Informatik I Übung, Woche 44

Giuseppe Accaputo

30. Oktober, 2014

Plan für heute

1. Lernziele für die heutige Übungsstunde
2. Nachbesprechung Übung 6
3. Vorbesprechung Übung 7

Lernziele

- Vor- und Nachbedingungen
- Multidimensionale Arrays

Vorbedingung (Precondition)

Definition: Die Vorbedingung einer Funktion oder eines Programms gibt an, unter welchen Voraussetzungen das Verhalten der Funktion definiert ist

Nachbedingung (Postcondition)

Definition: Die Nachbedingungen einer Funktion oder eines Programms geben an, welche Aussagen nach der Ausführung gelten müssen.

Konzept von Vor- und Nachbedingungen

Wenn die Vorbedingung gilt, so müssen nach Ausführung der Funktion alle Nachbedingungen erfüllt sein, sonst ist das Programm nicht korrekt. Vor- und Nachbedingungen machen Aussagen über die Semantik des Programms aus.

Quiz-Frage 1

```
FUNCTION IndexOfMin (i : INTEGER) : INTEGER;
{ PRE: i ist vom Typ INTEGER }
...
```

Frage: Ist dies eine gute Vorbedingung?

Quiz-Frage 1

```
PROCEDURE IndexOfMin (arr : ARRAY OF INTEGER
    ↪ );
{ PRE: i ist vom Typ INTEGER }
...
```

Frage: Ist dies eine gute Vorbedingung?

Antwort: Nein, denn die Vorbedingung bezieht sich auf die Syntax und nicht auf die Semantik. Eine gute Vorbedingung wäre beispielsweise: `arr` sollte nicht leer sein, d.h. `Length(arr)> 0`.

Teamarbeit: Vorbedingung und Nachbedingung für die `Sqrt`-Funktion definieren

Aufgabe: Definiert für die Implementation der `Sqrt`-Funktion ($\sqrt{x}$ = `Sqrt(x)`) die Vor- und Nachbedingung. `Sqrt` arbeitet dabei nur mit nichtnegativen, reellen Zahlen.

Teamarbeit: Vorbedingung und Nachbedingung für die `Sqrt`-Funktion definieren

Aufgabe: Definiert für die Implementation der `Sqrt`-Funktion ($\sqrt{x}$ = `Sqrt(x)`) die Vor- und Nachbedingung. `Sqrt` arbeitet dabei nur mit nichtnegativen, reellen Zahlen.

Mögliche Lösung: **Vorbedingung:** $x \geq 0$

Nachbedingung: $\sqrt{x} \cdot \sqrt{x} = x$

Multidimensionale Arrays

```
TYPE
  TBoard = ARRAY [1..3, 1..3] OF INTEGER;
...
VAR
  board          : TBoard;
```

board =

board[1,1]	board[1,2]	board[1,3]
board[2,1]	board[2,2]	board[2,3]
board[3,1]	board[3,2]	board[3,3]

Multidimensionale Arrays: Werte lesen

42	2	6
4	5	11
22	54	4

Speichere den Wert der Zelle auf Zeile 3 und Spalte 2 in die `INTEGER`-Variable `myval` ab:

```
myval := board[3,2];
{ myval hat nun den Wert 54 }
```

Multidimensionale Arrays: Werte setzen

42	2	6
4	5	11
22	54	4

Schreibe in die Zelle auf Zeile 1 und Spalte 3 den Wert 34 rein:

```
board[1,3] := 34;
```

42	2	34
4	5	11
22	54	4

Multidimensionale Arrays: Iteration

Verwende zwei verschachtelte FOR-Schleifen, um durch ein zweidimensionales Array zu iterieren:

```
FOR i:= 1 TO size DO
  FOR j := 1 TO size DO
    board[i,j] := ...
```

Tipps zu Aufgabe 6.1.a

- `HasWon` und `ValidMove` sind im Gerüst vorhanden, werden erst in Aufgabe 6.1.b implementiert, können jedoch in 6.1.a bereits aber verwendet werden
- Spielzug wird anhand einer Spalten- und Zeilennummer in der Konsole eingegeben. Falls Spielzug nicht gültig ist, muss eine erneute Eingabe des gleichen Spielers erfolgen.
  - Frage: Welcher Schleifen-Typ können wir verwenden, um diese Funktionalität zu implementieren?
    `WHILE-DO`, `FOR`, `REPEAT-UNTIL`?

Tipps zu Aufgabe 6.1.b

`ValidMove(board, i, j)`: Gibt `TRUE` zurück, wenn Koordinate (i, j) auf dem Feld liegt und Feld frei ist

Quiz-Frage 2

```
board : ARRAY [1..3, 1..3] OF INTEGER;
...
board =
```

empty	empty	0
X	0	empty
empty	0	X

Frage: Was wird zurückgegeben in den folgenden Zeilen?

1. `ValidMove(board,1,2)`
2. `ValidMove(board,3,2)`
3. `ValidMove(board,3,4)`
4. `ValidMove(board,0,1)`

Quiz-Frage 2

```
board : ARRAY [1..3, 1..3] OF INTEGER;
...
board =
```

empty	empty	O
X	O	empty
empty	O	X

Antwort:

1. ValidMove(board,1,2): TRUE
2. ValidMove(board,3,2): FALSE
3. ValidMove(board,3,4): FALSE
4. ValidMove(board,0,1): FALSE

Tipps zu Aufgabe 6.1.c

Tipp auf dem Übungsblatt zu `HasWon`: Zähle
für jede Reihe, Spalte und Diagonale, wieviele Steine der Spieler hat.

Frage: Wieviel `FOR`-Schleifen benötige ich für diese Teilaufgabe?

Tipps zu Aufgabe 6.1.c

Tipp auf dem Übungsblatt zu `HasWon`: Zähle
für jede Reihe, Spalte und Diagonale, wieviele Steine der Spieler hat.

Frage: Wieviel `FOR`-Schleifen benötige ich für diese Teilaufgabe?

Antwort: 4 Schleifen:

1. Jede Reihe
2. Jede Spalte
3. Hauptdiagonale
4. Antidiagonale