Microsoft Visual Basic 2012: Reloaded

Fifth Edition

Chapter Five
More on the Selection Structure
Objectives

After studying this chapter, you should be able to:

• Determine whether a solution requires a nested selection structure
• Include a nested selection structure in pseudocode and in a flowchart
• Code a nested selection structure
• Determine whether a solution requires a multiple-alternative selection structure
• Include a multiple-alternative selection structure in pseudocode and in a flowchart
Objectives (cont'd)

• Code a multiple-alternative selection structure
• Include radio buttons in an interface
• Display a message in a message box
• Use a message box’s return value
• Prevent the entry of invalid characters in a text box
Making More than One Decision

• Up until now, our selection structures have contained only 2 alternatives:
  • A statement block for the True Path
  • A statement block for the False Path
What happens if you want to choose from more than 2 alternatives?

Use a Nested Selection Structure

When either a selection structure’s True path or its False path contains another selection structure, the inner selection structure is referred to as a nested selection structure because it is contained (nested) within the outer selection structure.
Making More than One Decision

Problem specification
Derek is practicing for an upcoming basketball game. Write the instructions that direct him to shoot the basketball and then say either one or two of four phrases, depending on whether or not the basketball went through the hoop and also where he was standing when he made the basket.

Result of shot
Basketball went through the hoop
Derek made the basket from either inside or on the 3-point line
Derek made the basket from behind the 3-point line
Basketball did not go through the hoop

Phrase
I did it!
2 points for me
3 points for me
Missed it!

1. shoot the basketball
2. if the basketball went through the hoop
   say “I did it!”
   if Derek was either inside or on the 3-point line
   say “2 points for me”
   else
   say “3 points for me”
   end if
   else
   say “Missed it!”
   end if

outer dual-alternative selection structure
Making More than One Decision

Problem specification
Derek is practicing for an upcoming basketball game. Write the instructions that direct him to shoot the basketball and then say either one or two of five phrases, depending on whether or not the basketball went through the hoop and also where he was standing when he made the basket.

<table>
<thead>
<tr>
<th>Result of shot</th>
<th>Phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basketball went through the hoop</td>
<td>I did it!</td>
</tr>
<tr>
<td>Derek made the basket from either inside or on the 3-point line</td>
<td>2 points for me</td>
</tr>
<tr>
<td>Derek made the basket from behind the 3-point line</td>
<td>3 points for me</td>
</tr>
<tr>
<td>Basketball did not go through the hoop</td>
<td>Missed it!</td>
</tr>
<tr>
<td>His missed shot hit the rim</td>
<td>So close</td>
</tr>
</tbody>
</table>

1. shoot the basketball
2. if the basketball went through the hoop
   say “I did it!”
   If Derek was either inside or on the 3-point line
     say “2 points for me”
   else
     say “3 points for me”
   end if
   else
   say “Missed it!”
   if the basketball hit the rim
     say “So close”
   end if
The Omaha Steaks Application

- The application calculates the total amount a customer owes, including shipping
- The shipping charge depends on two items:
  - the subtotal
  - the club membership
- If the subtotal is greater than $79
  - the order ships for free
- Else if the subtotal is not greater than $79
  - The shipping charge depends on whether the customer is a member of the company’s Shipping Club
  - If the customer is a member, the shipping charge is $10; otherwise, it’s $20
The Omaha Steaks Application

```
start

store boxes in a variable

subtotal = boxes * price per box

subtotal > 79

F

F

club member

shipping = 20

T

shipping = 10

F

T

shipping = 0

total = subtotal + shipping

display total

stop
```
The Omaha Steaks Application

Work problem together –
Nested Selection Structures
Multiple-Alternative Selection Structures

- Use the **Multiple-Alternative Selection Structure** if you want to *choose from several alternatives* at once.
- Is also referred to as an **Extended Selection Structure**
- It’s equivalent to nesting each condition in the **False Path** of the previous condition
- In the next example, the program will calculate the appropriate ticket price based on the number entered
Multiple-Alternative Selection Structures

```
If code = 1 OrElse code = 2 Then
    priceLabel.Text = "$15"
Else
    If code = 3 Then
        priceLabel.Text = "$25"
    Else
        If code = 4 Then
            priceLabel.Text = "$35"
        Else
            priceLabel.Text = "Invalid code"
        End If
    End If
End If
End If
```
Multiple-Alternative Selection Structures

- A simpler, and more compact way, of coding the multiple alternative selection structure is to use the keyword *ElseIf*.

```csharp
If code = 1 OrElse code = 2 Then
    priceLabel.Text = "$15"
ElseIf code = 3 Then
    priceLabel.Text = "$25"
ElseIf code = 4 Then
    priceLabel.Text = "$35"
Else
    priceLabel.Text = "Invalid code"
End If
```
Multiple-Alternative Selection Structures (cont'd.)

Flowchart:
- Start
- Store the code in a variable
- Code
  - 1, 2: Display "$15"
  - 3: Display "$25"
  - 4: Display "$35"
  - Other: Display "Invalid code"
- Stop
Multiple-Alternative Selection Structures (cont’d)

Write the code for Yardley Theater using the *ElseIf* statement
The Select Case Statement

- The **Select Case statement** is used when a multiple-alternative selection structure has many paths from which to choose.

