Fachhochschule Köln
Cologne University of Applied Sciences

Fakultät für Informations-, Medien- und Elektrotechnik
Institut für Nachrichtentechnik

Prof. Dr. Hans W. Nissen
Tel.: (0221) 8275 – 2489
Email: hans.nissen@fh-koeln.de

Software Engineering

Praktikumsversuch 3, Gruppe C
- Anwesenheitsaufgaben -

Abgabe der Lösungen: Heute bis 14:30 Uhr in Ilias im Kurs zu dieser Veranstaltung -> Praktikum -> Gruppe C -> Ihr Team -> Blatt 3, als 1 zip-Datei mit dem Namen Anwesenheit.zip.

Ziel: Erstellung von JUnit-Testfällen

26 Fälle

4-5 leer

A 3.1 JUnit-Testfälle implementieren

Implementieren Sie JUnit-Testfälle für die von Ihnen implementierte Set zur Prüfung der folgenden Aussagen.

Dabei sollen Sie die Möglichkeiten von JUnit möglichst geschickt ausnutzen. Die Testfälle müssen in beliebiger Reihenfolge und vollkommen unabhängig voneinander ausführbar sein! Es müssen also Seiteneffekte bei der Ausführung der Testfälle vermieden werden.

Sie können die Testfälle beliebig gruppieren und auf geeignete Klassen verteilen.

1 a) get() auf leerer Set wirft die korrekte Exception.
2 b) insert() auf voller Set wirft die korrekte Exception.
1 c) delete() auf leerer Set wirft die korrekte Exception.
1 d) Wenn in eine leere Set ein neues Objekt der Klasse Person (mit Namen Peter) mit insert() geschrieben und get() ausgeführt wird, dann liefert get() das erstellte Objekt der Klasse Person; nach einem reset() ist die Set dann wieder leer. Der gleiche Fall soll auch jeweils mit Personen mit den Namen Klaus, Jim, Michael, Monika, Sabine geprüft werden. Hierzu soll ein parametrisierter Testfall erstellt werden!

-> Jedes leer

1 e) insert() funktioniert korrekt bei leerer Set
3 f) insert() funktioniert korrekt bei einer einelementigen Set
g) insert() wirft die korrekte Exception bei zweimaligem Eintragen des gleichen Objekts.
1 h) delete() funktioniert korrekt bei einer vollen Set
3 i) delete() funktioniert korrekt bei einer einelementigen Set
j) Set() liefert korrektes Set-Objekt
2 k) für eine volle Set liefert 20-maliges delete() eine leere Set
1 l) für eine leere Set liefert 20-maliges insert() verschiedener Objekte eine volle Set
1 m) isEmpty() funktioniert korrekt für leere Set
3 n) isEmpty() funktioniert korrekt für einelementige Set
2 o) isEmpty() funktioniert korrekt für volle Set
2 p) reset() funktioniert korrekt für volle Set
1 q) reset() funktioniert korrekt für leere Set
3 r) reset () funktioniert korrekt für einelementige Set
4 s) isFull() funktioniert korrekt für leere Set
0 t) isFull() funktioniert korrekt für einelementige Set
7 u) isFull() funktioniert korrekt für volle Set

A 3.2 Erstellen Sie eine TestSuite zur Durchführung aller Testfälle aus A 3.1.