



## Eclipse-Plugin zum Annotieren von Java-Dateien

Bakkalaureatsaufgabe für Loidl Stefan

Matr.Nr.: 0255886

Der Übungsbetrieb am Institut hat einen festen Ablauf:

1. Ausgabe der Übung auf Papier (und elektronisch)
2. Ausarbeiten der Übung durch den Studenten
3. Abgabe der Übung auf Papier (und elektronisch)
4. Kontrolle der Übung durch den Tutor
  - a. Prüfen der Übung auf Richtigkeit (kompilieren, testen)
  - b. Anmerken der Fehler in der Papierversion
5. Rückgabe der Papierabgabe an die Studenten

Bei diesem Ablauf gibt es mehrere (mögliche) Problemstellen:

- Die Papierabgabe geht auf einem der vielen Wege verloren
- Der Student gibt auf Papier und elektronisch verschiedene Versionen ab
- Der Student erkennt nach der Abgabe der Papierversion einen Fehler in seiner Ausarbeitung und möchte ihn gerne beheben (das ist komplizierter als bei der elektronischen Abgabe).
- Die Papierabgabe bedeutet Zeitaufwand für Studenten (ausdrucken, am Institut abgeben, kontrollierte Version am Institut abholen), Tutoren (abholen, abgeben am Institut, vergleichen ob die Papier und elektronische Abgabe identisch sind) und Übungsleiter (zeitgerecht aus dem Postkasten entfernen, sortieren der Übungen, ...)
- Unmengen an „verschwendetem“ Papier

Viele dieser Probleme könnten mit einem rein elektronischen Abgabesystem gelöst werden, dafür sind aber mehrere Komponenten nötig. Eine dieser Komponenten muss die Möglichkeit zur Verfügung stellen die abgegebenen Quelldateien zu annotieren (Fehler/Bemerkung, Punkteabzug).

### Idee

Um nicht ein komplett neues Programm entwickeln zu müssen, soll die Infrastruktur von Eclipse genutzt werden. Die Vorteile sind ein geringerer Implementierungsaufwand für Sie und ein geringer Einarbeitungsaufwand für die Benutzer.

Die Annotierung soll für einen beliebigen Bereich im Quellcode möglich sein und einen Text und eine Zahl beinhalten. Als Text wird vom Tutor eine Bemerkung eingetragen und als Zahl der Punkteabzug (oder Bonuspunkte). Diese Annotierungsinformation soll in einfacher Form (mensch- und maschinen-lesbar) in eine zusätzliche Datei gespeichert werden.

Die unten angehängten Bildschirmfotos sollen die Idee und den Ablauf verdeutlichen. Ihre Visualisierung muss nicht den Bildern entsprechen.

```
Test.java x
/*
 * Specification:
 * Write a program which writes the string "Hello World" to
 * System.out.
 */
package at.jku.ssw.ml.test;

/**
 * @author ml
 */
public class Test {

    public static void main(String[] args) {
        String[] bla = {"Hello", "World"};
        int len = 2;

        for (int i = 0; i < len; ++i) {
            System.out.print(bla[i] + " ");
        }
    }
}
```

Abbildung 1) Original vom Studenten

```
Test.java x
/*
 * Specification:
 * Write a program which writes the string "Hello World" to
 * System.out.
 */
package at.jku.ssw.ml.test;

/**
 * @author ml
 */
public class Test {

    public static void main(String[] args) {
        String[] bla = {"Hello", "World"};
        int len = 2;

        for (int i = 0; i < len; ++i) {
            System.out.print(bla[i] + " ");
        }
    }
}
```

Das Programm ist viel zu kompliziert, verwende doch einfach System.out.println(„Hello World“); „bla“ ist ein schlechter Name!
-4

Abbildung 2) Tutor beschreibt Programm

```

Test.java x
/*
 * Specification:
 *   Write a program which writes the string "Hello World" to
 *   System.out.
 */
package at.jku.ssw.ml.test;

/**
 * @author ml
 */
public class Test {

    public static void main(String[] args) {
        String[] bla = {"Hello", "World"};
        int len = 2;

        for (int i = 0; i < len; ++i) {
            System.out.print(bla[i] + " ");
        }
    }
}

```

Verwende bla.length anstelle von „2“  
-2

Abbildung 3) Tutor beschreibt Redundanz

```

Test.java x
/*
 * Specification:
 *   Write a program which writes the string "Hello World" to
 *   System.out.
 */
package at.jku.ssw.ml.test;

/**
 * @author ml
 */
public class Test {

    public static void main(String[] args) {
        String[] bla = {"Hello", "World"};
        int len = 2;

        for (int i = 0; i < len; ++i) {
            System.out.print(bla[i] + " ");
        }
    }
}

```

Das Programm ist viel zu ... (-4)  
Verwende bla.length anst ... (-2)

Abbildung 4) Fertig annotierter Quellcode

## Implementierung

Das Werkzeug ist in Java für die Eclipse 3 Plattform zu implementieren.

Der Fortgang der Arbeit ist in 14-tägigem Abstand mit dem Betreuer zu besprechen. Für die Ausarbeitung der schriftlichen Diplomarbeit sind die Richtlinien des Instituts für Systemsoftware zu beachten.

Nähere Auskünfte: Dipl.-Ing. Markus Löberbauer