

The Select Case structure

- The SELECT CASE structure is another branching structure provided by VBA. It is a more elegant and transparent version of an IF-structure, which tests always the same variable.
Example: write a UDF that determines the sign of a number

Function sig(x As Single) As String

If x > 0 Then

sig = "positive"

ElseIf x < 0 Then

sig = "negative"

Else

sig = "zero"

End If

End Function

Select Case x

Case Is > 0: sig = "positive"

= Case Is < 0: sig = "negative"

Case Else: sig = "zero"

End Select

Syntax: **Select Case** testvariable

[**Case** expressionlist
[statements]]... } can be repeated many times

[**Case Else**
[elsestatements]]

End Select

- testvariable ≡ a numeric or string expression
- expressionlist ≡ list of one or more expressions separated by a comma
 - expression
 - expression **To** expression
 - **Is** comparisonoperator expression
- statements ≡ executed when one condition from expressionlist is true
- elsestatements ≡ executed when no previous condition is true

- Examples (Select case):

a) Function si(x)

Select Case x

Case 0: si = 1

Case Else: si = Sin(x) / x

End Select

End Function

$$Si(x) = \begin{cases} \frac{\sin x}{x} & \text{for } x \in \mathbb{R} \setminus 0 \\ 1 & \text{for } x = 0 \end{cases}$$

b) Function F(x As Single) As Single

Select Case x

Case Is < 0: F = 0

Case 0 To 4: F = 3 * x

Case Else: F = 12

End Select

End Function

$$F(x) = \begin{cases} 0 & \text{for } x < 0 \\ 3x & \text{for } 0 \leq x \leq 4 \\ 12 & \text{for } x > 4 \end{cases}$$

• Note that "a To b" means " $a \leq x \leq b$ "

c) Function G(x As Single) As Single

Select Case x

Case -4 To 4: G = 1

Case Else: G = 0

End Select

End Function

$$G(x) = \begin{cases} 1 & \text{for } -4 \leq x \leq 4 \\ 0 & \text{otherwise} \end{cases}$$

· Note that "a To b" means " $a \leq x \leq b$ "

d) Function entry(age As Integer) As Variant

Select Case age

Case 0 To 5, Is > 65: entry = 0

Case 6 To 15: entry = 2

Case 15 To 65: entry = 5

Case Else: entry = "Age not valid!"

End Select

End Function

e) Function price(product As String) As Variant

Select Case product

Case "Mangoes": price = 2.5

Case "Bananas": price = 1.8

Case "Pears", "Apples": price = 0.9

Case Else: price = "Fruit not in price list!"

End Select

End Function

- Note that the test variable can also be of string type
- Note that price is of type Variant, as it could be a number or a string
- Note that the test is case sensitive, e.g.
=price("mangoes") → "Fruit not in price list!"
- Note that when the "Case Else" line is dropped
=price("Papayas") → 0

f) Function pricec(product As String, country As String) As Variant

Select Case country

Case "Brasil"

Select Case product

Case "Mangoes" , "Papayas": pricec = 2.5

Case "Bananas": pricec = 1.3

Case Else: pricec = "Fruit not in the list!"

End Select

Case "Thailand"

Select Case product

Case "Mangoes": pricec = 2.2

Case "Papayas": pricec = 2.8

Case Else: pricec = "Fruit not in the list!"

End Select

Case Else: pricec = "Country not the list!"

End Select

End Function

- One can also nest the SELECT structure similar to the IF-structure

Example from exam 2007

Use a SELECT CASE structure to produce a UDF called `anniversary`. The function should take a wedding date as input and produce the following output:

If the wedding falls in the first 6 months of the year the function should return the message: “the wedding was between January and June”

Otherwise it should return: “the wedding was between July and December”

Define the variable type of the input and output.

Function anniversary(x as date) as string

y=month(x)

Select Case y:

Case 1 To 6: anniversary="the wedding
falls between January and June"

Case Else: anniversary="the wedding falls between
July and December"

End Select

End Function

- A feature of the Select Case structure is that it can only handle well three kinds of cases:
the variable is smaller than a value (**Case Is <** value),
the variable equals a value (**Case** value),
the variable varies between two values (those values included) (**Case** value 1 **To** value 2),
- How do we handle more general cases?
Example: the function

$$g(x) = \begin{cases} 0 & \text{for } x \leq -1 \\ 1/2 & \text{for } -1 < x \leq 9 \\ 0 & \text{for } 9 < x \end{cases}$$

$$g(x) = \begin{cases} 0 & \text{for } x \leq -1 \\ 1/2 & \text{for } -1 < x \leq 9 \\ 0 & \text{for } 9 < x \end{cases}$$

Function g(x as Single) as Single

Select Case x:

Case Is <= -1: g=0

Case Is <= 9: g=1/2

Case Else: g=0

End Select

End Function

Alternatively:

Function g(x as Single) as Single

Select Case x:

Case Is <= -1, Is > 9: g=0

Case Else: g=1/2

End Select

End Function

Announcements:

- The Programming Part I test will take place on January the 5th between 10:00 and 11:30. The exams timetable is available at:

<http://www.city.ac.uk/exams/exam-timetables.html>

- More information about the test will be given in the next lecture.
- Next week we will have our last lecture for Part I of the module. The Lab after that will also be the last Lab of the term.
- I will be at University and available for questions up until the 17th of December. After that I will be able to reply to queries by e-mail up to the 29th of December. Between the 30th and the 8th of January I will not be accessible.