Repository

Für ein Backup der Daten ist jeder selbst verantwortlich, bei einem möglichen Ausfall des SVN-Servers müssen die Daten umgehend auf einem Datenträger eingereicht werden können.

Anders als in der Übung “Programmstrukturen 2” wird im Programmierpraktikum eine Repository-Struktur vorgegeben, wie man sie für größere Softwareprojekte verwendet. Grundsätzliches Prinzip ist dabei die Verwendung von drei verschiedenen Ordnern auf der obersten Ebene: `branches`, `tags` und `trunk`. Für das Programmierpraktikum geben wir dabei auch die zweite Ebene vor, so dass am Ende die folgende Ordnerstruktur entsteht:

```
[svn root]
 ├── branches
 │   └── binaries
 ├── tags
 │   ├── final
 │   ├── final-binaries
 │   └── intermediate
 └── trunk
```

- Im Ordner `trunk` findet die normale Entwicklung mit Netbeans oder einer anderen Entwicklungsumgebung statt. Bei größeren Softwareprojekten spricht man häufig von “Bleeding Edge”, weil dieser Entwicklungsstand weitestgehend ungetestet und damit möglicherweise instabil ist.
- Der Ordner `tags` dient der Versionierung von bestimmten Zeitpunkten im Laufe der Entwicklung. Jedes Mal, wenn man eine Version an einen Kunden ausliefert, legt man einen Tag mit dem exakten Zustand dieser Auslierung an. So kann man auch noch Jahre später nachvollziehen, welche Version des Quelltextes beim Kunden im Einsatz ist. Auch wenn es technisch möglich ist, darf ein einmalig angelegtes Tag nicht mehr verändert werden.
- `branches` (deutsch: Zweig) sind separat zu pflegende Zustände des Projektes. Die genaue Verwendung ist bei realen Projekten hochgradig unterschiedlich, in der Regel ist es im Gegensatz zu Tags aber erlaubt, in Branches auch Veränderungen vorzunehmen.

In unserem Fall gibt es genau zwei auszuliefernde Versionen, welche als Tags zu speichern sind: Die Zwischenpräsentation und die Abgabe. Im Branch `binaries` sollen kompilierte Zwischenstände mit z.B. allen benötigten Grafiken und `.jar` Dateien abgelegt werden. Daraus ergibt sich, dass im Normalfall also nur in dem `trunk` Ordner des Repositories gearbeitet wird. Sobald dann ein bestimmter Meilenstein erreicht ist, wird eine Kopie des Entwicklungsstandes für die Zwischenpräsentation oder die Abgabe angelegt.

trunk enthält

- den Quellcode in einem NetBeans-Projekt (Ausnahme s.u.) samt allen Entwicklungsumgebungseinstellungen und sonstigen notwendigen Dateien (z.B. icons). Der Projektname sollte auch den Nachnamen des Erstellers beinhalten. Nicht enthalten sein sollen Cache-Dateien oder beim Kompilieren entstandene Verzeichnisse (dist und build),
- ein Verzeichnis mit Dateien, die für den Programmablauf notwendig sind, z.B. Initialisierungsdateien, Spielstände, Logdateien oder Ähnliches.
- Bei der Bewertung wird dieser Ordner nicht betrachtet!

tags/intermediate enthält

- den bis dahin erstellten Quellcode in einem NetBeans-Projekt. Der Projektname muss hier auch den Nachnamen des Erstellers beinhalten. Nicht enthalten sein sollen Cache-Dateien oder beim Kompilieren entstandene Verzeichnisse (dist und build),
- gerne (aber nicht unbedingt) auch die Darstellung eines UML-Diagramms.

tags/final enthält

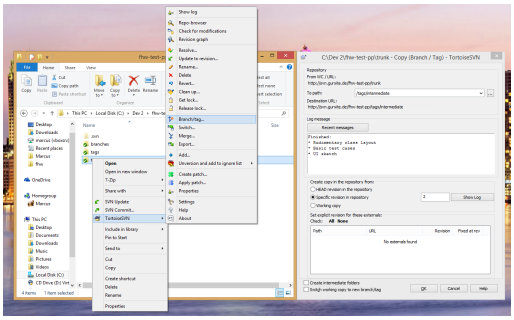
das NetBeans-Projekt in seinem finalen Zustand. Der Projektname muss hier auch den Nachnamen des Erstellers beinhalten. Es muss auf einem Rechner von RZ1-4 ohne weitere Installationen kompilierbar und ausführbar sein. Dieser Ordner ist die Basis für unsere Bewertung des Programmcodes. Sollte dieser Ordner nicht korrekt gefüllt sein, führt dies zu Punktabzug.

tags/final-binaries enthält

die jar-Datei mit allen benötigten Dateien wie Spiel-, Initialisierungs- oder Bilddateien eingebunden und die Dokumentation im PDF-Format. Die jar-Datei muss auf jedem beliebigen Rechner (mit installierter jre) ausführbar sein. Dieser Ordner ist die Basis für unsere Bewertung der Programmfunktionalität und der Dokumentation. Sollte dieser Ordner nicht korrekt gefüllt sein, führt dies zu Punktabzug.

Vorgehensweise

- Beim initialen SVN Checkout wird **ausschließlich** der `trunk` Ordner ausgecheckt. Hier kann ganz normal gearbeitet werden, bis die erste Versionierung notwendig ist.
- Das Anlegen eines Tags oder Branches lässt sich mit Tortoise SVN sehr einfach bewerkstelligen: Macht einen Rechtsklick auf Euren lokalen Ordner, der sich auf den `trunk` Ordner bezieht, und wählt dann den Menüpunkt `Branch/tag`. Es öffnet sich dann der abgebildete Dialog: Hier wird der Zielpfad für die Operation ausgewählt und es sollte eine kurze Nachricht angegeben werden. Ganz wichtig: Der Zweig oder Tag sollte sich **immer** auf eine bereits im Repository vorhandene Version beziehen. Macht also erst einen commit im `trunk` und dann den Zweig bzw. den Tag.



Arbeit mit anderen Entwicklungsumgebungen

Das Projekt im trunk darf mit jeder beliebigen Entwicklungsumgebung erstellt und bearbeitet werden.

Es ist dann jedoch kein 'svn tag' bzw. 'svn branch' möglich, da die Abgabe in einem NetBeans-Projekt erfolgen muss. Zur Abgabe muss in diesem Fall im jeweiligen tag- bzw. branch-Verzeichnis ein NetBeans-Projekt erstellt werden und die bis dahin mit Eclipse, IntelliJ oder einer anderen Entwicklungsumgebung in 'trunk' erarbeiteten Klassen per 'svn copy' hineinkopiert werden.

Weitere Info

Wer sich noch weiter mit möglichen Organisationsformen von Repositories beschäftigen möchte, verwendet dazu am Besten das SVN Book (das Buch ist auch in Deutsch zu finden):

- **Branch maintenance: trunk, branches and tags**
- **Branching and merging**

Kontakt FH Wedel

Staatlich anerkannte
Fachhochschule Wedel
Gemeinnützige Schul-
gesellschaft mbH
Feldstraße 143
D - 22880 Wedel
Tel.: +49 (0)4103 - 8048-0
Fax: +49 (0)4103 - 8048-39
E-Mail: sekretariat@fh-wedel.de