

Datatyper og Programmering i Visual BASIC

Imperativ Programmering og Datastrukturer

René Rydhof Hansen

October 2007

Ingen forelæsning eller øvelser i BAIT/PROB

torsdag den 25. oktober 2007

- 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

- En *type* er en mængde af værdier
- Hvorfor anvende typer?
 - Typer forbedrer programmets læsbarhed og forståelighed
 - Typer gør det muligt for computeren at generere bedre og mere effektiv kode
 - Typer gør det muligt at opdage fejl under oversættelsen
- Hvordan bruges typer?
 - I variabel-erklæringer: (Dim <var> as <type>)

Datatyper i Visual BASIC

- String
 - Ex: "foo", "42", "41+1"
- Integer
 - Ex: 42, 17, 117
- Single, Double
 - Ex: 41.9999, 3.141E3
- Boolean
 - Ex: True, False

Typekonvertering

- Konvertering fra streng til tal: `Val`
 - `Val("42") = 42`
 - `Val("8.7") = 8.7`
- Konvertering fra tal til streng: `Str`
 - `Str(42) = "42"`
 - `Str(11.7) = "11.7"`
- Implicit konvertering
 - `"41" + 1 = 42`
 - `"100" + "17" =`

Typekonvertering

- Konvertering fra streng til tal: Val
 - Val("42") = 42
 - Val("8.7") = 8.7
- Konvertering fra tal til streng: Str
 - Str(42) = "42"
 - Str(11.7) = "11.7"
- Implicit konvertering
 - "41" + 1 = 42
 - "100" + "17" = "10017"

Scope (virkefelt)

- Det “område” i programmet hvor en given variabel “gælder”

Example

```
Dim x as Integer = 41
```

```
Private Sub Main()
```

```
    Dim x as Integer = 86
```

```
    x = x + 1
```

```
    print x
```

```
End Sub
```

```
Private Sub Foo()
```

```
    x = x + 1
```

```
    print x
```

```
End Sub
```


Operationer på strenge

- Sammenligning: `"foo" = "bar"`
- Konkatenering: `"foo" & "bar" = "foobar"`
- Indeholder: `InStr(foobarbaz,bar) = 4`
- Erstat: `Replace("foobarbaz","ba","!!") = foo!!r!!z`
- Del-streng: `Mid$("foobarbaz",4,3) = bar`
- Tildel del-streng:

```
s = "foobarbaz"  
Mid$(s,4,3) = "!!!"  
s = "foo!!!baz"
```
- Længde: `Len("foobar") = 6`

Operationer på tal

- Addition: $111 + 6 = 117$
- Subtraktion: $89 - 2 = 87$
- Multiplikation: $2 * 21 = 42$
- Division: $10/3 = 3.333333$
- Heltals-division: $10 \setminus 3 = 3$
- Modulus: $10 \bmod 3 = 1$
- Potensopløftning: $2^{10} = 1024$

Arrays

- Indekserede variable
- Matematik: x_i for $i = 1..100$
- Erklæring af array

```
Dim titel(10) as String
```

- Brug af array

```
titel(0) = "???"    ' tilladt?
```

```
titel(1) = "foo"
```

```
titel(2) = "bar"
```

```
titel(10) = "baz"
```

```
titel(11) = "!!!"    ' IKKE tilladt
```

Example

```
Dim titel(10) as String  
for i = 1 to 10  
    titel(i) = InputBox("Indtast _titel_nr._" & i)  
next
```

Arrays i flere dimensioner

Example

```
Dim matrix(5,10) as String  
for i = 1 to 5  
    for j = 1 to 10  
        matrix(i,j) = Str(i) & ", " & Str(j)  
    next  
next
```

1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10
2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10
3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9	3,10
4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	4,10
5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9	5,10

Arrays i flere dimensioner

Example

```
Dim matrix(5,10) as String  
for i = 1 to 5  
    for j = 1 to 10  
        matrix(i,j) = Str(i) & ", " & Str(j)  
    next  
next
```

```
matrix(3,5) = "XX"
```

1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10
2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10
3,1	3,2	3,3	3,4	XX	3,6	3,7	3,8	3,9	3,10
4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	4,10
5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9	5,10