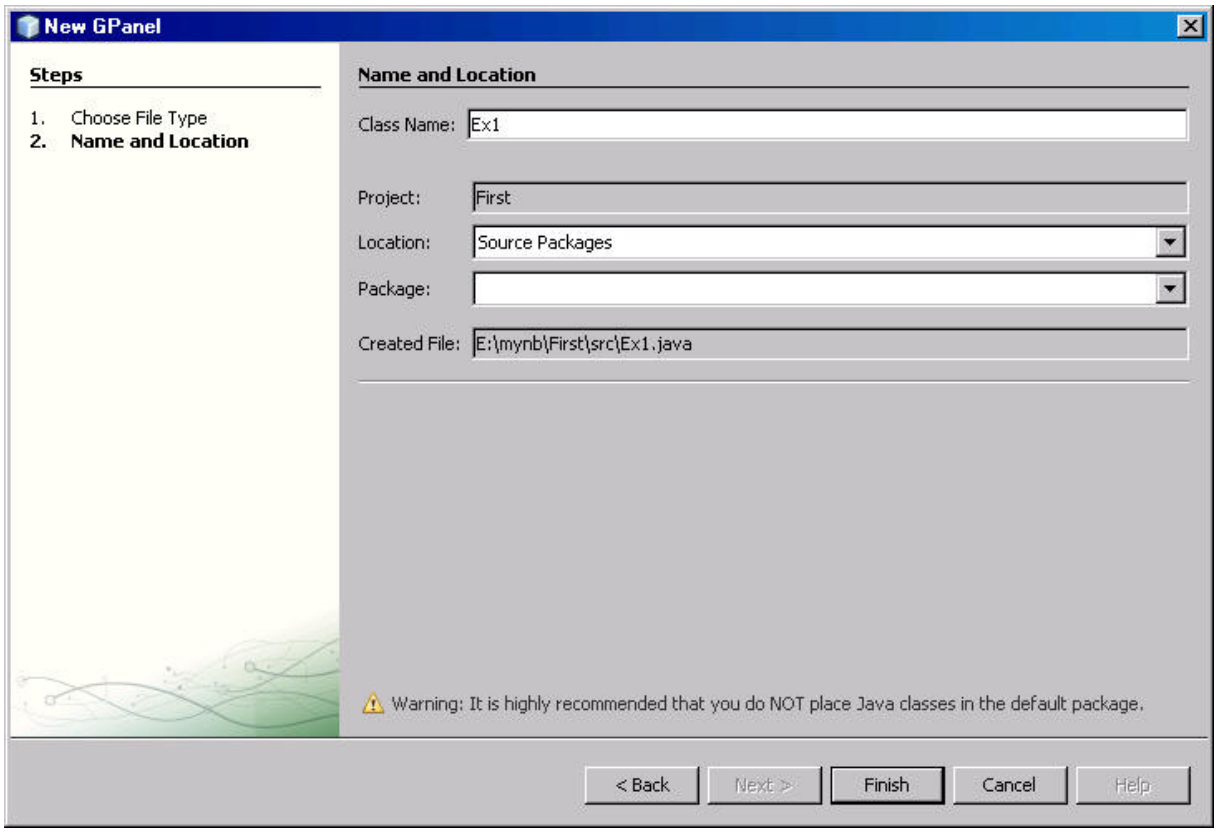
Create Main Class und *Set as Main Project* abwählen. *Finish* klicken.

Rechter Mausklick im Projektfenster auf *Libraries*. Dann *Add Add JAR/Folder* wählen. Die Datei *aplu5.jar* angeben, die man von *www.aplu.ch/download* heruntergeholt hat.

Rechter Mausklick im Projektfenster auf *First* | *Source Packages* und *New Other* auswählen. Unter *Categories Java* und *File Types GPanel* wählen



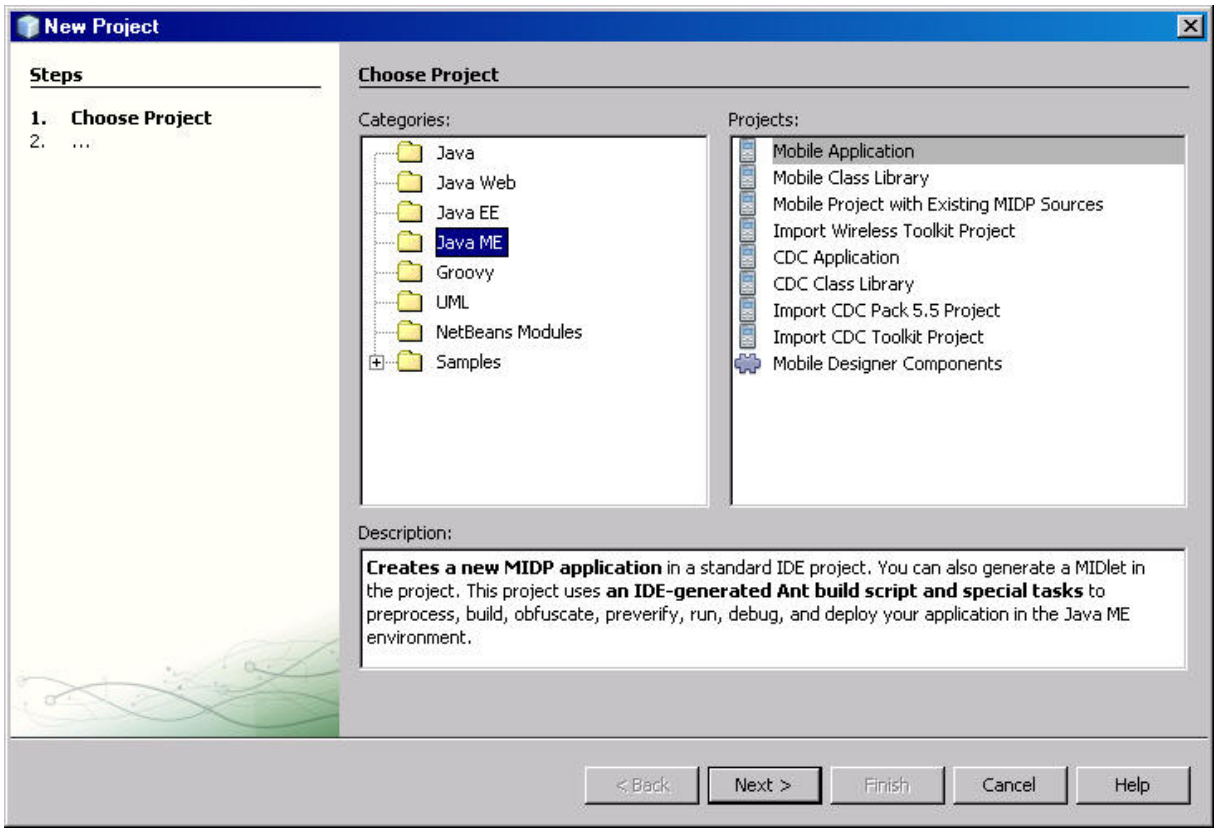
Next klicken. Unter *Class Name* einen Klassennamen eintippen, z.B. *Ex1*.



