

Programmieren I – Kapitel 3

JAVA – ARBEITEN MIT ECLIPSE

Qualitätskriterien mit Eclipse sicherstellen (1/2)

Eclipse bietet Unmengen von Werkzeugen, welche die Entwicklung nicht nur beschleunigen, sondern auch die Qualität der Software sicherstellen. Die folgenden **Qualitätskriterien sind universell und bei jeder Entwicklung zu berücksichtigen.**

Verträglichkeit (Kompatibilität)

das Maß der Leichtigkeit, mit der Softwareprodukte mit anderen verbunden werden können. Zu oft gibt es große Interaktionsprobleme, weil ein Programm unverträgliche Annahmen über den Rest der Welt macht. Der Schlüssel zu einer besseren Verträglichkeit liegt in der Standardisierung der Kommunikation zwischen verschiedenen Programmen.

Portabilität

die Leichtigkeit, mit der Softwareprodukte auf verschiedene Hardware- und Software-Umgebungen angepaßt werden können.

Verifizierbarkeit

die Leichtigkeit, mit der AbnahmeprozEDUREN (Fehlererkennung und -verfolgung) erzeugt werden können.

Integrität

Fähigkeit des Softwaresystems, seine verschiedenen Komponenten gegen unberechtigte Zugriffe und Veränderungen zu schützen.

Benutzerfreundlichkeit

Leichtigkeit, mit der die Benutzer die Handhabung des Softwaresystems wahrnehmen und erlernen können.

Qualitätskriterien mit Eclipse sicherstellen (2/2)

Korrektheit

die Fähigkeit von Softwareprodukten, die definierten Anforderungen exakt zu erfüllen.

Robustheit

die Fähigkeit von Softwaresystemen, auch unter außergewöhnlichen Bedingungen zu funktionieren.

Erweiterbarkeit

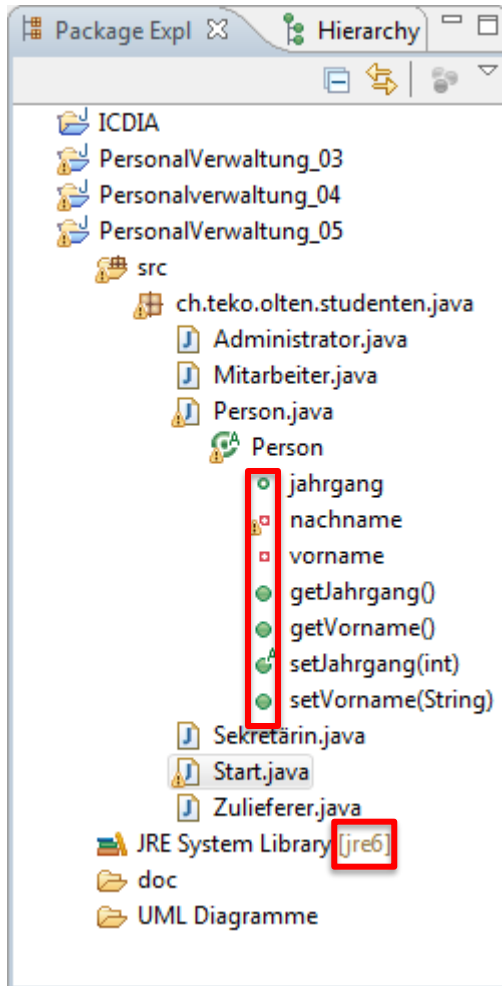
die Leichtigkeit, mit der Softwareprodukte an neue Anforderungen angepaßt werden können. Erweiterbarkeit wird begünstigt durch die Einfachheit des Entwurfs und durch Dezentralisierung in Module mit hoher Autonomie.

Wiederverwendbarkeit

Eigenschaft der Software, ganz oder teilweise für neue Aufgaben wiederbenutzt zu werden.

