

**Softwaretechnik SoSe 2012·Übungsblatt 9: Analytische Qualitätssicherung 1**  
Bearbeitung zum Tutorium in KW 24

Bereiten Sie Ihre **Lösungen** grundsätzlich so vor, dass Sie diese in der Übung Ihren Kommilitonen in geeigneter Form **zeigen** und **diskutieren** können. Geben Sie bitte stets Ihre verwendeten **Quellen** an.

**Aufgabe 9-1 (Begriffe)**

1. Erklären Sie den Unterschied zwischen Verifizieren und Validieren.
2. Woraus besteht ein Testfall? Wann nennt man einen Testfall erfolgreich?
3. Wie hängen Versagen, Fehler und Defekt zusammen?
4. Erklären Sie jeweils die Gemeinsamkeiten und Unterschiede zwischen
  - a. Strukturtest und Durchsicht
  - b. Lasttest und Stresstest
  - c. Testen und „Debugging“
  - d. Funktionstest und Akzeptanztest
  - e. Top-Down-Testen und Bottom-up-Testen

**Aufgabe 9-2 (Testtechniken anwenden und bewerten)**

Eine Routine `klassifiziereDreieck(int seite1, int seite2, int seite3)` soll zu einem durch seine Kanten gegebenen Dreieck berechnen, ob es rechtwinklig, gleichschenkelig, gleichseitig oder nichts von diesen drei ist. Das Ergebnis ist also entweder `Rechtwinklig`, `Gleichschenkelig`, `Gleichseitig` oder `Normal`

1. Erstellen Sie die Vorbedingung für `klassifiziereDreieck` in OCL.
2. Erstellen Sie mindestens sieben Testfälle allein aufgrund der Schnittstelle der Routine (black-box) durch Betrachtung unterschiedlicher Fälle mit verschiedenartigem Verhalten. Beschreiben Sie bei jedem Testfall kurz, wie Sie zu diesem gekommen sind.
3. Auf der zweiten Seite dieses Übungsblattes finden Sie eine Implementierung für `klassifiziereDreieck`. Erstellen Sie hiermit Testfälle aufgrund der Struktur des Programms (white-box). Bilden Sie dazu eine Menge von Testfällen, die dafür sorgen, dass jeder Programmzweig mindestens einmal durchlaufen wird (Zweigabdeckung, branch coverage,  $C_1$ ), d.h. jede Steuerbedingung (if-Fall) mindestens einmal wahr und einmal falsch wird.
4. Fügen Sie all die in 2. und 3. entstandenen Testfälle zusammen zu einer von Ihnen empfohlenen Testfallmenge. Haben Sie eine Anweisungsüberdeckung (statement coverage,  $C_0$ ) erreicht? Wie geeignet erscheinen Ihnen diese Überdeckungskriterien?
5. Welche Ihrer Testfälle sind erfolgreich?
6. Gibt es weitere Versagensmöglichkeiten, die die Testfälle nicht aufgedeckt haben? Wie haben Sie diese entdeckt? Wie nennt sich diese „Entdeckungstechnik“?

### Aufgabe 9-3 (Diskussion)

Eine der „goldenen“ Regeln im Bereich der Softwareentwicklung heißt „Ein Programmierer sollte nicht seinen eigenen Code testen“. Diskutieren Sie den Sinn dieser Regel.

Implementierung zu Aufgabe 9-2 3.:

```
1  enum DreieckArt {Rechtwinklig, Gleichschenklig, Gleichseitig, Normal}
2
3  DreieckArt klassifiziereDreieck (int seite1, int seite2, int seite3)
4  {
5      int quad1, quad2, quad3;
6      if ( (seite1 == seite2) || (seite2 == seite3) || (seite1 == seite3) )
7          return Gleichschenklig;
8      if ( (seite1 == seite2) && (seite2 == seite3) )
9          return Gleichseitig;
10     quad1 = seite1 * seite1;
11     quad2 = seite2 * seite2;
12     quad3 = seite3 * seite3;
13     if ( (quad3 + quad2 == quad3) ||
14         (quad1 + quad3 == quad2) ||
15         (quad3 + quad2 == quad1) )
16         return Rechtwinklig;
17     return Normal;
18 }
```