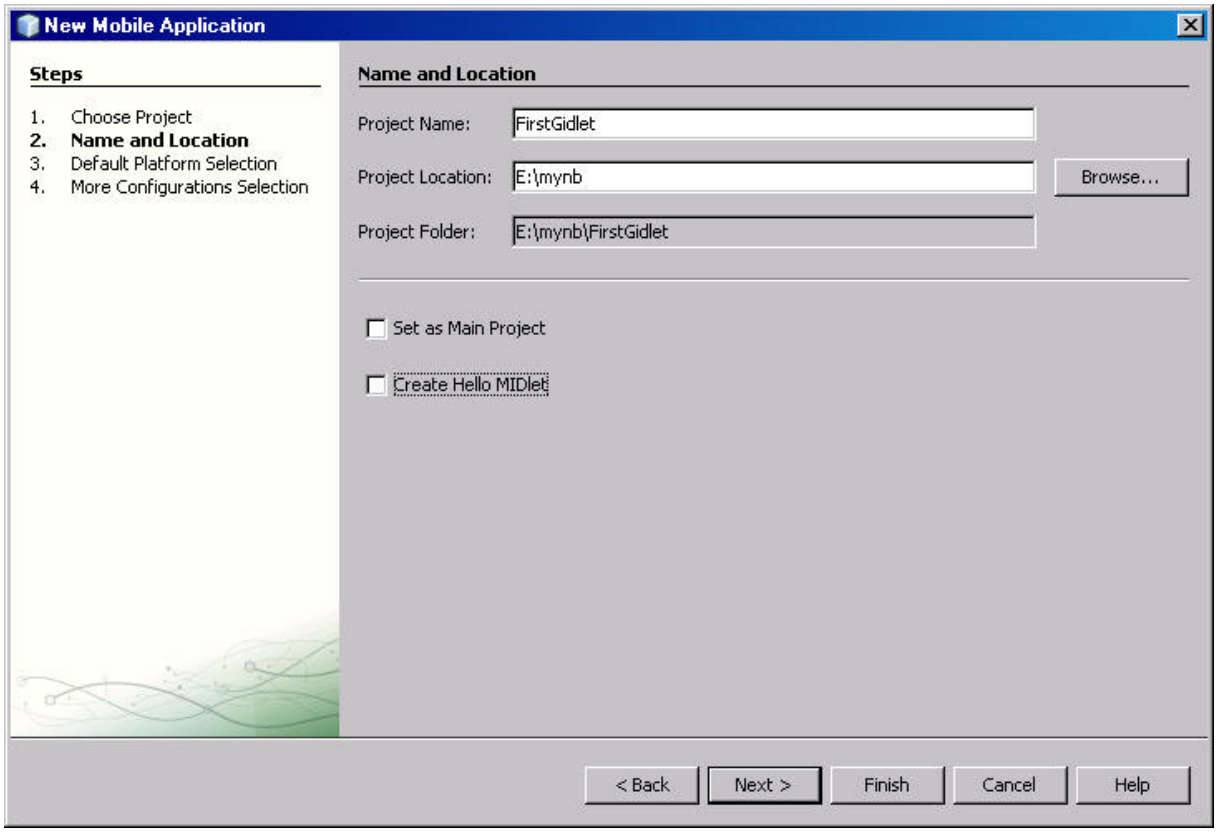
Finish klicken. Das Editorfenster mit dem Template wird geöffnet. Im Editorfenster sollten keine Fehler angezeigt werden. Im Projektfenster rechter Mausklick auf *Ex1.java* und *Run File* auswählen. Die Source wird compiliert und ausgeführt.

3.2 Applikation für J2ME (MIDlet, Gidlet)

File | New Project und unter *Categories Java ME* und unter *Projects Mobile Application* wählen.



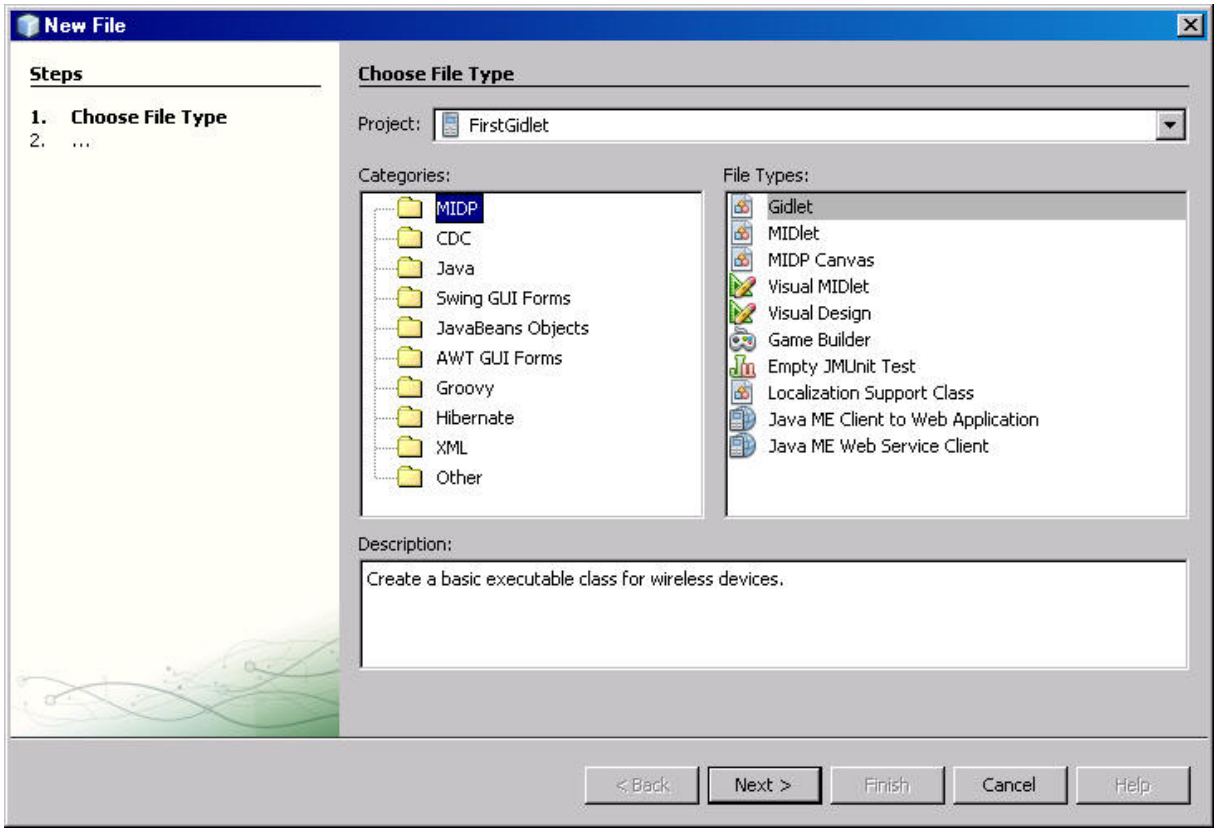
Auf *Next* klicken. Den Projektnamen, z.B. *First* und den Projektort, z.B. *E:\mynb*, eingeben.



