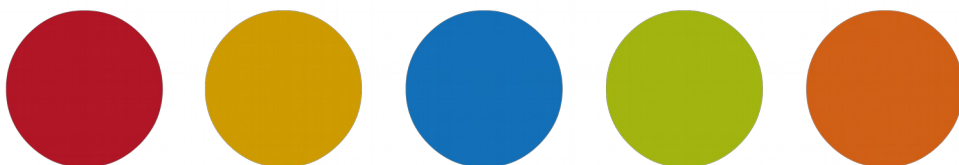


Erzeugen und Testen

Leittext

Modulname:	Lehrerdaten
------------	-------------

Stand: 28. Nov 2016



© Christine Janischek

Inhaltsverzeichnis

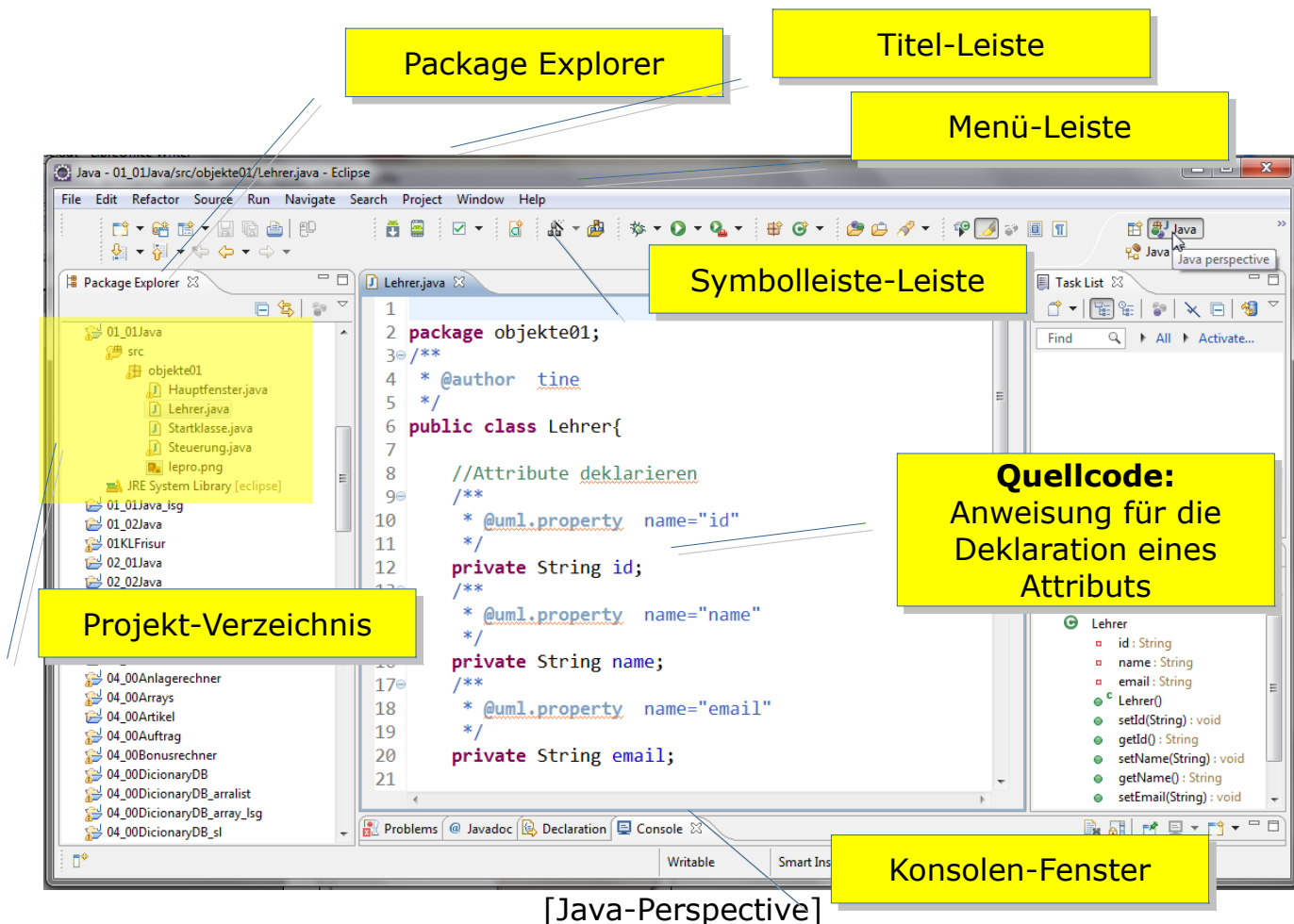
1 Grundlagen.....	3
1.1 Programming-Rules.....	3
1.2 Benutzeroberfläche.....	4
1.3 Projekte anlegen und bearbeiten.....	5

