



Programmatie logica

Multi-dimensionale arrays als parameters

Webleren

School je gratis bij via het internet. Waar en wanneer je wilt.

www.vdab.be/webleren

© COPYRIGHT 2015 VDAB

Niets uit deze syllabus mag worden verveelvoudigd, bewerkt, opgeslagen in een database en/of openbaar gemaakt door middel van druk, fotokopie, microfilm, geluidsband, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van VDAB.

Hoewel deze syllabus met zeer veel zorg is samengesteld, aanvaardt VDAB geen enkele aansprakelijkheid voor schade ontstaan door eventuele fouten en/of onvolkomenheden in deze syllabus en of bijhorende bestanden.

Inhoud

Hoofdstuk 1.	Multi-dimensionale arrays als parameter.....	4
1.1.	Theorie.....	4
1.2.	Voorbeeld.....	4

Hoofdstuk 1. Multi-dimensionale arrays als parameters

In deze module bekijken we kort even wat je moet doen om multidimensionale arrays door te geven als parameters van een functie/procedure.

1.1. Theorie

Om een multidimensionale array door te geven als parameter moet je een aantal extra zaken doen.

Je moet een type definiëren:

```
type multiDim = array[1..50, 1..50]
```

Bij de declaratie van je array, gebruik je dit als gegevenstype:

```
var mijnArray: multiDim
```

Ook in de declaratie van je procedure/functie gebruik je hetzelfde type:

```
procedure VerwerkingTabellen(pArray: multiDim)
```

Bij het oproepen van je procedure hoeft je niets speciaals meer te doen.

```
VerwerkingTabel(mijnArray)
```

1.2. Voorbeeld

We illustreren dit aan de hand van een voorbeeld.

Je hebt in je hoofdprogramma twee multidimensionale arrays. Je wil deze via subroutines opvullen met willekeurige getallen, volledig tonen en een bepaalde kolom tonen.

Omdat je dezelfde subroutines wil gebruiken voor verschillende arrays, moet je de array als parameter meegeven.

Hoofdprogramma

Definieer een nieuwe type **tweeDim** voor een multidimensionale array van 10 rijen en 10 kolommen. Dit type gebruik je als gegevenstype van de twee arrays **eersteTabel** en **tweedeTabel**

Bij het oproepen van de subroutines hoeft niets speciaal te gebeuren.

Hoofdprogramma

```
TYPE tweeDim = array[0..9, 0..9] of integer
var eersteTabel, tweedeTabel: tweeDim
```

```
InitTabel(eersteTabel)
```

```
ToonTabel("Eerste tabel: ", eersteTabel)
```

```
ToonKolom("Derde kolom van eerste tabel: ", 2, eersteTabel)
```

```
InitTabel(tweedeTabel)
```

```
ToonTabel("Tweede tabel: ", tweedeTabel)
```

```
ToonKolom("Vijfde Kolom van tweede tabel: ", 4, tweedeTabel)
```

```
Write
```

```
Write "Druk op Enter om het programma te verlaten."
```

```
Read
```

De subroutines

Ook de parameters geef je als gegevenstype het nieuwe type **tweeDim**.

InitTabel(var pTabel: tweeDim)

```
var rij, kolom: integer
```

```
RANDOMIZE
```

```
For rij := 0 To 9
```

```
  For kolom := 0 TO 9
```

```
    pTabel[rij, kolom] := RANDOM(1000)
```

ToonTabel(pTitelVoor: string; pTabel: tweeDim)

```

var rij, kolom: integer
Write pTitelVoor
For rij := 0 To 9
  For kolom := 0 To 9
    Write pTabel[rij, kolom]:5
  Write
Write

```

ToonKolom(pTitelVoor: string; pKolom: integer; pTabel: tweeDim)

```

var rij: integer
Write pTitelVoor
For rij := 0 To 9
  Write pTabel[rij, pKolom]
Write

```

1.3. Opgave

Probeer het nu nog even zelf.

Schrijf een subroutine die het gemiddelde berekent van een te bepalen kolom in een te bepalen array.