

1.	BlueJ installieren	1
2.	BlueJ auf die deutsche Version umstellen.....	1
3.	BlueJ – Extensions	2
a.	Klassenkarte	2
i.	UML Extension	2
ii.	Klassenkarte zum Schulbuch	3
b.	CNU BlueJ Code Formatter	4
4.	Ein BlueJ-Projekt als jar exportieren	4
5.	Einstellungen für Schüler vornehmen	5
6.	Direkteingabe	6
7.	"Mitprotokolliermöglichkeit" via Unit-Test	7
8.	Vorlage für neue Klassen einbinden	7

1. BlueJ installieren (nach dem Buch „Java lernen mit BlueJ“ von David J. Barnes; Michael Kölling)

Um mit BlueJ arbeiten zu können, müssen Sie ein Software Development Kit (SDK) der Java 2 Standard Edition (J2SE) und die BlueJ-Umgebung installieren.

Sie finden das J2SE SDK mit detaillierten Installationsanweisungen unter <http://java.sun.com/j2se/>.

Wenn Sie mit dem Projekt *vordefinierte Klassen fuer Kapitel 1 und 2* oder *klasse10¹* arbeiten möchten, dann müssen Sie eine aktuelle Java-Version (1.6 oder höher) installieren.

Sie finden die BlueJ-Umgebung mit Installationsanweisung unter <http://www.BlueJ.org/>.

2. BlueJ auf die deutsche Version umstellen (nach dem Buch „Java lernen mit BlueJ“ von David J. Barnes; Michael Kölling)

Wir empfehlen für den Gebrauch in der Schule, BlueJ auf Deutsch umzustellen.

Sie können BlueJ so konfigurieren, dass die Benutzerschnittstelle in Deutsch angezeigt wird, indem Sie die Datei `bluej.defs` in einem beliebigen Editor bearbeiten (Diese Datei liegt im Ordner `Programme\BlueJ\lib` bzw. `<BlueJ-Home>\lib`, wobei `<BlueJ-Home>` den Ordner bezeichnet, in dem das BlueJ-System installiert ist.).

In der Datei finden Sie unter anderem die folgenden Zeilen:

```
bluej.language=english
#bluej.language=afrikaans
#bluej.language=catalan
#bluej.language=chinese
#bluej.language=czech
#bluej.language=dutch
#bluej.language=french
#bluej.language=german
#bluej.language=greek
#bluej.language=italian
```

¹ Das Projekt *klasse10* finden Sie im Bereich www.fim.uni-passau.de/schulen.

```
#bluej.language=japanese  
...
```

Ein Hash-Symbol (#) am Anfang der Zeile repräsentiert einen Kommentar. Die Zeile, die nicht mit einem Hash-Symbol beginnt, ist folglich die ausgewählte Einstellung (hier: `bluej.language=english`).

Sie können Deutsch als Benutzersprache auswählen, indem Sie diese Zeilen wie folgt ändern:

```
#bluej.language=english  
#bluej.language=afrikaans  
#bluej.language=catalan  
#bluej.language=chinese  
#bluej.language=czech  
#bluej.language=dutch  
#bluej.language=french  
bluej.language=german  
#bluej.language=greek  
#bluej.language=italian  
#bluej.language=japanese  
...
```

Speichern Sie die Datei und starten Sie BlueJ.

3. BlueJ – Extensions

Unter <http://www.bluej.org/extensions/extensions.html> finden Sie Extension zu BlueJ.