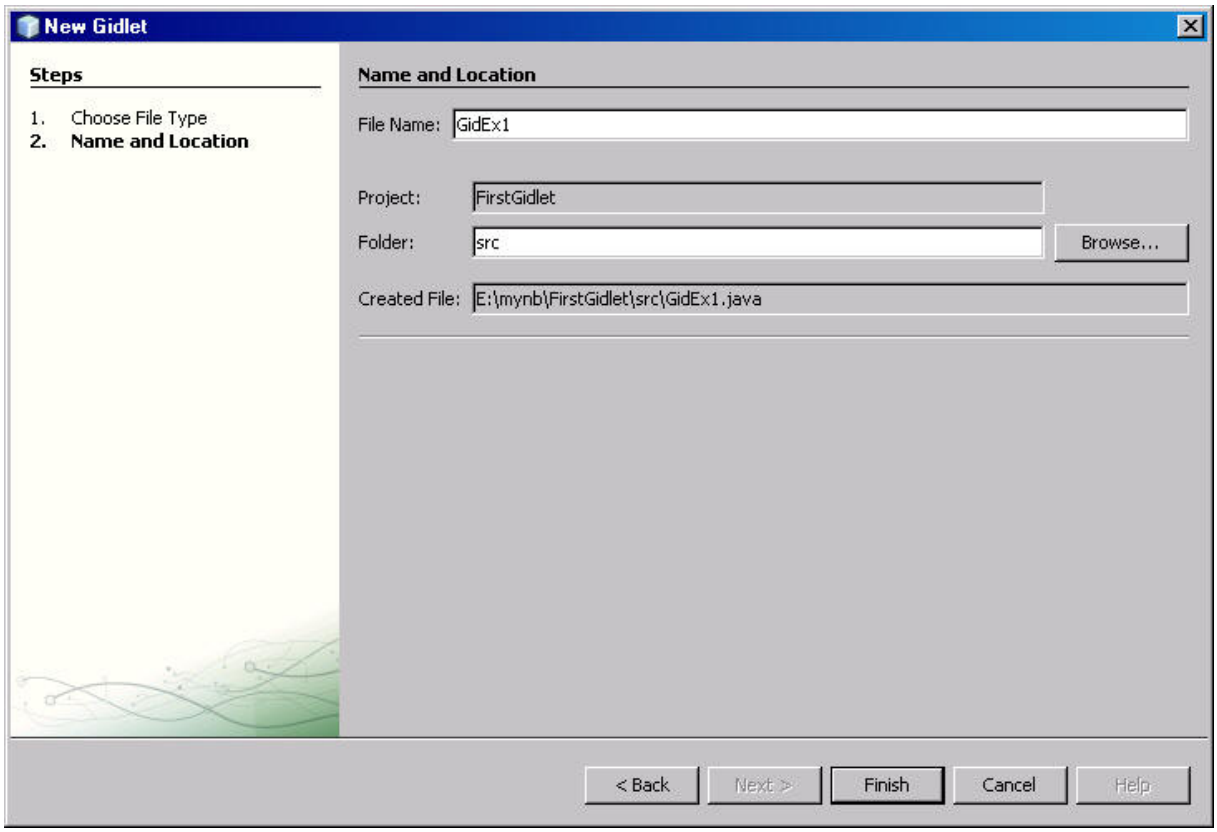
Set as Main Project und *Create Hello MIDlet* abwählen. *Finish* klicken.

Rechter Mausklick im Projektfenster auf *Resources*.. Dann *Add Add JAR/Folder* wählen. Die Datei *gidlet.jar* angeben, die man von *www.aplu.ch/gidlet* heruntergeholt hat.

Rechter Mausklick im Projektfenster auf *FirstGidlet* | *Source Packages* (NICHT auf *FirstGidlet* allein) und *New Other* auswählen. Unter *Categories MIDP* und *File Types GPanel* wählen



Next klicken. Unter *Class Name* einen Klassennamen eintippen, z.B. Gid*Ex1*.

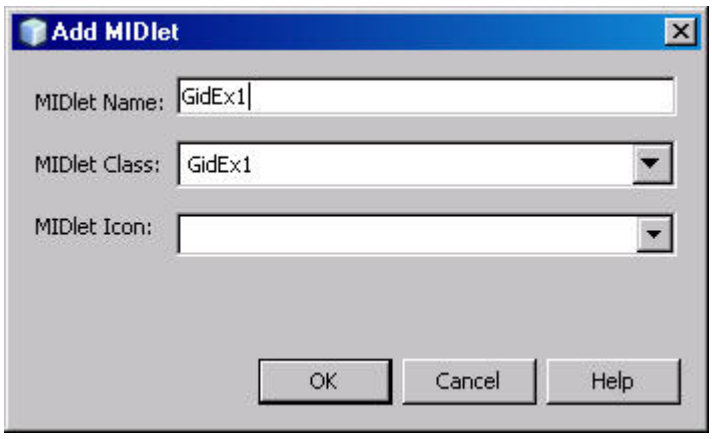


Finish klicken. Das Editorfenster mit dem Template wird geöffnet. Im Editorfenster sollten keine Fehler angezeigt werden. In der Methode *main()* die Zeile

```
showMessage("Hello World", false);
```

einfügen.