1 Grundlagen



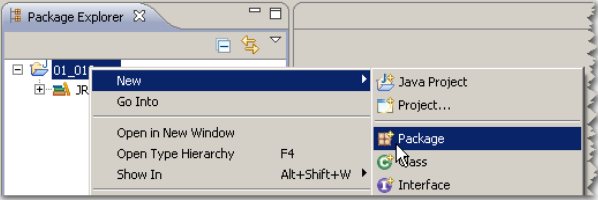
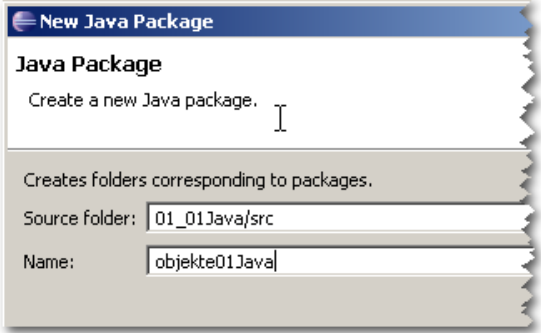
1.1 Programming-Rules

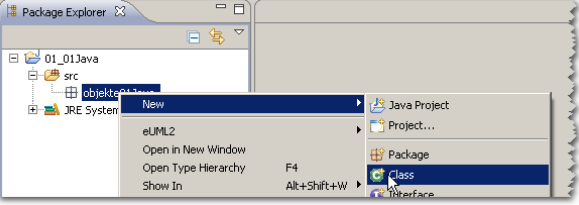
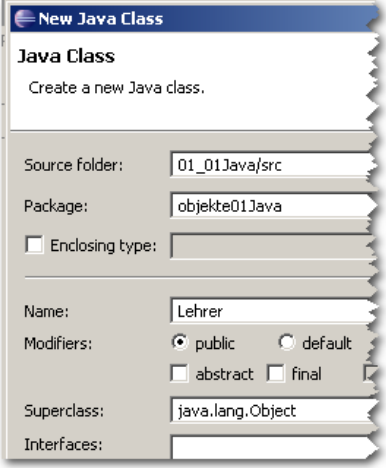
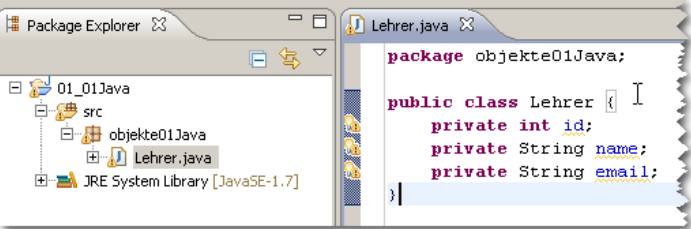
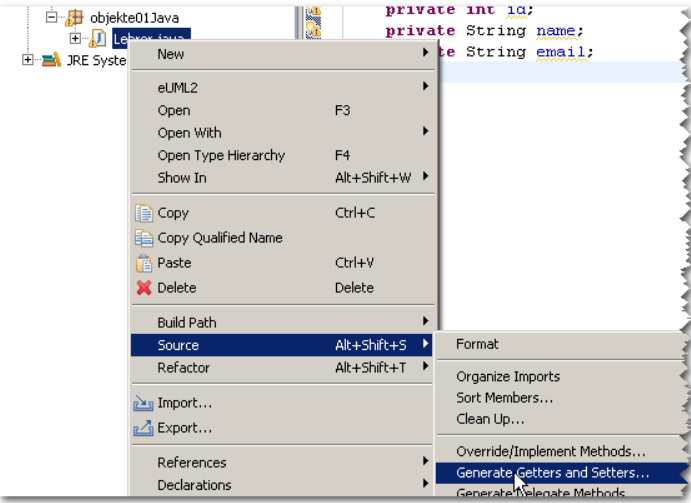
Klassennamen	Substantiv, Hauptwort, großschreiben		
Objektname	this (für aktuelles Objekt), Substantiv, Hauptwort, kleinschreiben		
Attributname	Substantiv, Hauptwort, kleinschreiben		
	Feld	Rule	Beispiel
	Übergabeparameter	pAttributname	pVorname
	Lokale Attribute	mAttributname	mVorname
Konstruktorname	Ist der Klassenname.		
Methodenname	Verben oder Adjektive, kleinschreiben		
Kontrollstrukturen	Fallunterscheidungen, Wiederholstrukturen		
Fachklasse	Klasse, Maske für die Deklaration (Definition) von Eigenschaften und Verhaltensweise einer ganzen Gruppe von Objekten.		
Hauptfenster	Klasse für eine Benutzeroberfläche		