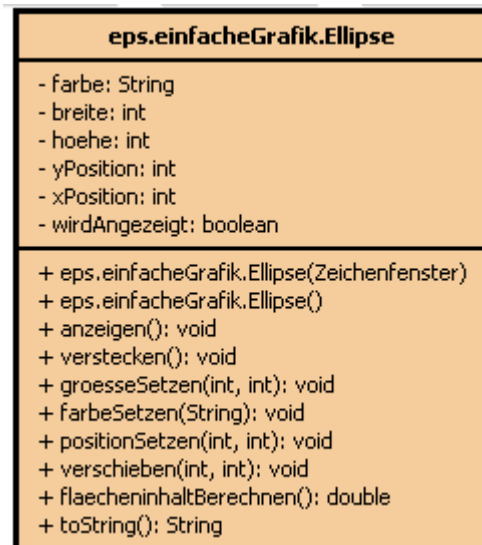
a. Klassenkarte

i. UML Extension

Mit dieser Erweiterung hat man die Möglichkeit sich die UML-Klassenkarte anzeigen zu lassen. Dazu muss man auf das Klassensymbol (in BlueJ) rechts klicken und den Menüpunkt „Display UML“ auswählen.

Downloadmöglichkeiten und die Installationsanweisung finden Sie unter:

<http://www.bluej.org/extensions/extensions.html>



ii. Klassenkarte zum Schulbuch

Unter <http://klassenkarte.steinhuber.de/> finden Sie den Download für unsere Klassenkarte in BlueJ.

Diese Erweiterung bietet Ihnen folgende Vorteile:

- Beim Rechtsklick auf ein Klassensymbol in BlueJ wird der Menüpunkt „Klassenkarte“ angezeigt.
- Es sind verschiedene Einstellungsstufen möglich. Darüber hinaus hat man die Möglichkeit auch direkt auszuwählen, was auf der Klassenkarte angezeigt werden soll (Typen, Sichtbarkeit, Eingangsparameter, usw.). Mit der UML Extension kann keine solche detaillierte Auswahl getroffen werden.
- In einem ausführlichen Handbuch finden Sie sowohl eine Bedienungs- als auch Installationsanleitung. Außerdem werden die verschiedenen Einstellungen genau beschrieben. Sie finden das Handbuch unter <http://klassenkarte.steinhuber.de/files/Handbuch.pdf>.
- Unsere Klassenkarte kann im Gegensatz zur UML Extension auf dem Bildschirm verschoben werden und verschwindet nicht durch Anklicken.
- Dienstbeschreibungen können in der Dokumentation einer Klasse nachgelesen werden. Dazu kann mit der Maus der grüne Schriftzug **Dokumentation**, den die Klassenkarten bereitstellen, angeklickt werden. Es öffnet sich ein Browserfenster mit einer grundsätzlichen Klassenbeschreibung und den einzelnen Dienstbeschreibungen.



b. CNU BlueJ Code Formatter

Mit Hilfe der Formatter-Erweiterung ist es möglich, korrekt übersetzten Programmtext automatisch zu formatieren. Der Formattierer fügt einen „Format“-Button im Editor hinzu, der zum Formatieren des Codes benutzt werden kann.

Unter <http://www.bluej.org/extensions/extensions.html> finden Sie diese Erweiterung. Bitte beachten Sie, dass diese Erweiterung nur für Windows und Macintosh Systeme zur Verfügung steht und installiert werden muss (anders als die Klassenkarten-Erweiterung, die muss nur kopiert werden).