1. Projekte verwalten mit Eclipse

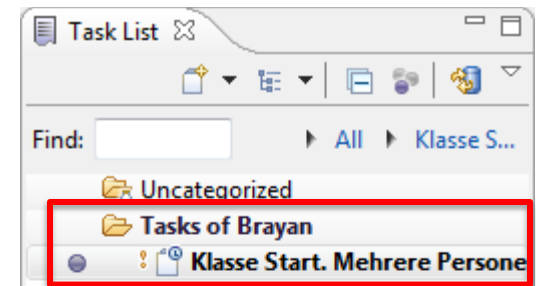
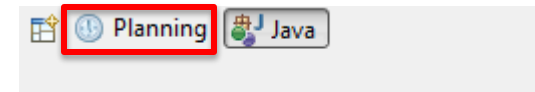
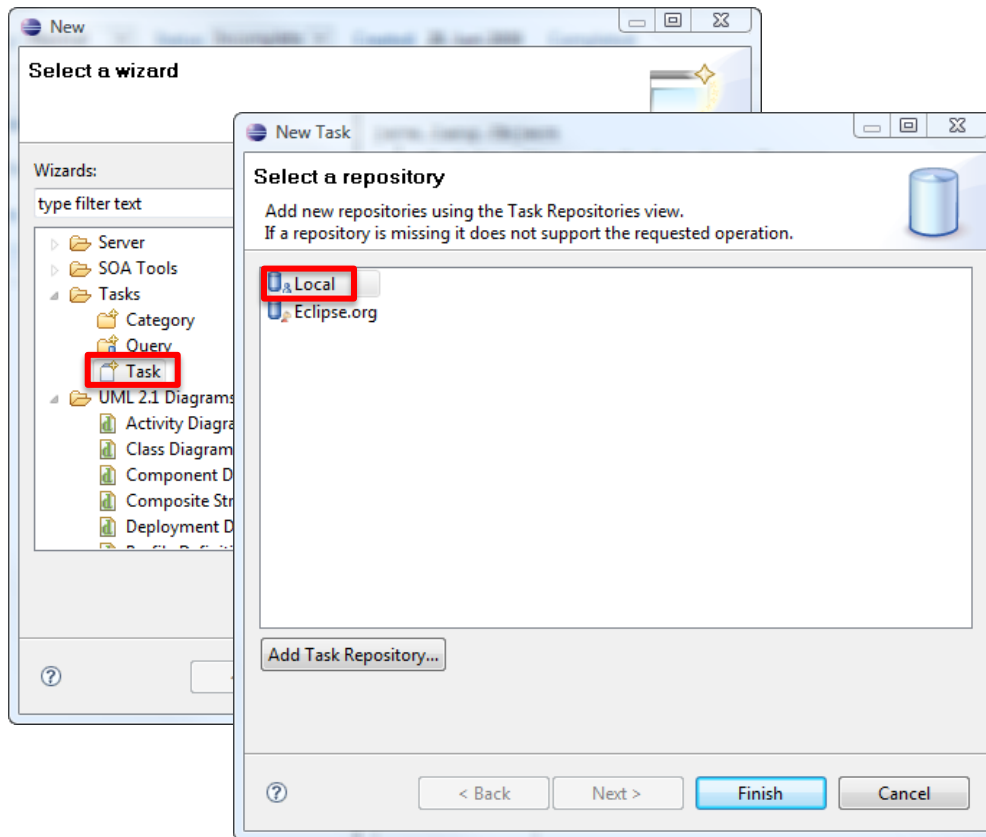
Projektorganisation



- Im Package Explorer (PE) werden alle Projekte, darunter fallen auch Web und andere Projekte, aufgelistet.
- Im Quell Ordner (src) reihen sich alle Pakete des Programms auf, die zu kompilieren sind.
- Wird die Klasse im PE aufgeklappt, erscheinen die dazugehörigen Attribute und Methoden. Vorstehend ein Symbol mit für den Zugriffsmodifikator. (private, package, public)
- Im Ordner JRE System Library [jre6] liegt die JRE. Die Java Version für welche entwickelt wird. In diesem Fall Java 6. Sie kann gewechselt werden.
- Alle weiteren Ordner können frei durch den Entwickler erstellt werden. Der Ordner „doc“ wurde erstellt durch JavaDocs.

