

Dieses Übungsblatt dient der Vorbereitung auf die Klausur. Die Lösungen der Aufgaben sollen nicht abgegeben werden. Die Aufgaben werden in den Übungen der Woche vom 14.5.-18.6.2017 besprochen.

Übungsblatt 4

Aufgabe 1

funktionsorientierter Test

Überlegen Sie, welche Aspekte beim Testen der Methode `gradeToString` beachtet werden müssen. Versuchen Sie, geeignete Äquivalenzklassen zu identifizieren und die ungefähre Zahl der notwendigen Testfälle abzuschätzen.

Die Methode `gradeToString` besitzt einen Parameter des Typs `double`. Die Methode soll den übergebenen Wert als Note interpretieren und die zugehörige textuelle Notenangabe als Ergebnis liefern. Die Methode hat die folgende JAVA-Signatur:

```
String gradeToString( double grade )
```

Zulässig sind Notenwerte zwischen einschließlich 1.0 bis einschließlich 5.0. Jeweils ab einem Nachkommawert von „.60“ wird die nächstschlechtere Notenangabe verwendet. Notenangaben sind: „sehr gut“, „gut“, „befriedigend“, „ausreichend“ und „mangelhaft“.

Aufgabe 2

funktionsorientierter Test

Überlegen Sie, welche Aspekte beim Testen der Methoden `isTriangle` und `contains` beachtet werden müssen. Versuchen Sie, geeignete Äquivalenzklassen zu identifizieren und die ungefähre Zahl der notwendigen Testfälle abzuschätzen.

- a) Die Methode `isTriangle` besitzt drei Parameter des Typs `double`, die die Kantenlängen der drei Seiten eines Dreiecks darstellen. Die Methode soll den Wert `true` liefern, falls die Kanten ein Dreieck bilden, und sonst `false`. Die Methode hat die folgende JAVA-Signatur:

```
boolean isTriangle( double a, double b, double c )
```

Hinweis: Geometrisch wird ein Dreieck durch drei Punkte definiert, die nicht auf einer Geraden liegen.

- b) Die Methode `contains` besitzt zwei Felder von `char`-Werten als Parameter. Der Aufruf der Methode `contains` soll `true` zurückgeben, falls die Folge aller Zeichen des Feldes `pattern` in genau der dort gegebenen Reihenfolge in `arr` enthalten ist. Sonst soll `false` zurückgegeben werden.

Die Methode hat die folgende JAVA-Signatur:

```
boolean contains ( char[] arr, char[] pattern )
```

Aufgabe 3

zustandsbasierter funktionsorientierter Test

Das Konzept eines binären Suchbaums ist bekannt:

Die Knoten eines binären Suchbaums enthalten miteinander vergleichbare Werte und alle Werte im linken Unterbaum sind kleiner als der Wert der Wurzel und alle Werte im rechten Unterbaum sind größer als der Wert der Wurzel sind. Für jeden Teilbaum gilt zudem, dass er wieder ein binärer Suchbaum ist.

Bereiten Sie für die folgenden beiden Methoden einen zustandsbasierten funktionsorientierten Test vor. Bestimmen Sie geeignete Äquivalenzklassen, die insbesondere möglichen Zustände des binären Suchbaums beachten, und charakterisieren Sie diese.

- a) Die Methode `Object minimum()` soll den kleinsten im zugehörigen binären Suchbaum abgelegten Wert als Ergebnis liefern.
- b) Die Methode `void add(Object o)` soll das Objekt `o` so in den zugehörigen binären Suchbaum einfügt, dass die Eigenschaften des binären Suchbaums erhalten bleiben. Falls `o` schon vorhanden ist, soll nicht eingefügt werden – also nichts geschehen.