Es muss noch der *Application Descriptor* richtig eingestellt werden. Dazu im Projektfenster mit rechtem Mausklick auf *FirstGidlet* und *Properties* wählen. Unter *Category Application Descriptor* und Register *MIDlets* wählen. Button *Add* klicken und unter *MIDlet Name* sowie *MIDlet Class* den Eintrag *GidEx1* einfügen (falls er nicht automatisch eingesetzt ist).

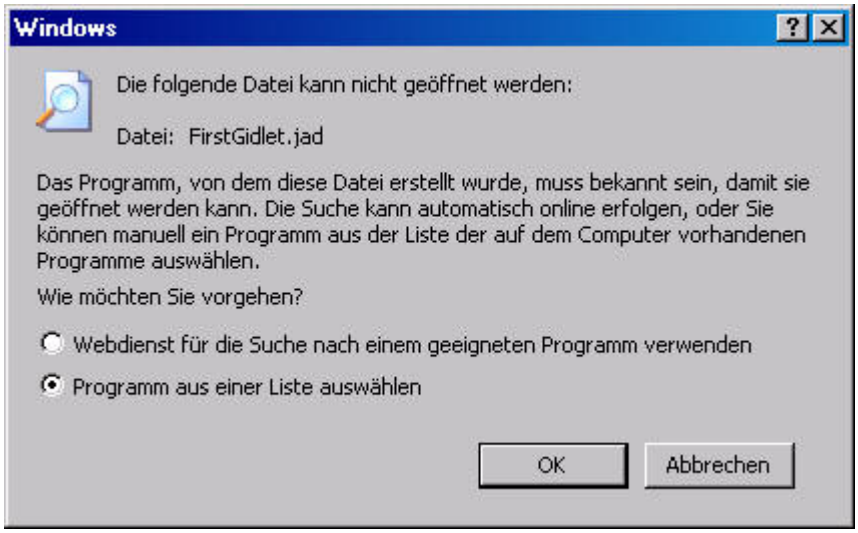


Mit zweimal *OK* bestätigen.

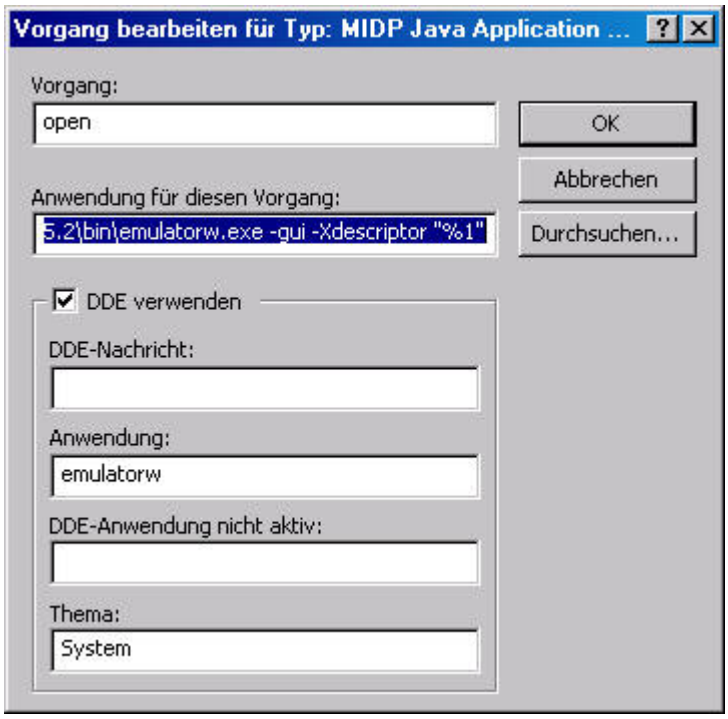
Im Projektfenster rechter Mausklick auf *FirstGidlet* und *Run* auswählen. Die Source wird compiliert, preverified und im Emulator ausgeführt. Die Applikationsdateien *FirstGidlet.jar* und *FirstGidlet.jad*, welche auf das Handy herunter geladen werden, befinden sich um Projektunterverzeichnis *dist*.

Registrierung des Dateityps JAD

Normalerweise ist dem Windows Explorer die Dateiendung JAD nicht bekannt, d.h. der Dateityp JAD nicht registriert. Klickt man auf die JAD-Datei, so öffnet sich ein Fenster:



Man wählt *Programm aus einer Liste auswählen* und klickt *OK*. Im folgenden Dialog wählt man *Durchsuchen* und gibt die Datei *emulatorw.exe* im Unterverzeichnis *mobility8\wtk2.5.3\bin* an (die WTK-Version kann ändern). Im Explorer unter *Extras* | *Ordneroptionen* wählt man im Register *Dateitypen* unter der *Erweiterung JAD* aus und klickt auf den Button *Erweitert* und dann *Bearbeite*n. Unter Anwendung für diesen Vorgang muss man noch den Eintrag *-gui -Xdescriptor* ergänzen:



Installiert man das Wireless Toolkit (WTK) unabhängig von Netbeans auf dem Rechner, so wird der JAD-Dateityp durch das Installationsprogramm registriert.

3.3 Applikation für Lego NXT

Der Rechner muss mit Bluetooth (Widcomm compatible) ausgerüstet sein. Der Bluetooth-Manager des Rechners wird nicht verwendet.