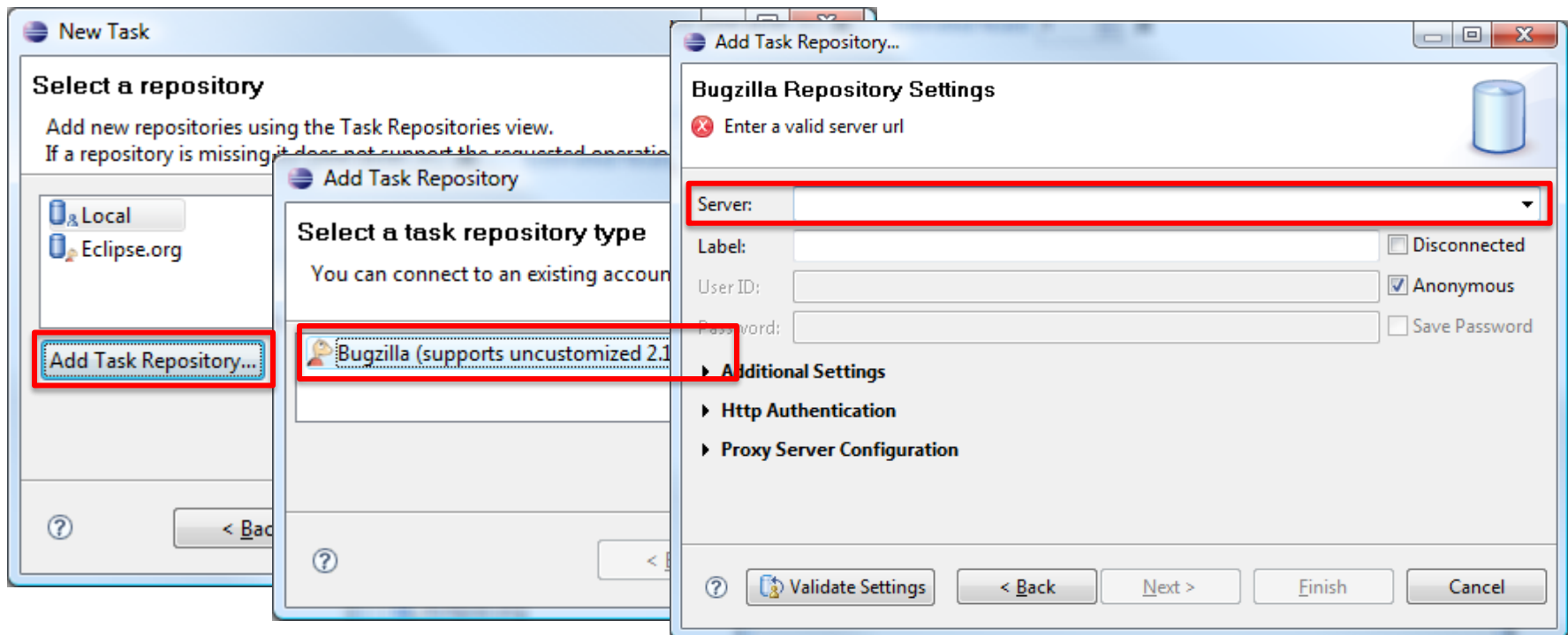
Task und Fehlerverwaltung - lokal

Die Perspektive Planning muss hinzugefügt werden, um die Task aufzulisten.



