

1 Grundlegendes

Im Rahmen dieser Live-Aufgabe soll mit Hilfe des Composite-Patterns ein sehr stark vereinfachter Bug-Tracker abgebildet und in Java umgesetzt werden:

2 Erstellen Sie folgende Klassen/Interfaces

2.1 Bug

Entspricht einem einfachen Bug-Eintrag in unserem Tracker. Der Einfachheit halber enthält der Bug einen einfachen String in dem die Beschreibung („description“) abgelegt ist.

Außerdem noch das Feld „int estimate;“ welches den geschätzten Zeitbedarf (in Stunden) für den Bug repräsentiert.

2.2 Comment

Diese Klasse entspricht einem (stark vereinfachten) Kommentar. Der Kommentar enthält ebenfalls nur einen einfachen String, welcher den Text des Kommentars repräsentiert.

2.3 BugTracker

Ein BugTracker enthält das Field „children“ in welches Comments und Bugs eingetragen werden können.

3 Composite-Pattern

Wenden Sie nun das Composite Pattern an um die Hierarchie flexibler zu gestalten und die gemeinsame Funktionalität bereit zu stellen.

Führen Sie dazu folgendes ein:

3.1 BugTrackerEntity (Interface)

Dieses Interface bietet die gemeinsame Funktionalität aller in der Hierarchie vorhandener Klassen.

Dies sind folgende beiden Methoden:

int getEstimation() Liefert den geschätzten Zeitbedarf für das einzelne Entity innerhalb des Bugtrackers - also ohne die Zeiten der (eventuell vorhandenen Kindern). Comment und BugTracker selbst haben dabei immer einen „estimate“ von „0“

int getTotalEstimation() Liefert den geschätzten Gesamt-Aufwand - also inklusive der Zeiten aller Kinder und Kindeskindern usw.

3.2 Hierarchie

Die Klassen BugTracker, Bug und Comment sollen dabei in die Hierarchie aufgenommen werden. Dabei soll (mindestens) folgendes möglich sein:

BugTracker enthält Bugs

Bugs enthalten weitere Bugs Diese könnten verwandte/abhängige Bugs repräsentieren.

Bugs enthalten Comments

Comments enthalten weitere Comments

3.2.1 Wichtiger Hinweis

Sie müssen explizit **nicht** ausschließen, dass die Hierarchie auch fachlich „unsinnig“ aufgebaut wird. D.h. wenn der Benutzer z.B. einen BugTracker als Kind eines Comments hinzufügt, ist das ein fachliches Problem, welches nicht technisch verhindert werden muss.

Hinweise

Unit-Tests

Erstellen Sie ausreichend Unit-Tests mit den entsprechenden Assertions um die Korrektheit Ihrer Lösung beweisen zu können.

Composite-Pattern nutzen

Die Lösung ist nur dann korrekt, wenn sie mit Hilfe des Composite-Patterns umgesetzt wurde. Alternative Lösungen ohne das Composite-Pattern sind nicht gestattet.

Encoding

Bitte achten Sie darauf, dass Ihre Lösung im UTF-8-Encoding abgegeben wird.