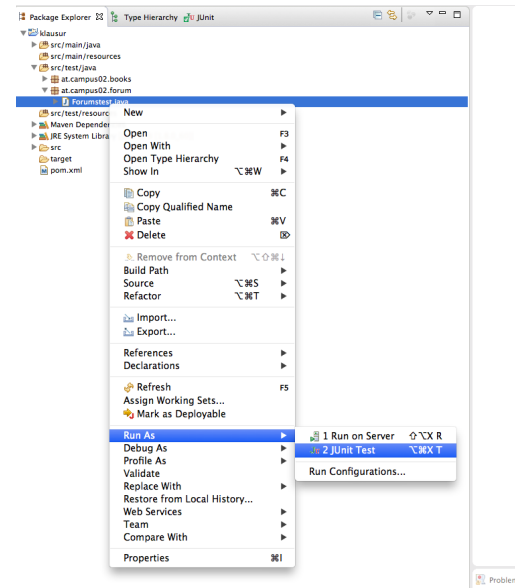


Allgemeines

Das Starter Kit der Klausur beinhaltet neben den Klassenrümpfen auch Testfälle. Diese können mit einem Klick mit der rechten Maustaste auf die Testklassen gestartet werden (siehe Screenshot). Die Testklassen dienen zur Überprüfung ob Sie die Implementierung korrekt gemacht haben und Sie können diese auch benutzen um die genauen Methodenaufrufe nachzuvollziehen (Debug).



Bewertung

Bei der Klausur sind in den beiden Beispielen insgesamt 16 Punkte zu erreichen, ab 8 Punkten ist man positiv wobei bei der Bewertung auch Teilpunkte auf eine Aufgabe vergeben werden können.

Viel Erfolg!

Beispiel Bibliothek

Konstruktor und toString (1 Pkt)

Implementieren Sie den Konstruktor der Klasse `Book` so, dass die Variablen gespeichert werden und schreiben Sie eine `toString` Methode, die in eckigen Klammern und durch einen Beistrich getrennt Autor, Titel und Seiten ausgibt. (Test: `BookTest.konstruktor`)

Bibliothek Seiten zählen (2 Pkt)

Implementieren Sie die Klasse `Bibliothek` so, dass sie Bücher speichern kann (`addBook()` Methode) und implementieren Sie die `countPages()` Methode so, dass die Seiten aller Bücher summiert werden. (Test: `BookTest.seiten()`)

Durchschnittliche Seitenanzahl (1 Pkt)

Implementieren Sie die Methode `avaragePages()` um die durchschnittliche Seitenanzahl aller Bücher in der Bibliothek zu berechnen (Test: `BookTest.durchschnitt()`).

Nach Autor suchen (2 Pkt)

Implementieren Sie die Methode `booksByAuthor(..)` so, dass in der zurückgegebenen Liste nur Bücher enthalten sind, die von dem angegebenen Autor sind. Die Methode muss auch mit einem `null`-String als Autor-Angabe umgehen können (Tests: `BookTest.author01`, `BookTest.author02`).

Allgemeine Suche (2 Pkt)

Implementieren Sie die `match(..)` Methode in der Klasse `Book` so, dass sowohl im Titel als auch im Autor der Klasse gesucht wird, ob die angegebene Zeichenkette enthalten ist (`String contains(..)`) und wenn sie enthalten ist `true` zurückgegeben wird. Implementieren Sie die `findBook` Methode in der `Bibliothek` so, dass die `match` Methode der Klasse `Book` herangezogen wird und nur die Bücher in der zurückgegebenen Liste stehen, wo die `match` Methode `true` zurück liefert.

Applikation (2 Pkt)

Erstellen Sie im Default Package eine Beispielapplikation `BookApp`, wo Sie 3 Bücher in die Bibliothek einfügen und die Anzahl der Seiten auf der Konsole ausgeben.

Beispiel Forum

Konstruktor und toString (1 Pkt)

Vervollständigen Sie den Konstruktor der Klasse `Forumseintrag` und implementieren Sie die `toString` Methode (Test: `Forumseintrag.konstruktor`)

Antworten (1 Pkt)

Implementieren Sie die Methode `antworten`. Dabei soll ein neuer `Forumseintrag` angelegt, intern in einer Liste gespeichert werden und dieser Eintrag zurückgegeben werden. Die Methode `getAntworten()` soll die Liste der Antworten zu einem Eintrag zurückliefern. (Test: `ForumseintragAntworten`)

Antworten zählen(2 Pkt)

Implementieren Sie die Methode `anzahlDerEintraege()` so, dass alle Antworten unterhalb des `Forumseintrages` zusammengezählt werden. (Test: `ForumseintragAntwortenZaehlen`)

Antworten Ausgeben (2 Pkt)

Implementieren Sie die `toString` Methode nun so, dass Antworten auf einen Eintrag in eckigen Klammern zusammengefasst an den Eintrag drangehängt werden. Wenn es keine Antworten gibt, sollen die eckigen Klammern nicht eingefügt werden. (Test: `ForumseintragAntwortenString`)