

## Übung Softwaretechnik II

Blatt 2 — Ausgabe am 29.04.2010, Besprechung ab 06.05.2010

### Aufgabe 2.1: Äquivalenzklassenbildung / Grenzwertanalyse

[ mittel / Klausur-Bearbeitungszeit ca. 10 Min.;  
tatsächlich: \_\_\_\_\_; vorbereitende Lektüre: \_\_\_\_\_ ]

Ein Stromversorgungsunternehmen bietet seinen Geschäftskunden einen verbrauchsabhängigen Staffeltarif mit Rabattoption gemäß folgender Spezifikation an:

Geschäftskunden sind Kunden, die 60 MWh oder mehr im Jahr verbrauchen. Solche Kunden erhalten bis einschließlich 80 MWh einen Tarif von €75 pro MWh. Ab 80 MWh bis einschließlich 100 MWh beträgt der Tarif €70 und ab 100 MWh €65. Kunden haben nur dann den Großkunden-Status, wenn sie mehr als 100 MWh verbrauchen. Nur diesen Kunden kann ein zusätzlicher Rabatt eingeräumt werden, der zwischen 0% und 20% betragen kann. Generell gilt, dass Kunden des Staffeltarifs nur Energie verbrauchen, nicht aber in das Stromnetz einspeisen können, die Verbrauchswerte folglich nie negativ sind.

Zur Berechnung der Endpreise nutzt das Unternehmen die Funktion `double endpreis(int verbrauch, double rabatt)`, die gegen ihre Spezifikation validiert werden soll.

1. Leiten Sie Äquivalenzklassen für gültige und ungültige Werte her.
2. Führen Sie eine Grenzwertanalyse durch und geben Sie geeignete Repräsentanten der einzelnen Äquivalenzklassen an.
3. Kombinieren Sie nun diese Repräsentanten zu Testfällen und geben Sie eine angemessene Testfallmenge an.

### Aufgabe 2.2: Kontrollflussbasierter Test

[ mittel / Klausur-Bearbeitungszeit ca. 10 Min.;  
tatsächlich: \_\_\_\_\_; vorbereitende Lektüre: \_\_\_\_\_ ]

Gegeben sei eine Funktion zur Berechnung der Fakultät einer Zahl  $x$ . Da die Fakultät nur für natürlichen Zahlen und Null definiert ist, wird der übergebene ganzzahlige Wert entsprechend überprüft und gegebenenfalls der Fehlercode -1 zurückgegeben.

```
int fakultaet(int x) {  
    int tmp = -1;  
    if(x >= 0) {  
        tmp = 1;  
        for(int i=2; i <= x; i++) {  
            tmp = tmp * i;  
        }  
    }  
    return tmp;  
}
```

1. Erstellen Sie den zur `fakultaet`-Funktion gehörigen Kontrollflussgraphen.
2. Geben Sie eine Menge von Testfällen (Pfade durch den Kontrollflussgraphen) an, die eine C0-Überdeckung von 100% sicherstellt.
3. Geben Sie nun eine Menge von Testfällen an, die eine C1-Überdeckung von 100% sicherstellt.
4. Können Sie eine 100%ige C $\infty$ -Überdeckung erreichen? Geben Sie eine Ihrer Meinung nach sinnvolle Menge von Testfällen an.