3.3.1 oopDirect

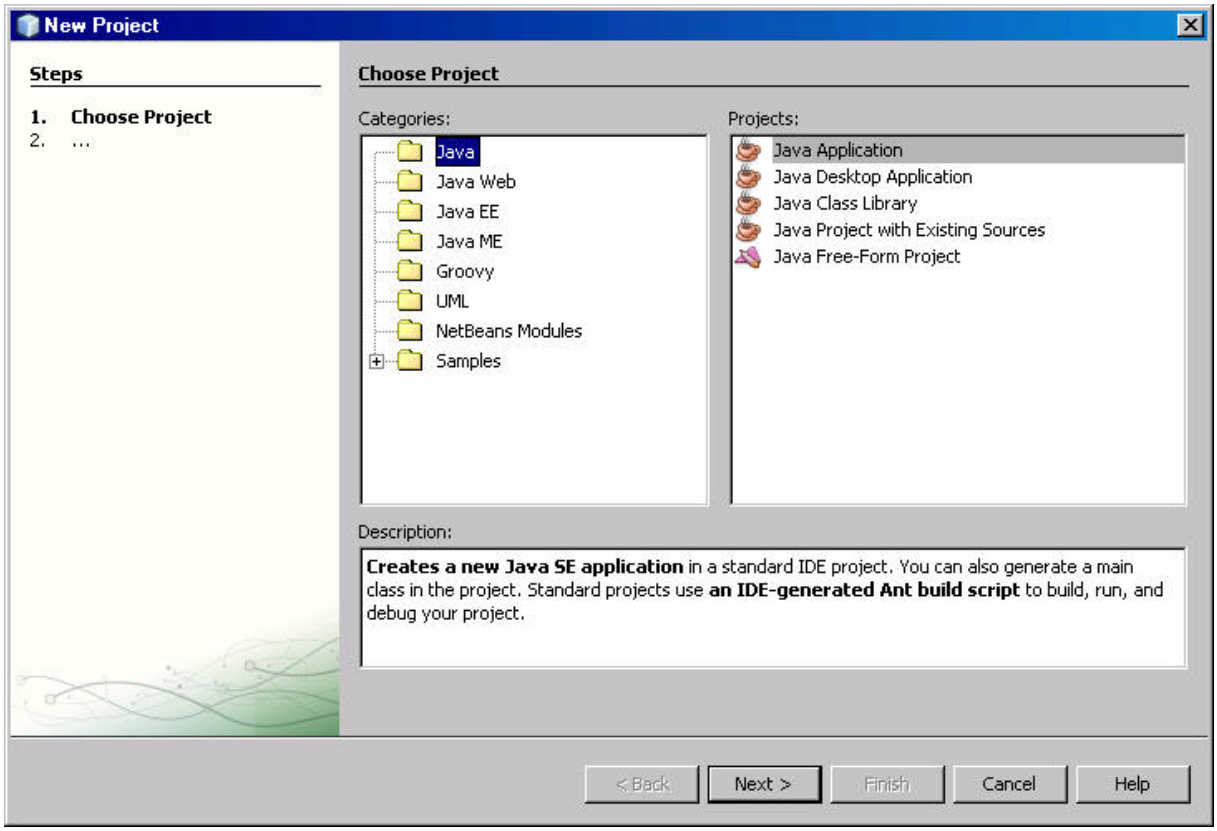
Am besten erstellt man, wie oben beschrieben, ein Template mit dem Namen NxtDirect und dem Inhalt:

```
// ${name}.java

import ch.aplu.nxt.*;

public class ${name}
{
  private NxtRobot robot = new NxtRobot();

  public ${name}()
  {

  }

  public static void main(String[] args)
  {
    new ${name}();
  }
```

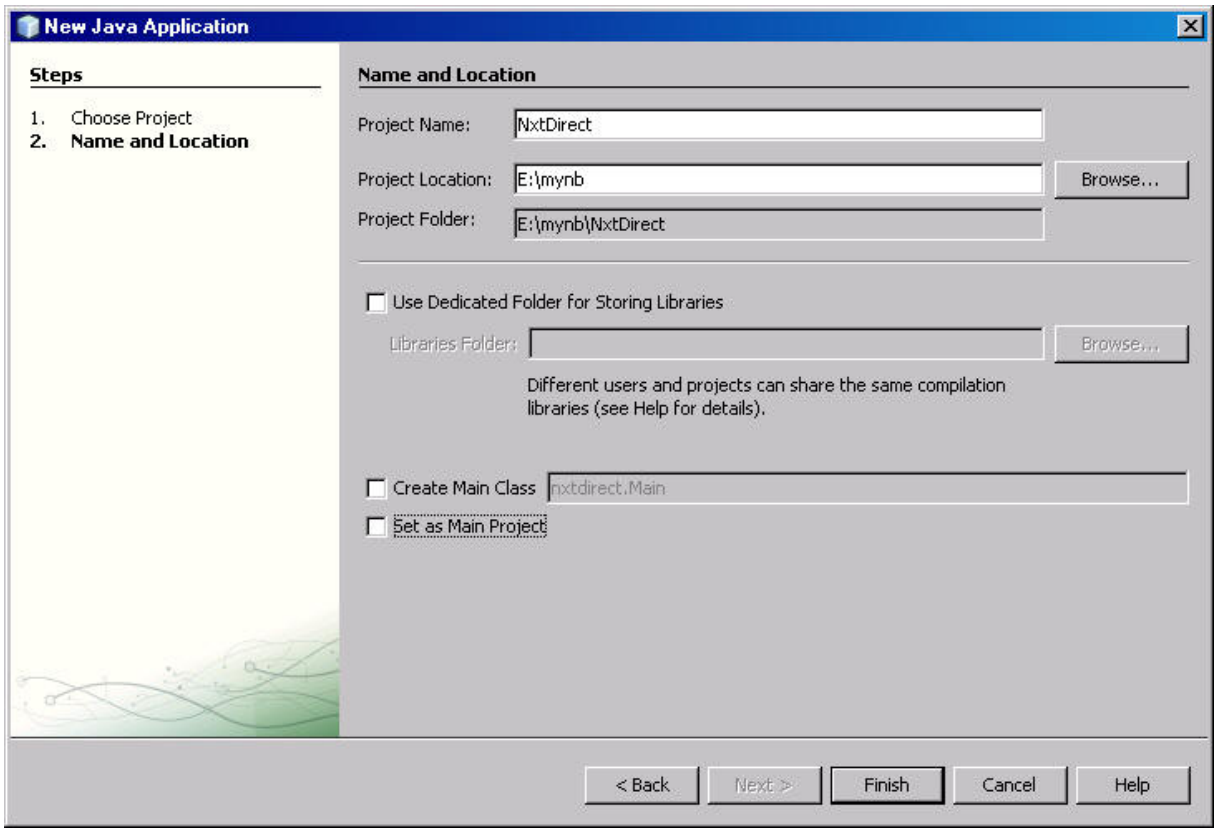
```
}
```

Um ein Programm für oopDirect zu editieren, kompilieren und auszuführen, wie folgt vorgehen:

File | New Project und unter *Categories Java* und unter *Projects Java Application* wählen.



