

Arbeitsblatt (3.11.2015)

White-Box-Test

In dieser Übung lernen Sie einen White-Box-Test durchzuführen, dazu:

- ✓ Leiten Sie den Kontrollflussgraph ab
- ✓ Bestimmen Sie die Anweisungsüberdeckung
- ✓ Bestimmen Sie die Zweigüberdeckung

Spezifikation des Programms (siehe letzte Vorlesung)

Das Programm soll Einkäufe in einem Einkaufswagen verwalten.

Man kann Produkte einkaufen. Ein Produkt hat einen Typ, einen Namen (beliebige Zeichenkette) und einen Preis (ganze Zahl nicht größer als 100). Im Moment werden nur die Typen „Kekse“ und „Waffeln“ unterstützt.

Beim Einkauf werden die Produkte in einen Einkaufswagen gelegt. Ein Einkaufswagen hat einen Namen (beliebige Zeichenkette). Es passen max. 10 Produkte in einen Einkaufswagen. Es können mehrere gleiche Produkte in einem Einkaufswagen liegen.

Das Programm bietet die folgenden Funktionen zur Verwaltung von Produkten und Einkaufswagen

- Einkaufswagen anlegen bzw. anzeigen
- Produkt (identifiziert durch Typ, Name und Preis) einkaufen bzw. aus einem Einkaufswagen wieder entfernen
- Gesamtpreis des Einkaufswagen errechnen und anzeigen
- 2 Einkaufswagen miteinander vergleichen. Dabei werden die Produkte angezeigt, die in beiden Einkaufswagen die gleichen sind.

Daraus ergeben sich die folgenden Funktionen mit Ein- und Ausgabetypen. Der Systemzustand (also die Menge der vorhandenen Einkaufswagen) ist nicht als Ein/Ausgabe zu behandeln.

- EAnlegen(Name) : EWagen
- EAnzeigen(Name des EWagens)
- Einkaufen(Typ, Name des Produktes, Preis des Produkts, Name des EWagen)
- EinkaufZurücklegen (Typ, Name des Produktes, Preis des Produkts, Name des EWagen)
- EPreis(Name des EWagens)
- EVergleich(Name1, Name2) : ProduktMultimenge

Aufgabe 1: Kontrollflussgraph für Produktvergleich

Die folgende Operation der Klasse `Product` berechnet, ob ein anderes Produkt das gleiche Produkt ist. Geben Sie den Kontrollflussgraph dieser Operation an.

```
1 public boolean equalsProduct(Object other) {
2     if (other == null) {
3         return false;
4     }
5     if (!(other instanceof Product)) {
6         return false;
7     }
8     // Required cast from Object to Product, because an Object has
9     // no getName() or getPrice() operation
10    Product product2 = (Product) other;
11    if (this.getName().equals(product2.getName())) {
12        if (this.getPrice() == product2.getPrice()) {
13            return true;
14        }
15    }
16    return false;
17}
```

Aufgabe 2: Anweisungsüberdeckung

Geben Sie eine Menge von Testfällen (durch Angabe der Eingabe und der Ausgabe) an, die eine vollständige Anweisungsüberdeckung erreicht.

Aufgabe 3: Zweigüberdeckung

Welche Kanten werden durch die obige Menge nicht abgedeckt? Geben Sie eine Menge von Testfällen an, die eine vollständige Zweigüberdeckung erreicht.