Task und Fehlerverwaltung - remote

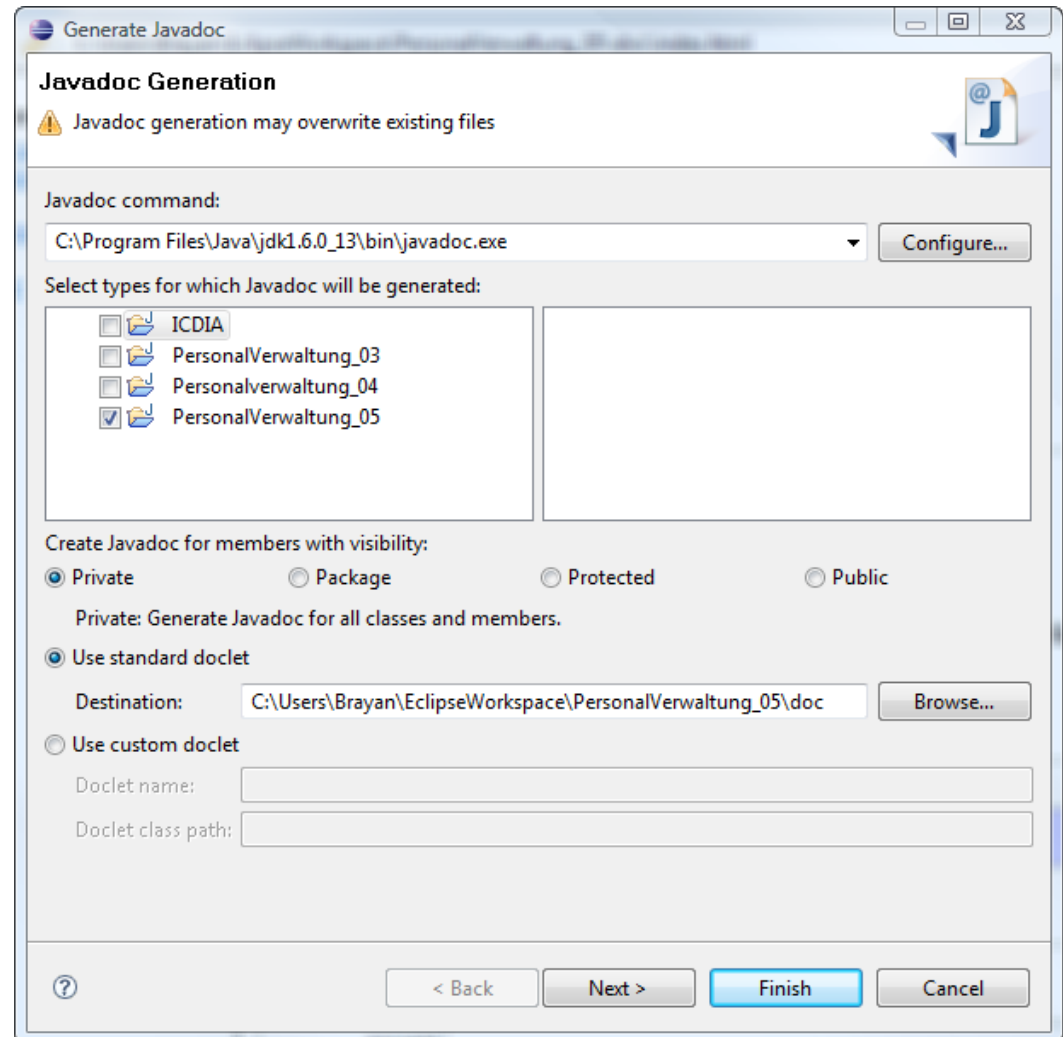
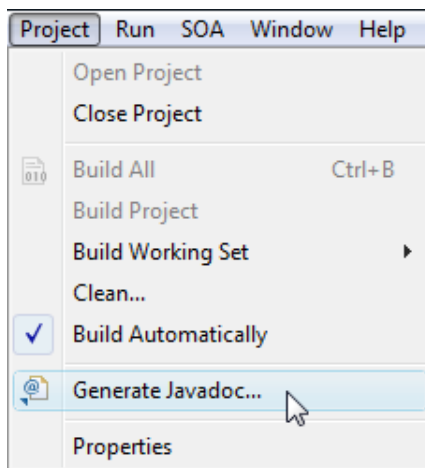
- [Bugzilla](#) hilft beim Festhalten und Verarbeiten von Fehlern. Damit trägt Bugzilla direkt zur Qualitätssicherung in Teams bei und fungiert in der Support Klasse von [ITIL](#) als [Problem Management](#) Werkzeug.
- Download: <http://www.bugzilla.org/download/#v32>
- Installation: <http://www.bugzilla.org/docs/3.2/en/html/installation.html>