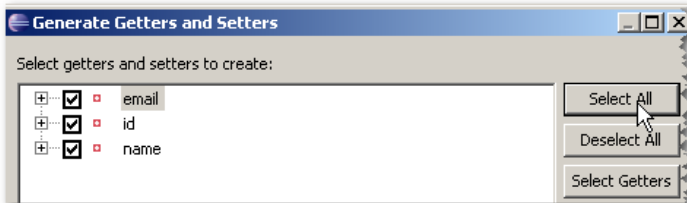
1.2 Benutzeroberfläche



1.3 Projekte anlegen und bearbeiten

Bild	Hinweise
 <p>The screenshot shows the Package Explorer window with a context menu open. The menu items are: New, Show In, Alt+Shift+W, Java Project, Project..., and Package.</p>	<p><i>Im Kontextmenü (rechte Maustaste) die Option → New → JavaProject wählen</i></p>
 <p>The screenshot shows the 'New Java Project' wizard dialog. The title is 'New Java Project'. The main text says 'Create a Java Project' and 'Create a Java project in the workspace'. At the bottom, there is a text field for 'Project name:' containing the text '01_01Java'.</p>	<p><i>Geben Sie den Projektnamen ein und klicken Sie auf → Next und → Finish</i></p>
 <p>The screenshot shows the Package Explorer window with a context menu open. The menu items are: New, Go Into, Open in New Window, Open Type Hierarchy (F4), Show In, Alt+Shift+W, Java Project, Project..., Package, Class, and Interface.</p>	<p><i>Im Kontextmenü (rechte Maustaste) die Option → New → package wählen</i></p>
 <p>The screenshot shows the 'New Java Package' wizard dialog. The title is 'New Java Package'. The main text says 'Java Package' and 'Create a new Java package.'. Below that, it says 'Creates folders corresponding to packages.'. There are two text fields: 'Source folder:' containing '01_01Java/src' and 'Name:' containing 'objekte01Java'.</p>	<p>Paketname eingeben und → <i>Finish</i> anklicken.</p>

	<p><i>Paketname anklicken >> Kontextmenü (rechte Maustaste) die Option >> New >> Class wählen</i></p>
	<p>Name einer Klasse eingeben und → Finish anklicken.</p>
	<p>Klassenattribute (Eigenschaften) für die Objekte einer Klasse festlegen.</p> <p>In jeweils einer Zeile wird mit dem Modifikator → <i>private</i> der Zugriff auf das Attribut geregelt.</p>
	<p><i>Kontextmenü → Source → Generate Getters and Setters</i></p>



→ *Select All* anklicken, um für alle Attribute die Get- und Set-Methoden zu erstellen. Danach auf → *OK* klicken.

```

19-   public void setId(int pId) {
20-       this.id = pId;
21-   }
22-
23-   public String getName() {
24-       return name;
25-   }
26-
27-   public void setName(String pName) {
28-       this.name = pName;
29-   }
30-
31-   public String getEmail() {
32-       return email;
33-   }
34-
35-   public void setEmail(String pEmail) {
36-       this.email = pEmail;
37-   }

```

Get- und Set-Methoden: Parameterattribute an die bestehenden „Programming-Rules“ anpassen.

```

9
10
11-   //Konstruktor
12-   public Lehrer() {
13-   }
14-

```

Konstruktor einfügen.

```

34-   public void lehrerAnzeigen() {
35-       System.out.println("ID: " + this.getId());
36-       System.out.println("Name: " + this.getName());
37-       System.out.println("E-Mail: " + this.getEmail());
38-   }
39-

```

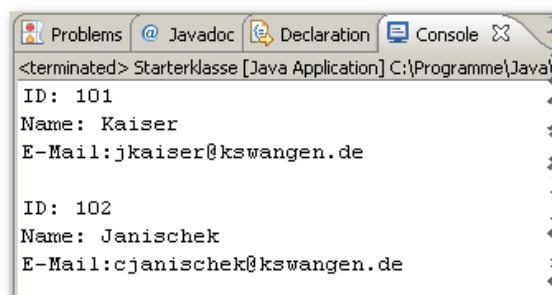
Quellcode für eine Methode ohne Rückgabewert erstellen. Fügen Sie den angezeigten Quellcode für die Ausgabe der Lehrerdaten auf der Konsole ein.



Eine Starterklasse erzeugen.

Die Starterklasse enthält die → *Main-Methode*. Die → *Main-Methode* ist der Startpunkt für den Compiler und für den Start der Anwendung.

