

Übung 2

Übungsaufgaben für den 5. November 2001

Gegeben ist die in Java implementierte Klasse *OrderedPair*.

```
class OrderedPair {
    private int global_a;
    private int global_b;
    public OrderedPair(int a,int b) {
        order(a,b);
    }
    private void order(int a,int b) {
        int tmp;
        if (a>b) {
            tmp = a;
            a = b;
            b = tmp;
        }
        global_a=a;
        global_b=b;
    }
}
```

Aufgabe A1 Kontrollflußorientierte Testverfahren

- Erstellen Sie einen Kontrollflußgraphen für die Operation *order(int a,int b)* der Klasse *OrderedPair*.
- Ermitteln Sie für die Operation *order(int a,int b)* der Klasse *OrderedPair* Testfälle für die Erfüllung der Anweisungsüberdeckung.
- Ermitteln Sie für die Operation *order(int a,int b)* der Klasse *OrderedPair* Testfälle für die Erfüllung der Zweigüberdeckung.

Aufgabe A2 Datenflußorientierte Testverfahren

- Modifizieren Sie den Kontrollflußgraphen der Operation *order(int a,int b)* der Klasse *OrderedPair* für die Durchführung datenflußorientierter Testverfahren.
- Ermitteln Sie für die Operation *order(int a,int b)* der Klasse *OrderedPair* Testfälle für die Erfüllung des all defs-Test. Geben Sie die geforderten Testpfade an und markieren Sie diese im Kontrollflußgraphen.
- Ermitteln Sie für die Operation *order(int a,int b)* der Klasse *OrderedPair* Testfälle für die Erfüllung des all p-uses-Test. Geben Sie die geforderten Testpfade an und markieren Sie diese im Kontrollflußgraphen.
- Ermitteln Sie für die Operation *order(int a,int b)* der Klasse *OrderedPair* Testfälle für die Erfüllung des all c-uses-Test. Geben Sie die geforderten Testpfade an und markieren Sie diese im Kontrollflußgraphen.

Übung 2

Gegeben ist die in Java implementierte Klasse *CharString*.

```
class CharString {
    private char[] string;
    public CharString(char[] string) {
        init(string);
    }
    private void init(char[] string) {
        this.string=(char[])string.clone();
    }
    public boolean arrayEquals(char[] comp) {
        boolean result;
        if (comp.length!=string.length) {
            result = false;
        } else {
            result = true;
            for (int i=0;i<string.length&&result==true;i++) {
                if (string[i]!=comp[i]) {
                    result = false;
                }
            }
        }
        return result;
    }
}
```

Aufgabe B1 Kontrollflußorientierte Testverfahren

- Erstellen Sie einen Kontrollflußgraphen für die Operation *arrayEquals(char[] comp)* der Klasse *CharString*.
- Ermitteln Sie für die Operation *arrayEquals(char[] comp)* der Klasse *CharString* Testfälle für die Erfüllung der Anweisungsüberdeckung.
- Ermitteln Sie für die Operation *arrayEquals(char[] comp)* der Klasse *CharString* Testfälle für die Erfüllung der Zweigüberdeckung.

Aufgabe B2 Pfadtest

- Beurteilen Sie die Anwendbarkeit des vollständigen Pfadüberdeckungstest für die Operation *arrayEquals(char[] comp)* der Klasse *CharString*.
- Die Klasse *CharString* ist Teil eines Programms, in dem nachweislich nur char-Arrays der Größe 10 verwendet werden. Beurteilen Sie erneut die Anwendbarkeit des vollständigen Pfadüberdeckungstests.
- Ermitteln Sie für die Operation *arrayEquals(char[] comp)* der Klasse *CharString* Testfälle für die Erfüllung des strukturierten Pfadüberdeckungstests für den Parameter $k=1$.
- Ermitteln Sie für die Operation *arrayEquals(char[] comp)* der Klasse *CharString* Testfälle für die Erfüllung des Boundary interior-Test.



Übung 2

Bei Fragen besteht jederzeit die Möglichkeit, sich an den zuständigen Wissenschaftlichen Mitarbeiter, Christopher Robinson-Mallett (Raum A2.11), zu wenden.