2. Codieren mit Eclipse

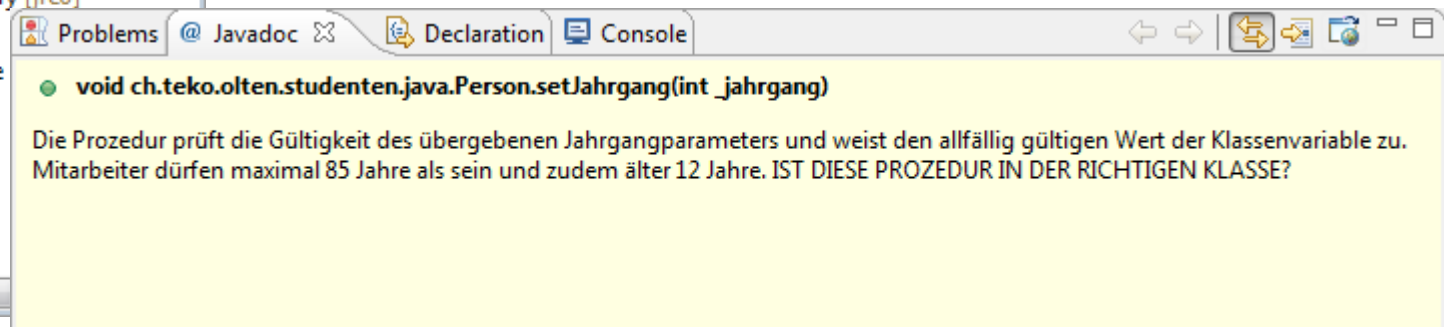
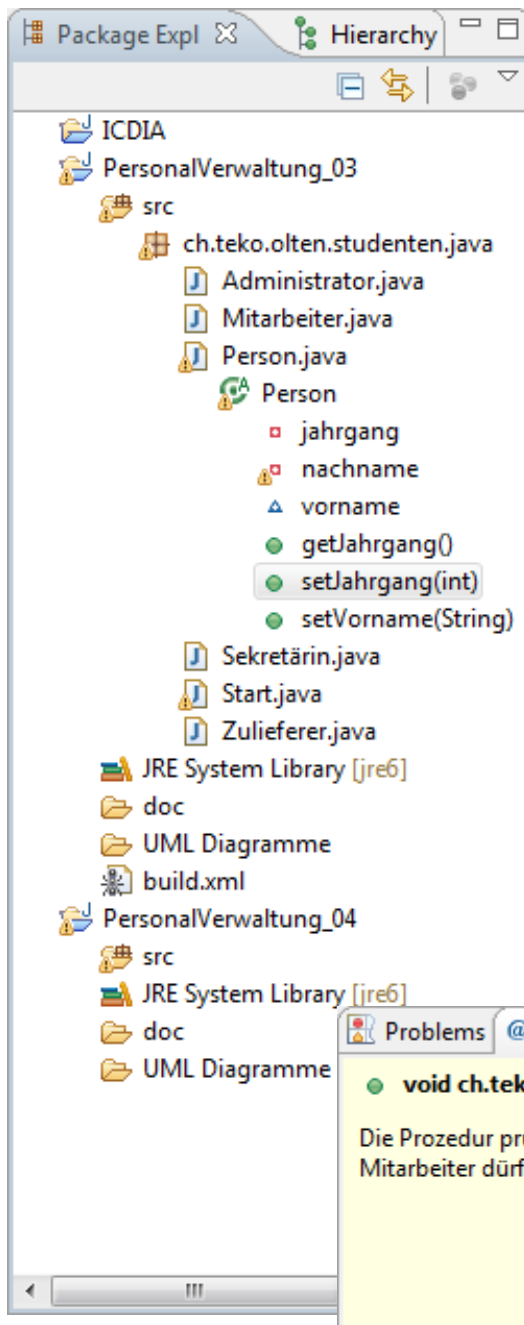
JavaDocs generieren

- Über das Menü Project können JavaDocs für den eigenen Code generiert werden.
- Wichtig ist, dass der Pfad zur javadoc.exe angegeben wird.
- **Vorsicht! Wird „Private“ nicht angewählt, werden private Attribute und Methoden nicht angezeigt.**



JavaDoc nutzen (1/2)

Die Bibliotheken, Klassen, Eigenschaften und Methoden, welche wir während der Entwicklungsphase benutzen, stellen meist eine Beschreibung zur Verfügung. Diese wird im Tab JavaDoc (unten) angezeigt und kann mit einem Klick auf ein Element im Package Explorer (links) gewechselt werden.



JavaDocs nutzen (2(2))

Die Anzeige eines Methodenbeschriebs kann auch mit einem Maus-Over erwirkt werden.