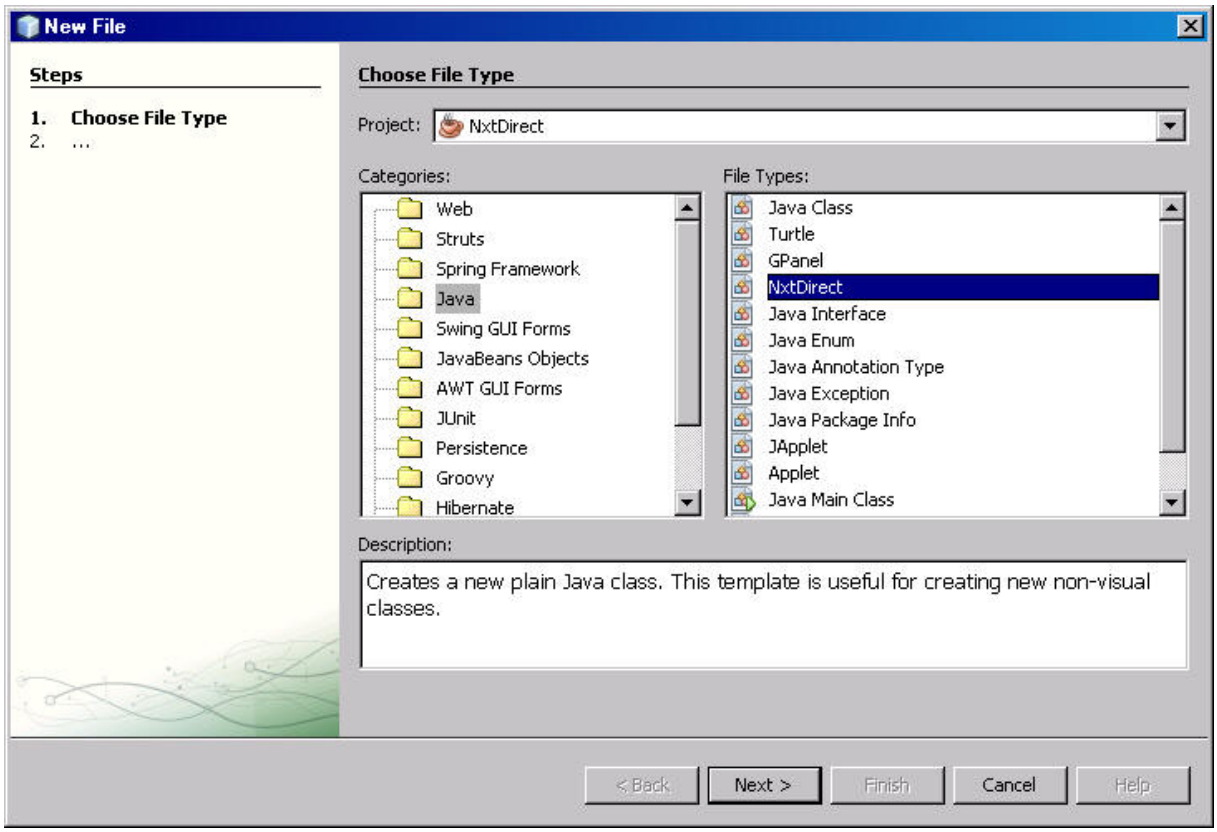
Auf *Next* klicken. Den Projektnamen, z.B. *NxtDirect* und den Projektort, z.B. *E:\mynb*, eingeben.



Create Main Class und *Set as Main Project* abwählen. *Finish* klicken.

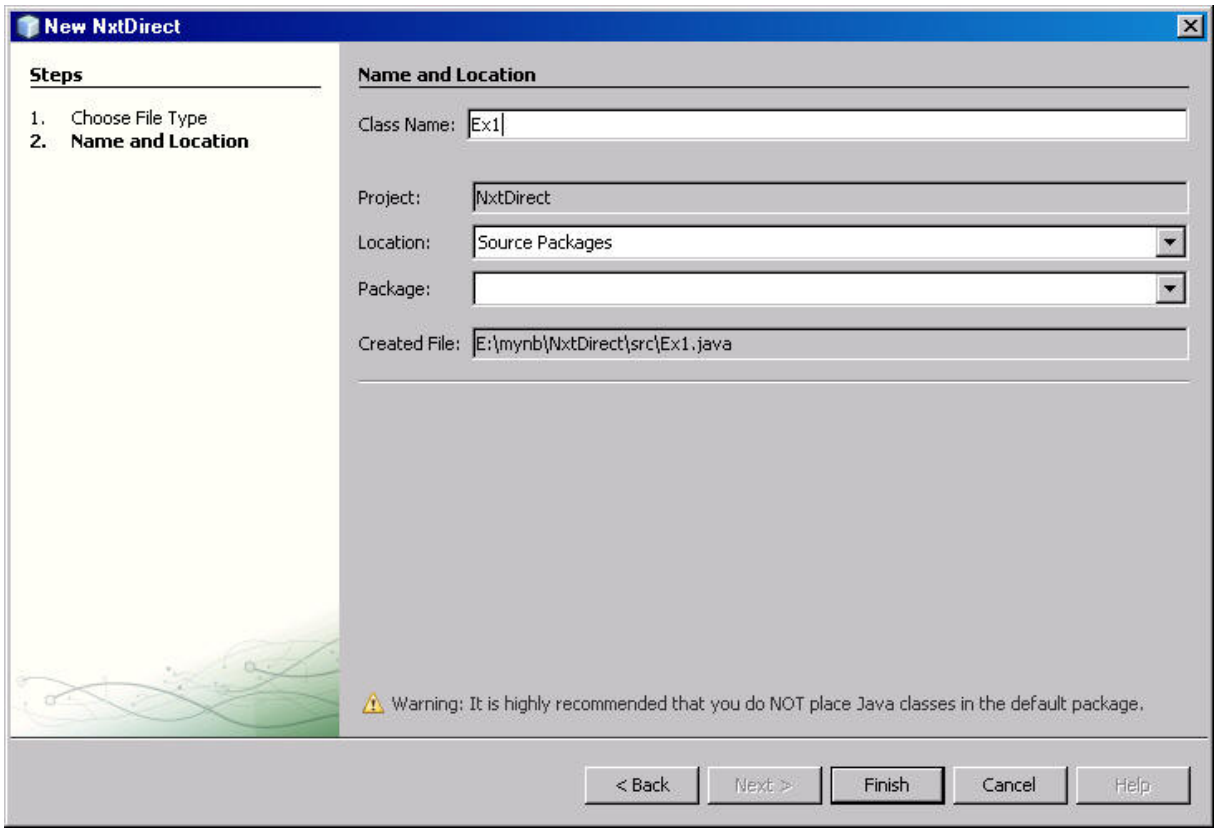
Rechter Mausklick im Projektfenster auf *Libraries*. Dann *Add Add JAR/Folder* wählen. Die Datei *NxtJLib.jar* angeben, die man von *www.aplu.ch/nxt* heruntergeholt hat. Ebenfalls die Datei *bluecove-2.1.0.jar* (oder neuer) angeben, die man von *www.bluecove.org* heruntergeholt hat.