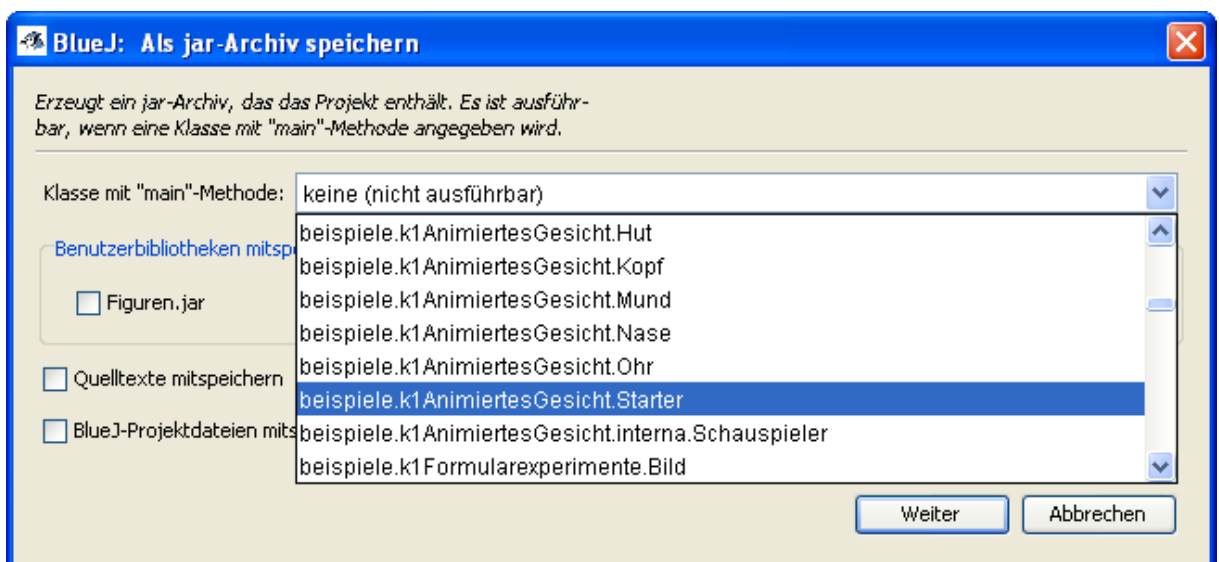
4. Ein BlueJ-Projekt als jar exportieren (nach dem Buch „Java lernen mit BlueJ“ von David J. Barnes; Michael Kölling)

Java-Projekte werden üblicherweise als eine Sammlung von Dateien in einem Verzeichnis abgelegt.

Um eine Anwendung an andere weitergeben zu können, ist es oft einfacher, wenn alle Teile in einer einzelnen Datei zusammengefasst sind. Der Mechanismus von Java für diesen Zweck ist das *Java Archive Format (jar)*. Alle Dateien einer Anwendung können dabei in einer einzelnen Datei zusammengefasst und dennoch ausführbar sein. (Wenn Sie bereits das Komprimierungsformat „zip“ kennen: Es ist das gleiche Format. jar-Dateien können mit zip-Programmen geöffnet werden und umgekehrt.)

Um eine jar-Datei ausführbar zu machen, muss festgelegt werden, welche die Startklasse ist. (In dieser Startklasse muss die `main`-Methode enthalten sein.) Dazu wird in einer jar-Datei eine Textdatei abgelegt (die so genannte *Manifest-Datei*), die diese Information enthält. BlueJ bietet hierbei Unterstützung an.

Um eine ausführbare jar-Datei in BlueJ zu erstellen, können Sie im Menü *Projekt* den Eintrag *Als jar-Archiv speichern...* aufrufen und im folgenden Dialog die Klasse auswählen, die die `main`-Methode enthält:



Falls Sie die java-Dateien mitexportieren wollen, müssen Sie *Quelltexte mitspeichern* auswählen. (Ansonsten werden nur die class-Dateien gespeichert, Sie können dann zwar das Programm ausführen, nicht aber den Quelltext ansehen.)

Sobald die ausführbare jar-Datei in BlueJ erzeugt wurde, kann sie mit einem Doppelklick gestartet werden. Dazu muss auf dem Rechner, der die jar-Datei ausführen soll, ein JDK (ein

Java Development Kit) oder eine JRE (Java Runtime Environment) installiert und mit *.jar*-Dateien assoziiert sein.

Falls Ihre *.jar*-Dateien nicht mit Ihrer Java Runtime Environment, sondern beispielsweise mit einem Zip-Programm, assoziiert sind, können Sie diese auch über die Konsole mit dem Befehl:

```
java -jar nameDerJarDatei.jar
```

ausführen. Sie müssen sich dazu im Verzeichnis befinden, in dem die *jar*-Datei abgelegt ist.

