

Aufgabe 10-1: Begriffe

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

Aufgabe 10-2: Testtechniken anwenden und bewerten

Eine Operation `klassifiziereDreieck(int seite1, int seite2, int seite3)` soll für ein durch seine Kanten beschriebenes Dreieck berechnen, ob es Rechtwinklig, Gleichschenkelig, Gleichseitig oder Normal ist.

Behalten Sie Ihre *Begriffsdefinitionen* aus Aufgabe **10-1** im Hinterkopf und bearbeiten Sie die folgenden Teilaufgaben:

- a) Spezifizieren Sie die Vorbedingung für `klassifiziereDreieck` in OCL.
- b) Black-Box: Erstellen Sie mindestens sieben Testfälle allein aufgrund der *Schnittstelle* der Operation durch Betrachtung unterschiedlicher Fälle. Die Fälle sollten sich darin unterscheiden, dass Sie jeweils ein anderes Verhalten des Systems erwarten; vermeiden Sie unnötige Dopplungen.
Listen Sie die Testfälle tabellarisch auf beschreiben Sie bei jedem Testfall kurz, warum Sie diesen in die Testmenge aufgenommen haben.
- c) White-Box: Auf der zweiten Seite dieses Übungsblattes finden Sie eine Implementierung für `klassifiziereDreieck`. Erstellen Sie hiermit Testfälle aufgrund der *Struktur* des Programms. Bilden Sie dazu eine Menge von Testfällen, die dafür sorgt, dass jeder Programmzweig mindestens einmal durchlaufen wird (Zweigüberdeckung, branch coverage, C_1), d.h. jede Steuerbedingung (if-Fall) mindestens einmal wahr und einmal falsch wird.
- d) Fügen Sie all die in **b)** und **c)** entstandenen Testfälle zusammen zu einer Testfallmenge und führen Sie die Testfälle gedanklich aus.
Haben Sie eine Anweisungsüberdeckung (statement coverage, C_0) erreicht? Wie geeignet erscheinen Ihnen diese Überdeckungskriterien?
- e) Welche Ihrer Testfälle sind erfolgreich?
- f) Gibt es weitere Versagensmöglichkeiten, die die Testfälle nicht aufgedeckt haben? Wie haben Sie diese entdeckt? Wie nennt sich diese „Entdeckungstechnik“?

Aufgabe 10-3★: Diskussion

Eine der „goldenen“ Regeln im Bereich der Softwareentwicklung heißt „*Ein Programmierer sollte nicht seinen eigenen Code testen*“. Diskutieren Sie den Sinn dieser Regel; gehen Sie dabei auf mindestens zwei Argumente ein.

Implementierung zu Aufgabe 10-2, Teilaufgabe c)

```
1 class Math {
2     enum DreieckArt { Rechtwinklig, Gleichschenkelig, Gleichseitig, Normal }
3
4     DreieckArt klassifiziereDreieck (int seite1, int seite2, int seite3) {
5         int quad1, quad2, quad3;
6
7         if ((seite1 == seite2) || (seite2 == seite3) || (seite1 == seite3))
8             return DreieckArt.Gleichschenkelig;
9
10        if ((seite1 == seite2) && (seite2 == seite3))
11            return DreieckArt.Gleichseitig;
12
13        quad1 = seite1 * seite1;
14        quad2 = seite2 * seite2;
15        quad3 = seite3 * seite3;
16
17        if ((quad3 + quad2 == quad3) || (quad1 + quad3 == quad2) ||
18            (quad3 + quad2 == quad1))
19            return DreieckArt.Rechtwinklig;
20
21        return DreieckArt.Normal;
22    }
23 }
```