

Übung Softwarequalität TINF21 Übung Nr.7

Konstruktive Qualitätsmaßnahmen

Aufgabe 1:

Wie können Sie nachfolgendes C-Programm ergänzen, um eine vertragsbasierte Programmierung zu erreichen?

```
/* mathetricks: Einige Tricks mit Mathe */
/* Parameter: a: Multiplikator */
/*           b: Divisor (darf nicht 0 sein) */
/* Rückgabe: Die Rechnung, ist nie negativ! */
int mathe_tricks( int a, int b)
{
    int c;

    c = a/b;
    c *= a;
    c *= b;

    return c;
}
```

Testen

Aufgabe 2:

Sie geben die Entwicklung für ein Modul Ihrer Mitarbeiterverwaltungssoftware an ein externes Unternehmen ab. Sie spezifizieren die Funktionalität des Moduls folgendermaßen:

„Mitarbeiter erhalten ab einer Zugehörigkeit zur Firma von mehr als drei Jahren 50% des Monatsgehalts als Weihnachtsgratifikation. Mitarbeiter, die länger als fünf Jahre in der Firma tätig sind, erhalten 75%. Bei einer Firmenzugehörigkeit von mehr als acht Jahren wird eine Gratifikation von 100% gewährt“

Bilden Sie geeignete Äquivalenzklassen, um in einem Test festzustellen, ob Ihr Zulieferer das Modul korrekt implementiert hat.

Aufgabe 3:

Es ist folgende Funktion spezifiziert:

```
/* check_pin
 *
 * @param: pin - a 4 digit PIN number which is validated, only digits
 * are accepted
 * @returns: True → pin is valid, False → PIN is invalid
 */
boolean check_pin( String pin);
```

Bilden Sie geeignete Äquivalenzklassen, um zu zeigen, daß die Funktion korrekt funktioniert.

Aufgabe 4:

Gegeben ist folgende Funktion:

```
1  int bonus( int x, int y)
2  {
3      int ergebnis = 0;
4
5      if( x>0 && y>0 )
6          ergebnis = 30;
7
8      if( x>10 && y<10 )
9          ergebnis += 20;
10     else
11         ergebnis += 10;
12
13     return ergebnis;
14 }
```

- a) Zeichnen Sie den Kontrollflussgraphen und geben Sie alle Testfälle an, um eine c_0 -Abdeckung zu erreichen.
- b) Ermitteln Sie alle Testfälle, um eine c_1 -Abdeckung zu erreichen.

Aufgabe 5:

Gegeben ist folgende Funktion:

```
1  void output( int selection, int y)
2  {
3      if( selection == 1)
4          printf("\n 1");
5      else
6          if (selection ==2)
7              printf("\n 2");
8          else
9              if (selection ==3)
10                 printf("\n 3");
11             else
12                 if (selection ==4)
13                     if (y == 0)
14                     {
15                         printf("\n 4");
16                         printf("\n 5");
17                     }
18                     else
19                         printf("\n 6");
20                 else
21                     printf("\n 7");
22 }
```

- a) Zeichnen Sie den Kontrollflussgraphen und geben Sie alle Testfälle an, um eine c_0 -Abdeckung zu erreichen.
- b) Ermitteln Sie alle Testfälle, um eine c_1 -Abdeckung zu erreichen.
- b) Mit C_0 ist in diesem Fall auch C_1 abgedeckt