Rechter Mausklick im Projektfenster auf *First* | *Source Packages* und *New Other* auswählen. Unter *Categories Java* und *File Types NxtDirect* wählen



Next klicken. Unter *Class Name* einen Klassennamen eintippen, z.B. *Ex1*.

(Für iCommand aus der leJOS-Distribution benötigt man das rechnerintegrierte Bluetooth und die Installation eines seriellen Treibers, z.B. RxTx, SerialIO, o.a.)



Finish klicken. Das Editorfenster mit dem Template wird geöffnet. Im Editorfenster sollten keine Fehler angezeigt werden. Im Projektfenster rechter Mausklick auf *Ex1.java* und *Run File* auswählen. Die Source wird compiliert und ausgeführt.

3.3.2 oopAutonom und lejosNXJ

leJos NXJ gemäss der Originalanleitung installieren. (Es muss eine Environment Variable *NXJ_HOME* und ein Pfad auf das bin-Verzeichnis vorhanden sein.) Im Unterverzeichnis *lib* von NXJ_HOME erstellt man einen Backup von *classes.jar* und ersetzt das Original durch *classes.jar*, das sich in der Distribution von *NxtJLibA* befindet (Download von *www.aplu.ch/nxt*)

Wie für oopDirect ein neues Projekt erzeugen und *classes.jar* als zusätzliche Jar-Datei angeben. Es kann dasselbe Template verwendet werden. Nach dem Editieren mit rechtem Mausklick auf die Source-Datei und *Compile File* (oder F9) compilieren. Die class-Datei entsteht im Unterverzeichnis *build\classes* des Projektverzeichnisses. In der Command-Shell in dieses Verzeichnis gehen und mit

```
nxjlink <classfile> -o <classfile>.nxj
```

linken, sowie mit

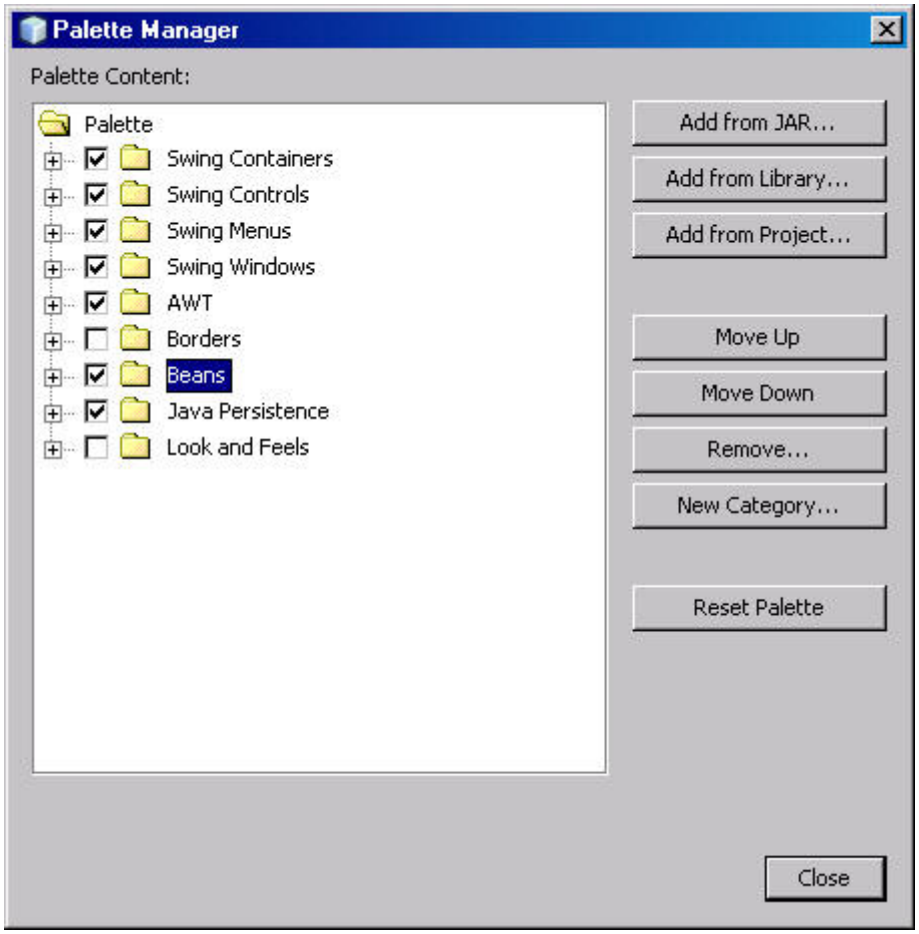
```
nxjupload -r <classfile>.nxj
```

in den NXT uploaden und ausführen lassen.