Für weitere Fragen zu dieser Operation können Sie das BlueJ-Tutorial lesen, das Sie im Menü *Hilfe* über den Eintrag *BlueJ-Tutorial* oder auf den BlueJ-Webseiten finden.

Das Projekt *klasse10* können Sie problemlos als *jar*-Datei exportieren. Es kann jedoch bei Projekten, die auf externe Ressourcen wie Bilder, Sound oder Texte zugreifen und keinen Classloader verwenden, zu Problemen kommen, wenn man diese als *jar*-Dateien exportieren will.

5. Einstellungen für Schüler vornehmen



Wenn Schüler den Menüpunkt „Bearbeiten“ auswählen, können sie die Implementierung der Klassen sehen und somit auch am Code etwas verändern. Wenn Sie das vermeiden und den Schülern trotzdem die Möglichkeit geben wollen, sich die API anzeigen zu lassen, dann gehen Sie folgendermaßen vor:

- 1) Via Menüpunkt „Werkzeuge“ den Punkt „Dokumentation erzeugen“ auswählen, damit werden für alle Klassen im ganzen Projekt die Javadoc-Dateien erzeugt. Anschließend sind im Unterverzeichnis *doc* die Javadoc HTML-Dateien zu finden. (Das Verzeichnis *doc* befindet sich im Ordner des jeweiligen BlueJ-Projektes.) Diese Dateien können entweder mit einem Interbrowser gelesen werden. (*Klassenname.html* enthält die jeweilige Beschreibung der Klasse). Oder man wählt über den Schriftzug „Dokumentation“, den jede Klassenkarte anbietet, direkt genau die Dokumentation zur angezeigten Klassenkarte an. Schritt 1) haben wir für unsere Projekte *vordefinierte Klassen fuer Kapitel 1 und 2* oder *klasse10*² schon durchgeführt.
- 2) Nun sollten Sie alle die *java*-Dateien aus Unterordnern löschen, deren Implementation Schüler nur ablenken würde. Die Schüler können dann im Kontextmenü der entsprechenden Klasse den Punkt „Bearbeiten“ nicht mehr auswählen.



² Das Projekt *klasse10* finden Sie im Bereich www.fim.uni-passau.de/schulen.

6. Direkteingabe

Die Direkteingabe wird mit *Strg+E* oder über *Ansicht* → *Direkteingabe anzeigen* aktiviert. Die Direkteingabe erscheint im BlueJ-Fenster rechts unten und eignet sich sehr gut um Methodeaufrufe auszuführen. Die Schüler können die Direkteingabe nutzen, um die Punktnotation einzuüben und typischen Fehlermeldungen in Java kennen zu lernen. Auch Objekte lassen sich über die Direkteingabe erzeugen.

```

> dreieck1.anzeigen();
> Rechteck haus = new Rechteck();
> haus.positionSetzen(20,23);
>

```

Objekt über die Direkteingabe erzeugen:

1. `Rechteck haus = new Rechteck();`
Mit dieser **Anweisung (mit Strichpunkt abgeschlossen)** wird ein Rechteck-Objekt mit dem Objektnamen `haus` erzeugt.
Diese Objekte sind jedoch nur über die Direkteingabe referenzierbar und nicht über die Objektleiste.
2. `new Rechteck`
Mit diesem **Ausdruck (hat keinen Strichpunkt am Ende!)** können durch Anwählen des roten Symbols und Hinüberziehen in die Objektleiste beliebig viele Objekte vom Typ `Rechteck` erzeugt werden.



```

> Rechteck haus = new Rechteck();
> haus.anzeigen();
> new Rechteck()
<object reference> (Rechteck)
>

```

weitere Tipps und Anmerkungen:

