

## Übung 4

---

### Übungsaufgaben für den 20. November 2001

Bei Fragen besteht jederzeit die Möglichkeit, sich an den zuständigen wissenschaftlichen Mitarbeiter, Bernhard Kaiser (Raum A2.11), zu wenden. (Tel. (0331) 5509-158, email: [bernhard.kaiser@hpi.uni-potsdam.de](mailto:bernhard.kaiser@hpi.uni-potsdam.de)).

#### Aufgabe 1 - Datenflussanalyse

Das folgende Beispiel zeigt einen in der Sprache C sehr häufigen Fehler.

- Welcher Irrtum wurde begangen?
- Welches Fehlverhalten wird sich bei der Ausführung zeigen?
- Wie äußert sich der Fehler bei einer Datenflussanalyse?
- Können alle Fehler dieser Art durch Datenflussanalyse gefunden werden?  
Hinweis: Betrachten Sie die Beispiele aus der Vorlesung!
- Glauben Sie, dass man durch andere Formen der automatisierten statischen Analyse (nicht datenflussorientiert) solche Fehler zuverlässig entdecken kann?  
Stellen Sie ggf. eine Regel zum Auffinden auf!

Anmerkung: Die Sprache C kennt keinen Typ `boolean` und benutzt stattdessen Ganzzahltypen (`char`, `integer`, `long`), wobei 0 für `FALSE` und alles andere (typisch 1) für `TRUE` steht. Deshalb wird am Anfang ein Typ `boolean` selbst definiert.

```
typedef enum {FALSE = 0, TRUE = 1} boolean;

boolean isVocal(char Zchn)
{
    boolean returnValue;

    if((Zchn=='A') || (Zchn=='E') || (Zchn=='I') ||
        (Zchn == 'O') || (Zchn == 'U'))
    {
        returnValue = TRUE;
    }
    else
    {
        returnValue = FALSE;
    }

    return returnValue;
} // end isVocal()
```

## Übung 4

---

### Aufgabe 2 - Datenflussanalyse

Analysieren Sie das folgende Code-Beispiel auf Datenflussanomalien!

Falls Sie welche finden,

- Welche Auswirkung auf die korrekte Ausführung des Codes haben sie?
- Welche Auswirkungen auf die Testfälle für einen all-defs-Test haben sie?
- Welche Gefahr besteht beim Weglassen einer redundanten Variablendefinition?
- Halten Sie den Code im Beispiel für qualitativ akzeptabel?

```
double Sqrt(double X)
{
    double returnValue = 0;
    double W = 0;

    if (X > 0.0)
    {
        while (ABS(W*W-X) > 0.01)
        {
            W = W - ((W*W-X) / (2.0 * W));
        } // end while
        returnValue = W;
    } // end if
    return (returnValue);
} // end Sqrt()
```

### Aufgabe 3 - Review und Inspektion

Welche der folgenden Probleme können **nur** durch Reviews / Inspektionen gefunden werden?

Welche können **nicht** unmittelbar durch Reviews / Inspektionen gefunden werden?

- Nichtssagende Variablennamen
- Datenflussanomalien
- Fehlverhalten
- Fehler
- Unvollständige Umsetzung der Spezifikation
- zu wenige Kommentarzeilen pro Programmtextzeilen
- Unverständliche Kommentare
- Nichteingehaltene Stilkonventionen