

SE2, Infoblatt zu Aufgabenblatt 2

Modul: Softwareentwicklung II – Sommersemester 2020

Anlegen von Templates in Eclipse, JavaDoc das Vertragsmodell beibringen

Erklärung

Im Aufgabenblatt 2 bringen wir euch das Vertragsmodell näher. Die pragmatische Umsetzung dieses Modells in Java geschieht u.a. durch `assert`-Anweisungen, was einiges an (leider gerade zu Anfang eher lästiger) Schreibarbeit bedeutet. Mit Eclipse könnt ihr euch Arbeit ersparen, indem ihr z.B. für die Prüfung der Vorbedingungen so genannte *Templates* anlegt. Templates sind Zeichenfolgen, die mit der **Tastenkombination *Strg+Leertaste*** in ein vorgefertigtes Code-Gerüst umgewandelt werden können.

Durchführung

Startet Eclipse und ruft über *Window>Preferences>Java>Editor>Templates* die Template-Optionen auf. Dort klickt ihr auf *New*, um ein neues Template anzulegen.

Im neu aufgegangenen Fenster vergebt ihr als erstes einen Namen für das Template. Dieser Name ist die Zeichenfolge, über die das Template im Editor aktiviert werden kann. Für eine Vorbedingung hat sich bei SE2 *pre* für *Precondition*, also Vorbedingung, eingebürgert.

Als nächstes vergebt ihr eine Beschreibung für das Template. Dieses Feld muss nicht ausgefüllt werden, es erleichtert jedoch das Wiederfinden und gibt eine schnelle Erklärung für die Funktion des Templates.

Das Listenfeld *Context* stellt ihr auf *java*. Damit legt ihr fest, dass dieses Template für Java-Quellcode benutzt wird. Setzt auch den Haken bei *Automatically insert*. Damit müsst ihr nach Betätigen der Tastenkombination *Strg+Leertaste* nicht noch eine extra Auswahl bestätigen.

In das große Feld kommt jetzt das eigentliche Template. Tragt dort folgenden Code ein:

```
assert ${bedingung} : "Vorbedingung verletzt: ${bedingung}";
```

Das `${bedingung}` bezeichnet eine Template-Variable mit dem Namen *bedingung*. Wenn ihr das Template in einem Methodenrumpf aktiviert, wird der Cursor an dieser Stelle stehen bleiben und ihr könnt die jeweilige Vorbedingung eingeben.

Sonderfall

Ein besonders häufig auftretender Fall ist der, dass ein Referenzparameter nicht `null` sein darf. Dafür lohnt es sich, ein extra Template anzulegen. Als Namen wählt *prenn*, was für *Precondition not null* steht. Hier tragt ihr nun folgenden Code ein:

```
assert ${localVar} != null : "Vorbedingung verletzt: ${localVar} ist null";
```

JavaDoc das Vertragsmodell beibringen

Da wir nun nicht nur Java ausführen möchten, sondern auch fortgeschrittene Funktionen bei der Code-Erstellung und -Kompilierung nutzen möchten, ist zusätzlich das Java SE Development Kit (JDK) erforderlich, das unter <https://adoptopenjdk.net/> zur Verfügung steht. Die JVM ist egak, wichtig ist nur, dass ihr Java 8 auswählt. Unter *Window>Preferences>Java>Editor>Installed JREs* könnt ihr über *Add>Standard VM>Next* und dort unter JRE Home den Unterordner „jre“ der JDK angeben und auf *Finish* gehen. Dann noch in der Liste den Haken für den neuen Eintrag setzen und auf *Apply and Close* klicken.

Damit im JavaDoc die von uns vorgeschlagenen Tags auch richtig angezeigt werden, müsst ihr noch folgende Einstellung vornehmen:

1. Project>Generate Javadoc
2. Javadoc Command: C:\<Pfad zum JDK>AdoptOpenJDK\bin\javadoc.exe (Windows)
<Pfad zum JDK> ist ein Platzhalter für den konkreten Pfad zum JDK auf eurem Rechner. Öffnet den WindowsExplorer und führt eine Suche nach `javadoc.exe` in C:\Programme\ aus, um den passenden Pfad zu ermitteln.
3. Projekt auswählen
4. Setzt die Sichtbarkeit auf Package
5. danach 2-mal Next klicken
6. Extra Javadoc option:

- -tag require:a:"Requires: "
- -tag ensure:a:"Ensures: "
- -charset UTF-8