



Software Engineering Übung 1

Programmverstehen, Nachdokumentation

1 Allgemeines

1.1 Daten zur Übung

- Ausgabe Di 18.09.2007
- Abgabe So 30.09.2007
- Besprechung am Fr 05.10.2007, 12:15 Uhr, BIN 2.A.01.

1.2 Formales

Abgegeben wird ein .zip-File mit den Lösungen als PDF und dem Source Code in einem Unterverzeichnis. Abgabe an stoiber@ifi.uzh.ch. Verspätete Abgaben werden nicht bewertet.

1.3 Gruppen

Die Übung ist in 3er Gruppen zu lösen. Falls die Aufgaben aufgeteilt wurden, muss klar ersichtlich sein, wer welchen Teil bearbeitet hat. Alle Gruppenmitglieder müssen über alle Teile Auskunft geben können.

1.4 Hinweise

Der Source Code zur Übung kann von der Übungsseite als .zip-File heruntergeladen werden. Auch die Entwicklungsrichtlinien für Java-Software finden Sie auf der Webseite.

2 Aufgabenstellung

2.1 Verstehen des Quellcodes (4 Punkte)

Dokumentieren Sie die logische Struktur der Ihnen vorliegenden Software. Stellen Sie die Vererbungshierarchien aller benutzten Klassen dar. Stellen Sie ebenso die Beziehungen der implementierten Klassen untereinander dar. Wählen Sie dazu eine geeignete Notation (zum Beispiel ein UML Klassendiagramm).

2.2 Reverse Engineering (4 Punkte)

Beschreiben Sie das abstrakte Verhalten der Klasse *CalculatorEngine*. Wählen Sie dazu eine geeignete Notation, zum Beispiel Zustandsautomaten bzw. Statecharts, Petrinetze, Interaktionsdiagramme und beschreiben Sie Ihr Modell ergänzend auch *kurz* in natürlicher Sprache. Aus dieser Dokumentation soll hervorgehen was die Aufgabe der genannten Klasse ist, welche Dienste sie anbietet und wie sie arbeitet.

2.3 Nachdokumentation im Code (6 Punkte)

Der gegebene Programmcode enthält bis auf die Klasse *CalculatorKeypad* keine Kommentare. Schreiben Sie für die Klassen *CalculatorEngine* und *CalculatorDisplay* die Klassen- und Methodenkomentare (Vor- und Nachbedingungen und Invarianten sind für diese Übung noch nicht gefordert). Hierbei helfen Ihnen die Ergebnisse aus Aufgabe 2.2. Halten Sie sich bei der Nachdokumentation an die Regelungen im mitgelieferten Java Styleguide. Identifizieren Sie ausserdem ungünstig gewählte Variablen- und Methodennamen in den genannten Klassen und benennen Sie diese um.

2.4 Fehlerbehebung und Erweiterung (6 Punkte)

Führende Nullen

Führen Sie die gegebene Anwendung aus und identifizieren Sie das Problem, dass der Kalkulator beim normalen Gebrauch führende Nullen anzeigt. Beheben Sie das Problem, so dass nur mehr bei einem Ergebnis gleich Null oder bei Betätigen der Clear-Taste eine Null erscheint.

Vorzeichenwechsel, Nachkommastellen

Identifizieren Sie das Problem, dass beim Drücken der +/- Taste (Vorzeichenwechsel) immer eine Nachkommastelle erscheint und eine weitere Eingabe einer Vorkommastelle verunmöglicht. Beheben Sie diesen Fehler.

Quadratwurzel und Quadrieren

Der gegebene Kalkulator verfügt über keine Funktionen zum Ziehen einer Quadratwurzel und zum Quadrieren einer Zahl. Erweitern Sie den Taschenrechner um diese beiden Funktionalitäten.