1. Sie können in der Direkteingabe mit den **Pfeiltasten** arbeiten und dadurch das zuletzt Geschriebene wieder aufrufen.
2. Möchten Sie eine Methode mit Rückgabewert in der Direkteingabe aufrufen und den Rückgabewerte über die Direkteingabe zurückerhalten, dann benötigen Sie dazu einen Ausdruck und keine Anweisung. Ein Ausdruck wird ausgewertet und liefert Ihnen einen Wert zurück. Ausdrücke werden in BlueJ nicht mit einem Strichpunkt abgeschlossen, Anweisungen dagegen schon.

```

> haus.flaecheninhalteBerechnen();
> haus.flaecheninhalteBerechnen()
5000.0 (double)
> haus.flaecheninhalteBerechnen() + rechteck1.flaecheninhalteBerechnen()
10000.0 (double)
> haus.flaecheninhalteBerechnen() + rechteck1.flaecheninhalteBerechnen();
Error: not a statement
>

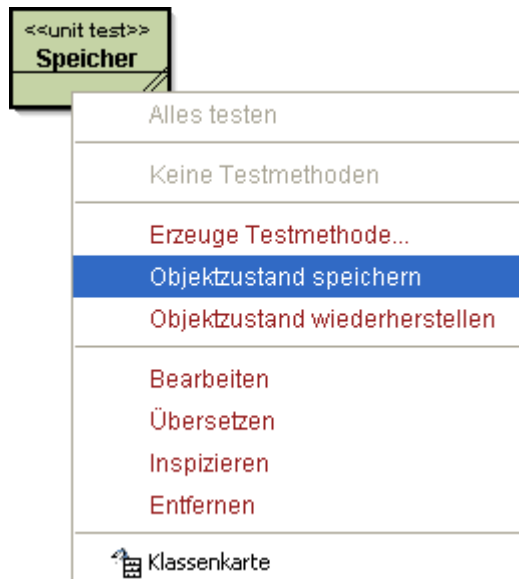
```

7. "Mitprotokolliermöglichkeit" via Unit-Test

Die `<<unit test>>` - Klassen können im 1.Kapitel zum Speichern von Anweisungsfolgen verwendet werden.

Anmerkung: Informatiker verwenden diese Klassen zum automatischen Testen von Umsetzungen. Daher der Name.

Bei manchen Projekten bietet es sich an, die erzeugten Objekte und die durchgeführten Methodenaufrufe zu speichern. Somit kann zu einem späteren Zeitpunkt weitergearbeitet werden. Wir haben dazu Schülern für manche Aufgaben eine Klasse `Speicher` zur Verfügung gestellt.



Über das Kontextmenü (Rechtsklick auf die Klasse) und den Menüpunkt *Objektzustand speichern* werden die erzeugten Objekte und die durchgeführten Methodenaufrufe gespeichert. Über dem Menüpunkt *Objektzustand wiederherstellen* werden diese Objekte wieder erzeugt sowie die zuvor durchgeführten Methodenaufrufe erneut ausgeführt.

In der Klasse `Speicher` kann jedoch immer nur ein Zustand gespeichert werden. Betrachtet man beispielsweise das Projekt *einfacheGrafik*, so kann immer nur ein „Bild“ (hier ist das Bild auf dem Zeichenfenster gemeint und nicht ein Objekt der Klasse `Bild`) gespeichert werden. Um ein weiteres Bild zu speichern muss eine neue Klasse angelegt werden (*Neue Klasse* und *Unit-Test* auswählen). Speichern Sie dennoch ein neues Bild, dann wird das alte Bild „überschrieben“ (die zuvor gespeicherten Objekte und die durchgeführten Methodenaufrufe werden durch die neuen erzeugten Objekte und durchgeführten Methodenaufrufe überschrieben).