The screenshot shows an IDE window titled 'Person.java'. The code is as follows:

```

35      * */
36  public void setJahrgang(int _jahrgang) {
37      //rufe d
38      Calendar
39      SimpleDa
40
41      int minJahrgang = Integer.parseInt(sdf.format(c
42      int maxJahrgang = Integer.parseInt(sdf.format(c

```

A tooltip is displayed over the `setJahrgang` method signature, showing the following information:

- Method signature: `void ch.teko.olgen.studenten.java.Person.setJahrgang(int jahrgang)`
- Description: `Die Prozedur prüft die Gültigkeit des übergebenen Jahrgangparameters und weist den allfällig gültigen Wert der Klassenvariable zu. Mitarbeiter dürfen maximal 85 Jahre als sein und zudem älter 12 Jahre. IST DIESE PROZEDUR IN DER RICHTIGEN KLASSE?`
- Footer: `Press 'F2' for focus`

Insight und Declerations

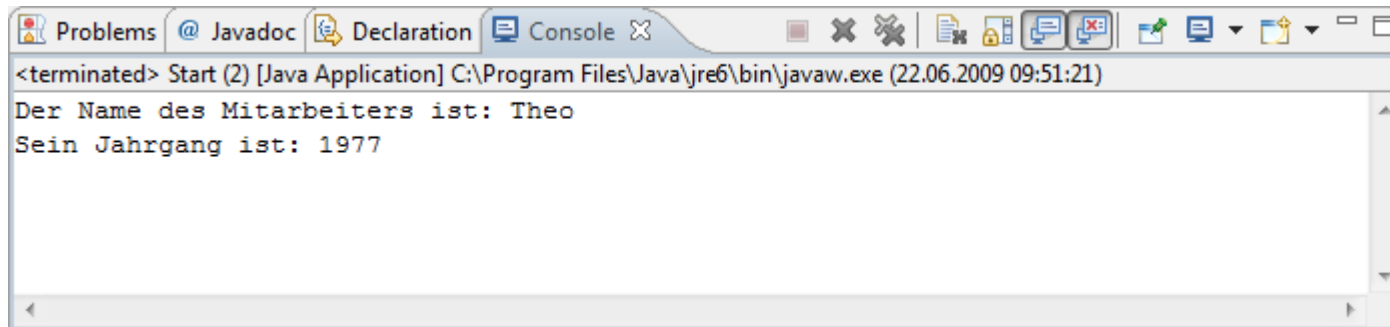
Mit einem Klick auf einen Bezeichner, in diesem Beispiel „jahrgang“, wird im Registerblatt Declaration die Deklaration angezeigt. Diese offenbart Datentyp und Zugriffsmodifikator. Wird eine Methode angeklickt erscheint die ganze Methodendeklaration.

```

Person.java
23 public void setVorname (String _vorname) {
24     vorname = _vorname;
25 }
26 /** Diese Funktion retourniert den in de
27 public int getJahrgang () {
28     return jahrgang;
29 }
30
Problems | Javadoc | Declaration | Console
int ch.teko.olten.studenten.java.Person.jahrgang
private int jahrgang;
  
```

The Console – Das Ausgabefenster

Die Konsole gibt Systemmeldungen aus. Darunter fallen vorwiegend Kompilierfehler, also Fehler zur Kompilierzeit, und Laufzeitfehler. Die Konsole kann aber auch mit eigenem Text, z. Bsp. zur Kontrolle, bedient werden.



The screenshot shows a console window with the following content:

```
<terminated> Start (2) [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (22.06.2009 09:51:21)  
Der Name des Mitarbeiters ist: Theo  
Sein Jahrgang ist: 1977
```

