

Opsamling og Datastrukturer

Imperativ Programmering og Datastrukturer

René Rydhof Hansen

November 2007

Mål

- At kunne definere hvad en datastruktur er
- At kunne definere og bruge en record (sammensat type)
- (At kunne argumentere for valg af datastruktur)

- At kunne forklare hvorfor programmering, og især imperativ programmering, er interessant.
- At kunne forklare, i oversigtsform, forskellene på et højniveausprog og et lavniveausprog samt fordele og ulemper ved samme.
- At kunne indplacere VB som et højniveausprog og argumentere for det.
- At kunne skrive (og få udført) "Hello World!" .
- At kunne skrive (og få udført) simple programmer med simpel input/ouput

Mål 01

- At kunne give et overblik over VB's kontrolstrukturer: sekventiel, selektiv og gentaget udførelse.
- At kunne forklare forskellen(e) på VB's kontrolstrukturer.
- At kunne beskrive de mest almindelige varianter og bruge dem i simple programmer
- At kunne benytte VB's visuelle programmeringsomgivelser
- At kunne forklare hvad hændelses-baseret programmering er

- At kunne forklare hvad en datatype er, hvad den kan bruges til og hvorfor datatyper er vigtige
 - At kunne give en oversigt over de mest basale datatyper i Visual BASIC
 - At kunne forklare forskellen på de basale datatyper
 - At kunne anvende datatyper korrekt i simple programmer
-
- At kunne starte et nyt projekt i Visual BASIC programmeringsomgivelsen
 - At kunne forklare hvad en "form" er
 - At kunne forklare hvad en "kontrol" er og hvordan den bruges
 - At kunne lave simple programmer v.hj.a. form's og kontroller

Mål 03

- At kunne forklare hvad en procedure/funktion er, samt forskellen på dem.
- At kunne bruge procedurer/funktioner til at strukturere simple programmer.

Mål 05

- At kunne forklare/definere hvad rekursion er
- At kunne implementere givne rekursive funktioner
- At kunne forklare hvordan rekursion kan bruges som problemløsningsstrategi

Definition

Data sammensat af forskellige datatyper, relationer mellem dem og operationer på dem.

- Organisering af data
- Effektivitet

- Arrays
- Records
- Linked lists (hægtede lister)
- Stakke
- Køer
- (Træer)
- ((Grafer))

Hvorfor Records (sammensatte typer)?

Arrays

```
Dim titel(117)      As String  
Dim udg_aar(117)   As Integer  
Dim spilletid(117) As Integer
```

- Spredt information
- Vanskeligt at se sammenhænge
- Ændringer skal ske mange steder

Records (VB6.0)

```
Type Album  
    titel      As String  
    udg_aar   As Integer  
    spilletid As Integer  
End Type
```

- Grupperer/samler et antal variable, muligvis af forskellig type, til et samlet logisk hele.
- De enkelte variable kaldes *felter* (fields)
- Hvert felt har et navn
- Felter tilgås med det *kvalificerede navn* (dot notation):
`record.fieldname`

Records (VB.NET)

```
Structure Album
    dim titel      As String
    dim udg_aar   As Integer
    dim spilletid As Integer
End Structure
```

- Grupperer/samler et antal variable, muligvis af forskellig type, til et samlet logisk hele.
- De enkelte variable kaldes *felter* (fields)
- Hvert felt har et navn
- Felter tilgås med det *kvalificerede navn* (dot notation):
`record.fieldname`

Example

```
Dim alb As Album
```

```
alb.titel = "FooBarBaz"
```

```
alb.udg_aar = 1963
```

```
alb.spilletid = 87
```

```
MsgBox("Titel:" & alb.titel)
```

Example

```
Dim albs(100) As Album
```

```
albs(42).titel = "FooBarBaz"
```

```
albs(42).udg_aar = 1963
```

```
albs(42).spilletid = 87
```

Example (Samlet Spilletid)

```
sum = 0
```

```
For i = 1 To 100
```

```
    sum = sum + albs(i).spilletid
```

```
Next
```