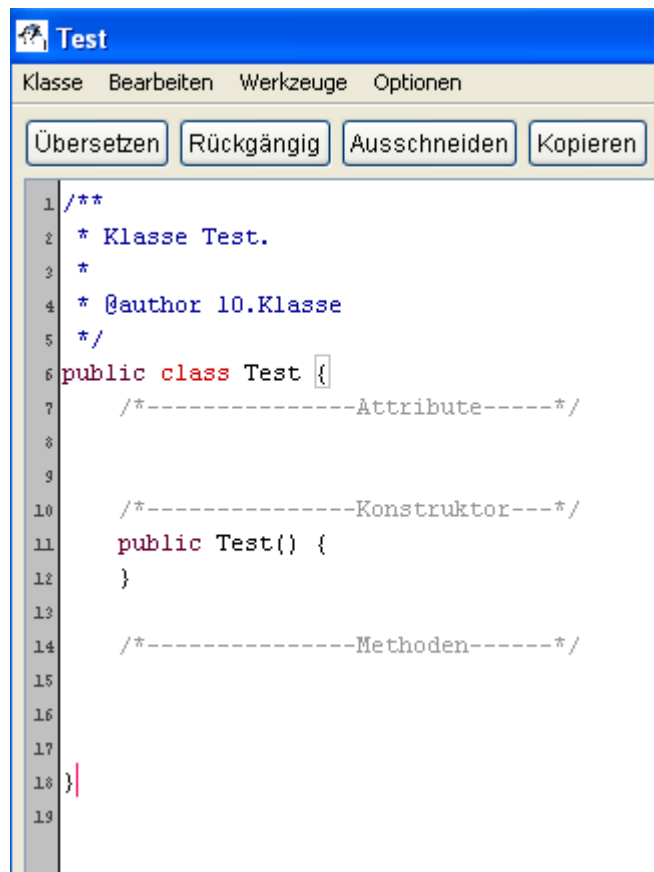
8. Vorlage für neue Klassen einbinden

Wird in BlueJ eine neue Klasse erzeugt, dann ist diese Klasse nicht leer, sondern es wird der Inhalt angezeigt, der in der Datei `stdclass.tmpl` (Abkürzung für: Standard Klassen Template) gespeichert ist. Sie haben die Möglichkeit diese Voreinstellung zu ändern und diesen Inhalt nach Ihren Vorstellungen anzupassen. Sie können aber auch unsere mitgelieferte Vorlage `stdclass.tmpl` verwenden.

Was ist zu tun, wenn man die Vorlage verwenden möchte?

Im Verzeichnis `<BLUEJ_HOME>/lib/german/templates/newclass/` ersetzen Sie die Datei `stdclass.tmpl` mit unserer Vorlage `stdclass.tmpl`.

Erzeugen Sie nun in der Umgebung BlueJ eine neue Klasse mit dem Klassennamen `Test`, dann erhalten Sie die unten abgedruckte Klasse:



The screenshot shows the BlueJ IDE interface for a class named 'Test'. The window title is 'Test'. The menu bar includes 'Klasse', 'Bearbeiten', 'Werkzeuge', and 'Optionen'. Below the menu bar are buttons for 'Übersetzen', 'Rückgängig', 'Ausschneiden', and 'Kopieren'. The main area displays the following Java code:

```
1 /**
2  * Klasse Test.
3  *
4  * @author 10.Klasse
5  */
6 public class Test {
7     /*-----Attribute-----*/
8
9
10    /*-----Konstruktor---*/
11    public Test() {
12    }
13
14    /*-----Methoden-----*/
15
16
17
18 }
19
```

Wie kann man *stdclass.tmpl* selbst verändern?

Die Datei einfach mit einem Texteditor öffnen, Änderungen vornehmen und speichern. Sodann die Datei in das oben angegebene Verzeichnis kopieren.

Anmerkung:

Wir haben Ihnen noch Vorlagen für die Dateien *abstract.tmpl* und *unittest.tmpl* zur Verfügung gestellt. Diese Vorlagen können Sie wie *stdclass.tmpl* im gleichen Ordner von BlueJ ersetzen und kommen dann zum Einsatz, wenn eine neue abstrakte Klasse bzw. eine neue Unit-Test Klasse erzeugt wird.