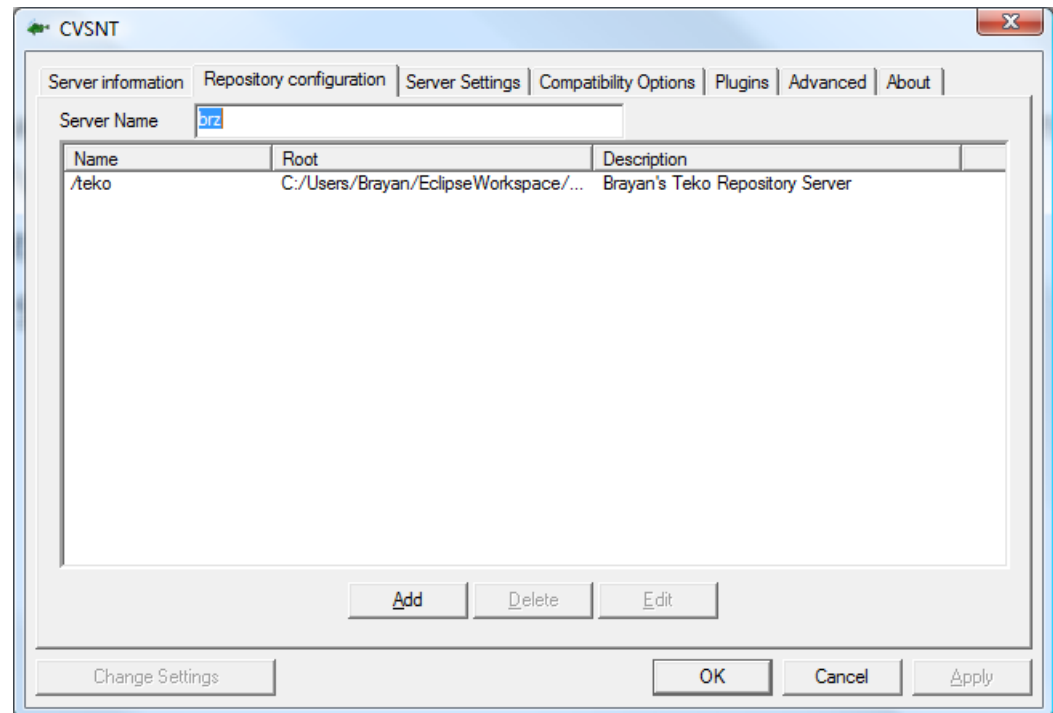
3. Team Work mit Eclipse

Versionierung im Team

- CVS steht für Concurrent Versioning System. Es dient der Versionierung von Quellcode einer Software oder mehreren Software-Paketen.
- Es existiert für jedes Betriebssystem mindestens ein CVS Server. Der CVS Client, in unserem Fall Eclipse, interessiert sich für das Betriebssystem des Servers nicht. Lediglich der Service (CVS) muss einwandfrei funktionieren.
 - Windows: [CVSNT](#), [CVSTortoise](#), WinCVS
 - Mac: [MacCVS](#)

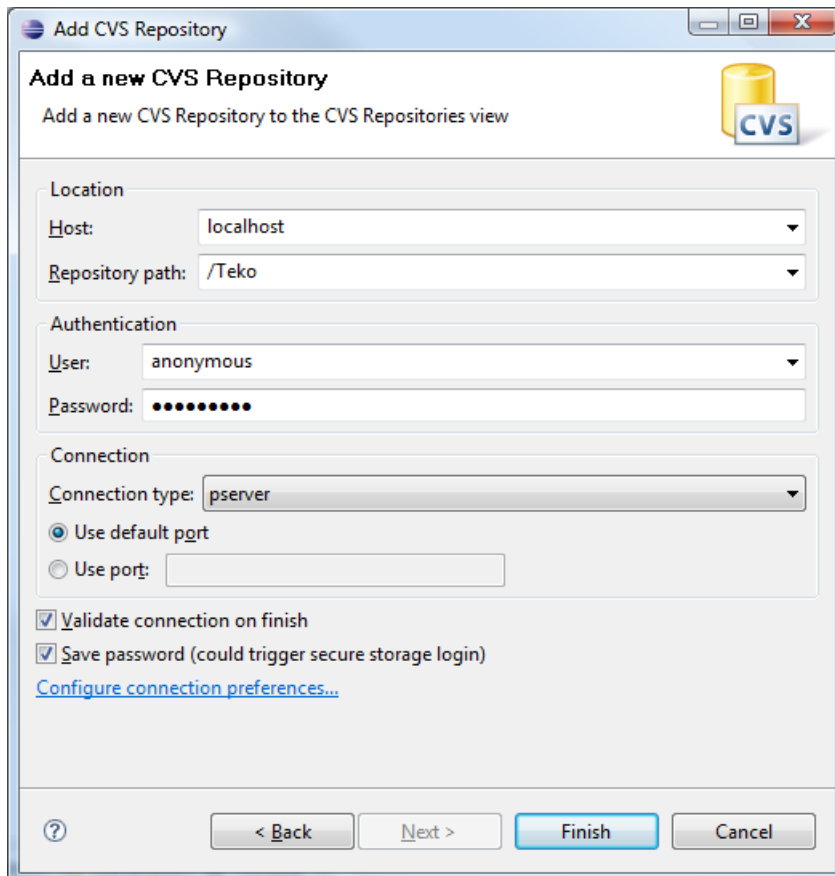
CVS - Server

- Die Firewall des Servers muss konfiguriert werden, um Anfragen auf dem TCP/IP Port 2401 entgegenzunehmen.
- Der Host (Server) sollte für alle Entwickler erreichbar sein.
- CVS verwalten Repositorys. Eine Art Container für Dateien.
- Ein CVS Server kann mehrere Repository verwalten.

