

# Funktioner og Procedurer; Datastrukturer (opsamling)

## Imperativ Programmering og Datastrukturer

René Rydhof Hansen

December 2007

# Mål

- At kunne forklare og bruge procedurer og funktioner (repetition)
- At kunne forklare hvad “parameteroverførsel” er
- At kunne forklare forskellen på “call-by-value” (CBV) og “call-by-reference” (CBR)
- At kunne bruge CBV og CBR i et Visual BASIC program

## Definition (Udtryk)

Et udtryk er

- Hvad der står på højresiden af en tildeling:

`var = udtr`

- Hvad der kan stå i parenteser i et procedurekald:

`proc(udtr, ..., udtr)`

- Hvad der kan stå i parenteser i et funktionskald:

`var = funk(udtr, ..., udtr)`

# Funktioner (eksempel): Euklidisk afstand

## Example

```
afstand = sqrt( (42 - 17)^2 + (87 - 117)^2 ) +  
          sqrt( (17 - 91)^2 + (117 - 47)^2 )
```

# Funktioner (eksempel): Euklidisk afstand

## Example

```
afstand = sqrt( (42 - 17)^2 + (87 - 117)^2 ) +  
          sqrt( (17 - 91)^2 + (117 - 47)^2 )
```

# Funktioner (eksempel): Euklidisk afstand

## Example

```
Function dist(x1, x2, y1, y2 as Integer) as Single
    dist = sqrt( (x1 - x2)^2 + (y1 - y2)^2 )
End Function
```

```
afstand = dist(42,87,17,117) +
            dist(17,117,91,47)
```

## Example (Indsættelse af albums)

```
Dim i as Integer
```

```
' ryk fra 42 og nedefter
```

```
For i = 42 To antal_albums
```

```
    albums(i+1) = albums
```

```
Next
```

```
albums(42) = ny_alb1    'indsæt nyt album på plads 42
```

```
' ryk fra 117 og nedefter
```

```
For i = 117 To antal_albums
```

```
    albums(i+1) = albums
```

```
Next
```

```
albums(117) = ny_alb2    'indsæt nyt album på plads 117
```

## Example (Indsættelse af albums)

```
' Indsæt et album i et givet index og ryk de efterfølgende
Sub indsaet_album(alb as Album, index as Integer)
  Dim i as Integer

  For i = index To antal_albums
    albums(i+1) = albums(i)
  Next
  albums(index) = alb
End Sub

...
indsaet_album(ny_alb1,42)
indsaet_album(ny_alb2,117)
```



# Parameteroverførsel

- Overfører *aktuel* parameter til *formel* parameter
- Call-by-Value: lav en kopi af den aktuelle værdi og giv den til funktionen
- Call-by-Reference: giv en *reference* til den aktuelle værdi til funktionen

## Example (Call-by-Value)

```
Function dist(x1, x2, y1, y2 as Integer) as Single
    dist = sqrt( (x1 - x2)^2 + (y1 - y2)^2 )
End Function
```

```
afstand = dist(42,87,17,117) +
           dist(17,117,91,47)
```

# Parameteroverførsel

- Overfører *aktuel* parameter til *formel* parameter
- Call-by-Value: lav en kopi af den aktuelle værdi og giv den til funktionen
- Call-by-Reference: giv en *reference* til den aktuelle værdi til funktionen

## Example (Call-by-Value)

```
Function dist(x1, x2, y1, y2 as Integer) as Single
    dist = sqrt( (x1 - x2)^2 + (y1 - y2)^2 )
End Function
```

```
afstand = sqrt( (42 - 17)^2 + (87 - 117)^2 ) +
            sqrt( (17 - 91)^2 + (117 - 47)^2 )
```

# Parameteroverførsel

- Overfører *aktuel* parameter til *formel* parameter
- Call-by-Value: lav en kopi af den aktuelle værdi og giv den til funktionen
- Call-by-Reference: giv en *reference* til den aktuelle værdi til funktionen

## Example (Call-by-Value)

```
Function dist(ByVal x1, x2, y1, y2 as Integer) as Single
    dist = sqrt( (x1 - x2)^2 + (y1 - y2)^2 )
End Function
```

```
afstand = sqrt( (42 - 17)^2 + (87 - 117)^2 ) +
           sqrt( (17 - 91)^2 + (117 - 47)^2 )
```

# Eksempel: Indsættelse af element i hængt liste

## Example (Call-by-Reference)

```
Sub AddElement(ByRef tmp as Element)
```

```
    Dim curr as Element
```

```
    If (head Is Nothing) OrElse (tmp.info < head.info) Then
```

```
        tmp.next = head
```

```
        head = tmp
```

```
    Else
```

```
        curr = head
```

```
        Do While (Not curr.next Is Nothing) And
```

```
            (curr.next.info < tmp.info)
```

```
            curr = curr.next
```

```
        Loop
```

```
        tmp.next = curr.next
```

```
        curr.next = tmp
```

```
    End If
```

```
End Sub
```