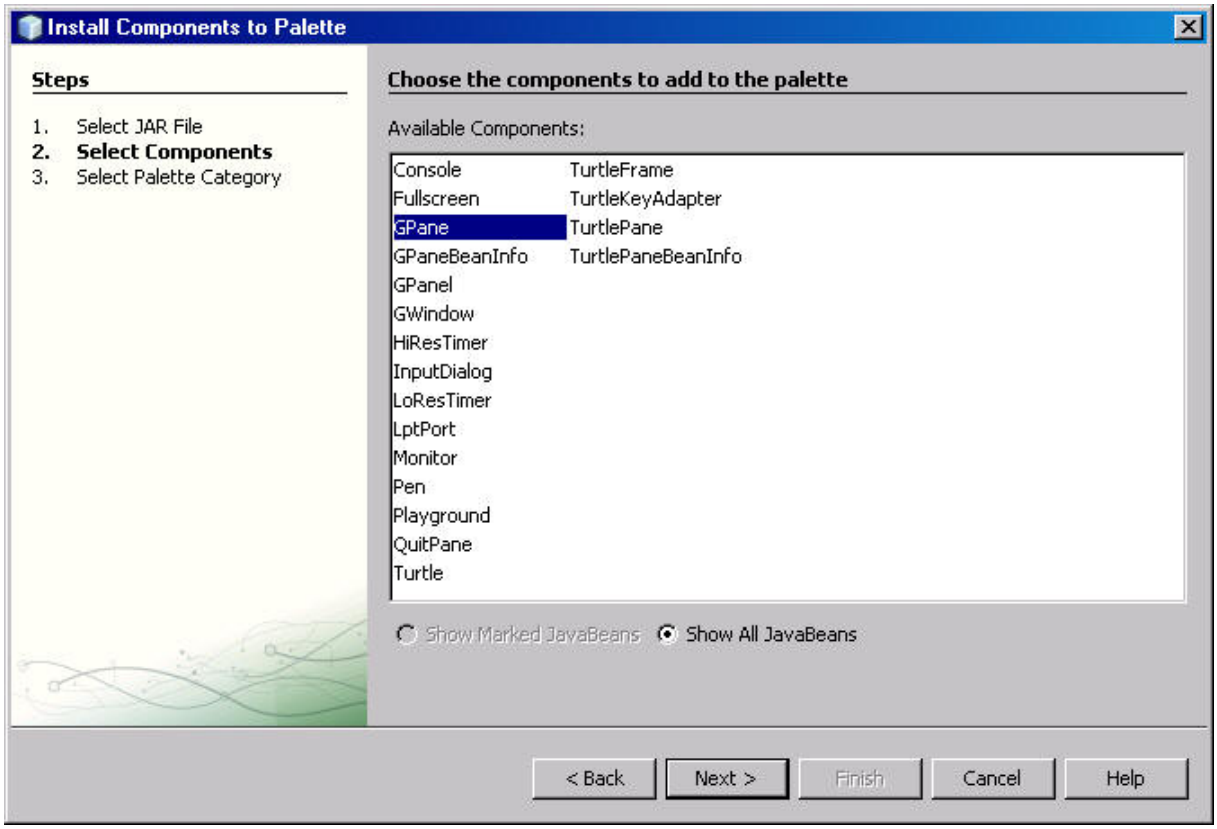
Für die beiden letzten Schritte kann an Ant-Script erstellt werden.

4 Entwicklung von GUIs mit GPane und TurtlePane

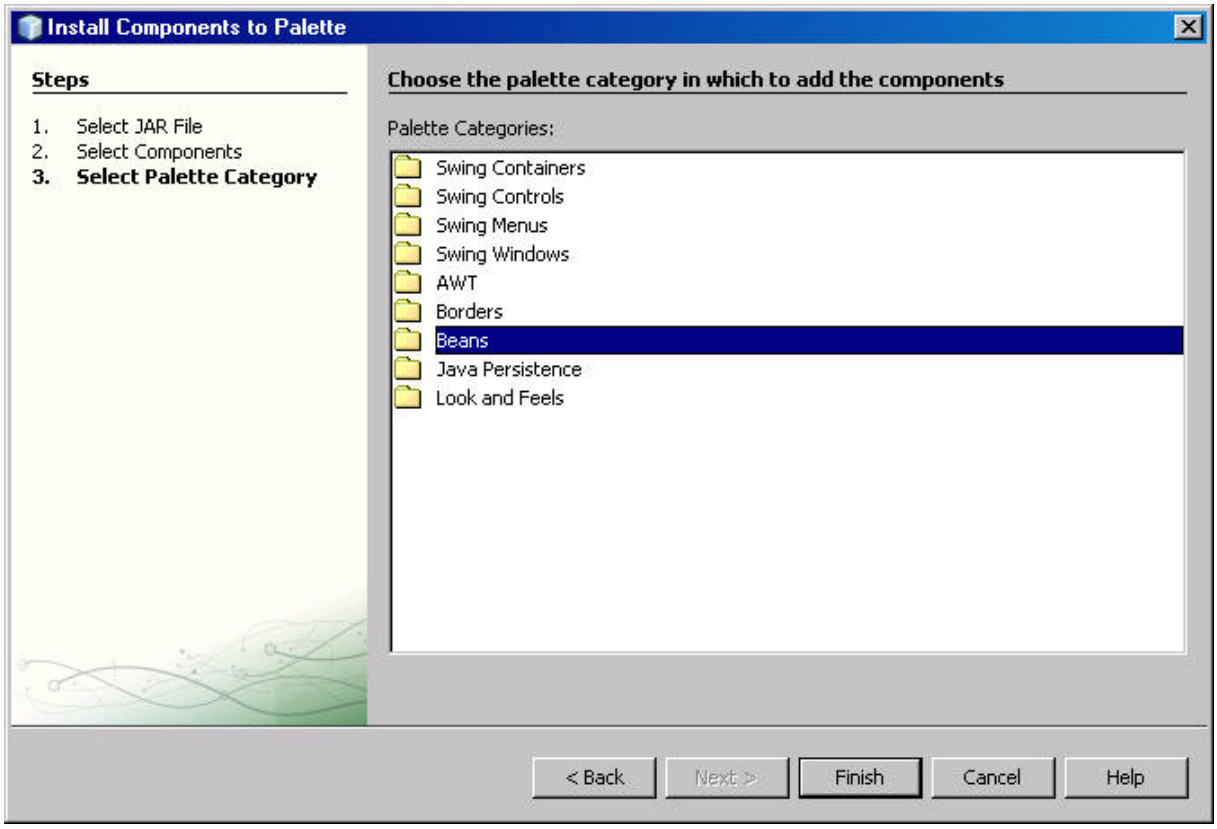
Zuerst werden die neuen Komponenten in die Palette der Swing-Komponenten aufgenommen. Dazu *Tools | Palette | Swing/ AWT Components* wählen. Im Palette Manager *Add from JAR* klicken.



Die Datei *aplu5.jar* wählen und im Fenster *Available Components GPane* auswählen.



Next klicken.



Beans markieren und *Finish* klicken. Dasselbe mit der Komponente *TurtlePane* durchführen.

Projekt, wie oben beschrieben, als Java-Projekt erstellen, aber unter *New* ein *JFrame Form* auswählen. Es wird ein Formular sichtbar. Im Palette-Fenster sind unter *Beans* die beiden Komponenten sichtbar und können wie andere Komponenten in das Formular gezogen werden.

Man beachte, dass animierte Grafik mit GPane und Turtleoperationen grundsätzlich nicht in Button-Callbackmethoden ablaufen sollten, sondern dass ein eigener Workerthread herangezogen werden muss.