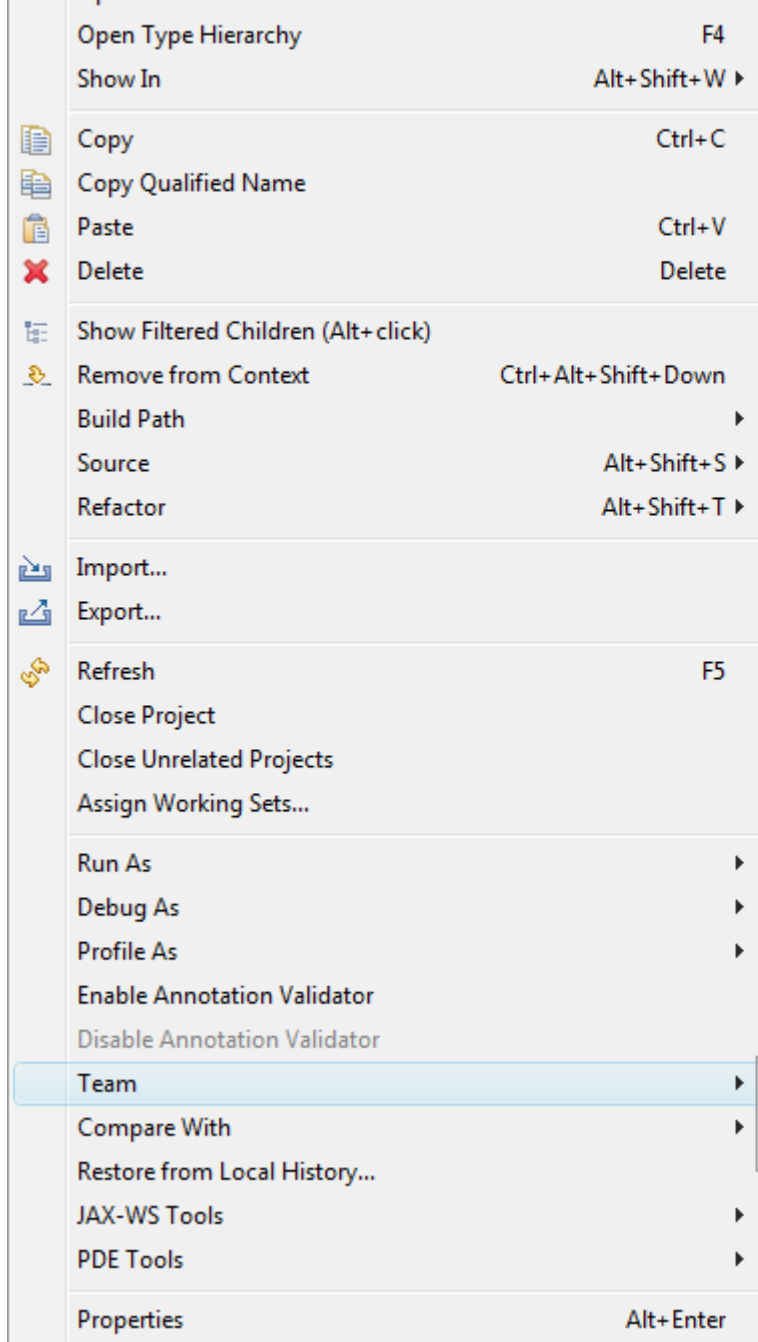


CVS mit Eclipse - Client

- Verbindung zu einem CVS Server mit Eclipse wird hergestellt über den Menüeintrag File, New ... , Other ... , CVS, Projects from CVS.



CVS - Client



- Ein bestehendes Projekt kann mit anderen geteilt (Share Project) und ins verbundene CVS Repository hochladen werden .