Hier muss stehen, was nacheinander passieren soll um die unten angezeigte Ausgabe zu erzeugen.



Aufgabe:

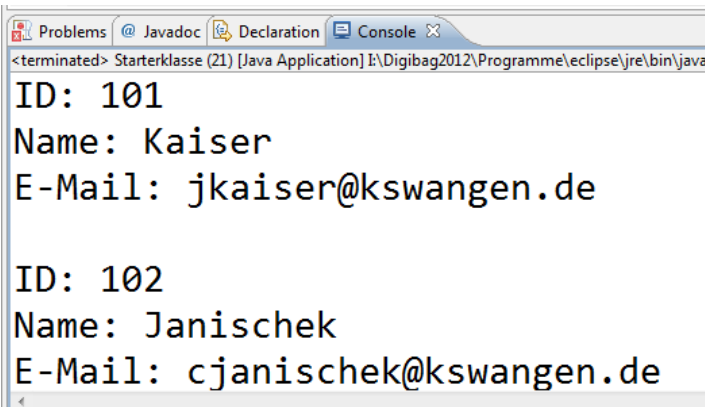
Erzeugen Sie dazu den Quellcode für die Starterklasse:

```
Lehrer.java  Starterklasse.java
3
4 public class Starterklasse{
5
6     public static void main(String[] args){
7
8         //Erzeuge ein Objekt der Klasse Lehrer
9         /**
10          * @uml.property name="lehrer1"
11          * @uml.associationEnd
12          */
13         Lehrer lehrer1 = new Lehrer();
14
15         //Eingabe: Übermittele die Werte an das Objekt der Fachklasse
16         lehrer1.setId("001");
17         lehrer1.setName("Christine Janischek");
18         lehrer1.setEmail("janischek@bs-wangen.de");
19
20         //Verarbeitung: Verarbeitet wird nichts!
21
22         //Ausgabe: Ermittle die Werte und gebe sie aus
23         System.out.println("ID: "+lehrer1.getId());
24         System.out.println("Name: "+lehrer1.getName());
25         System.out.println("E-Mail: "+lehrer1.getEmail());
26
27     }
28 }
```

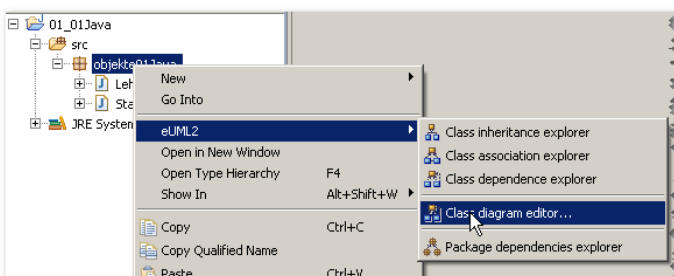



Mit einem Klick auf die Schaltfläche → Run wird die Anwendung kompiliert und ausgeführt.

Das bedeutet konkret, dass die Dateien von der höheren Programmiersprache Java in die Maschinensprache übersetzt und ausgeführt werden.



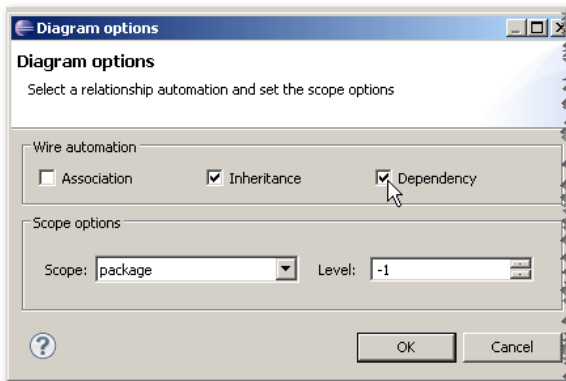
Wenn Sie keine Fehler gefunden haben so erfolgt die Ausgabe der Lehrerdaten auf der Konsole.



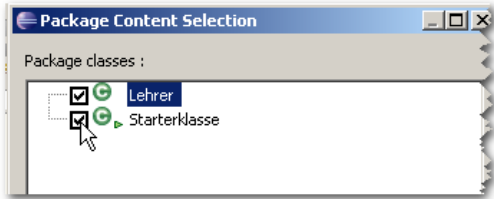
Im nächsten Schritt erzeugen wir per Reverse-Engineering (aus bestehendem Quellcode) ein UML-Klassendiagramm.

Wir nutzen dazu die Erweiterung → eUML2 Free Edition von Soyatec.

Paketname anklicken → Kontextmenü (rechte Maustaste) die Option → eUML2 → Class diagram editor



→ OK klicken.



Häkchen setzen und → *Finish* klicken.