Example

```vbnet
Select Case code
  Case 1, 2
    priceLabel.Text = "$15"
  Case 3
    priceLabel.Text = "$25"
  Case 4
    priceLabel.Text = "$35"
  Case Else
    priceLabel.Text = "Invalid code"
End Select
```

- The `selectorExpression` needs to match only one of these values.
The Select Case Statement

See handout comparing an If...Then...Else Statement to a Select Case Statement
The Select Case Statement

Work Together: English/Spanish translator for the days of the week
The Select Case Statement

- Work Together: Tutorial #1, pgs 259-266
Rock, Paper, Scissors Game
How to Specify a Range of Values in a Case Clause

Example
The ABC Corporation's price chart is shown here:

<table>
<thead>
<tr>
<th>Quantity ordered</th>
<th>Price per item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 5</td>
<td>$25</td>
</tr>
<tr>
<td>6 – 10</td>
<td>$23</td>
</tr>
<tr>
<td>More than 10</td>
<td>$20</td>
</tr>
<tr>
<td>Less than 1</td>
<td>$0</td>
</tr>
</tbody>
</table>

Select Case quantity
Case 1 To 5
  price = 25
Case 6 To 10
  price = 23
Case Is > 10
  price = 20
Case Else
  price = 0
End Select
How to Specify a Range of Values in a Case Clause

• Work Together: Grades Program
• Refer to handout for ranges
Using Radio Buttons in an Interface

- **Radio buttons** are commonly used when *multiple selection structures* are involved.
- Radio buttons allow you to limit the user to only one choice from a group of two or more related choices that are **mutually exclusive**.
- Each radio button’s label should be formatted using **sentence capitalization**.
- **Access keys** should also be assigned in the radio button’s label.
- If there is more than one set of radio buttons, place each set in a **group box** so that one item can be selected from each group.
Using Radio Buttons in an Interface (cont’d)

• Add a group box to a form using the GroupBox tool
• Place each group of radio buttons in a separate group box to allow the user to select one button from each group
• The recommended maximum number of radio buttons in a group is seven
• Windows applications normally have one of the radio buttons in each group already selected as a default radio button
  • Set that button’s Checked property to true
Using Radio Buttons in an Interface
If ...Then...Else

' determine occasion
If bdayRadioButton.Checked Then
  price = 35
ElseIf gradRadioButton.Checked Then
  price = 32.5
Else
  price = 30
End If

' if necessary, add $10 for a large bouquet
If largeRadioButton.Checked Then
  price += 10
End If
' determine occasion
Select Case True
  Case bdayRadioButton_Checked
    price = 35
  Case gradRadioButton_Checked
    price = 32.5
  Case Else
    price = 30
End Select
The MessageBox.Show Method

- Used when the program needs to communicate with the user during run time
# The MessageBox.Show Method

**Syntax**

```csharp
MessageBox.Show(text, caption, buttons, icon[, defaultButton])
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>text</code></td>
<td>text to display in the message box; use sentence capitalization</td>
</tr>
<tr>
<td><code>caption</code></td>
<td>text to display in the message box's title bar; use book title capitalization</td>
</tr>
<tr>
<td><code>buttons</code></td>
<td>buttons to display in the message box; can be one of the following constants:</td>
</tr>
<tr>
<td></td>
<td>MessageBoxButtons.AbortRetryIgnore</td>
</tr>
<tr>
<td></td>
<td>MessageBoxButtons.OK (default setting)</td>
</tr>
<tr>
<td></td>
<td>MessageBoxButtons.OKCancel</td>
</tr>
<tr>
<td></td>
<td>MessageBoxButtons.RetryCancel</td>
</tr>
<tr>
<td></td>
<td>MessageBoxButtons.YesNo</td>
</tr>
<tr>
<td></td>
<td>MessageBoxButtons.YesNoCancel</td>
</tr>
<tr>
<td><code>icon</code></td>
<td>icon to display in the message box; typically, one of the following constants:</td>
</tr>
<tr>
<td></td>
<td>MessageBoxIcon.Exclamation</td>
</tr>
<tr>
<td></td>
<td>MessageBoxIcon.Information</td>
</tr>
<tr>
<td></td>
<td>MessageBoxIcon.Stop</td>
</tr>
<tr>
<td><code>defaultButton</code></td>
<td>button automatically selected when the user presses Enter; can be one of the following constants:</td>
</tr>
<tr>
<td></td>
<td>MessageBoxIcon.DefaultButton.Button1 (default setting)</td>
</tr>
<tr>
<td></td>
<td>MessageBoxIcon.DefaultButton.Button2</td>
</tr>
<tr>
<td></td>
<td>MessageBoxIcon.DefaultButton.Button3</td>
</tr>
</tbody>
</table>
The MessageBox.Show Method

**Example 1**
```csharp
MessageBox.Show("Record deleted.", "Payroll", MessageBoxButtons.OK, MessageBoxIcon.Information)
displays an information message box that contains the message “Record deleted.”
```

**Example 2**
```csharp
MessageBox.Show("Delete this record?", "Payroll", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Button2)
displays a warning message box that contains the message “Delete this record?”
```
The MessageBox.Show Method Using the Return Value

**HOW TO** Use the MessageBox.Show Method’s Return Value

<table>
<thead>
<tr>
<th>Integer</th>
<th>DialogResult value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Windows.Forms.DialogResult.OK</td>
<td>user chose the OK button</td>
</tr>
<tr>
<td>2</td>
<td>Windows.Forms.DialogResult.Cancel</td>
<td>user chose the Cancel button</td>
</tr>
<tr>
<td>3</td>
<td>Windows.Forms.DialogResult.Abort</td>
<td>user chose the Abort button</td>
</tr>
<tr>
<td>4</td>
<td>Windows.Forms.DialogResult.Retry</td>
<td>user chose the Retry button</td>
</tr>
<tr>
<td>5</td>
<td>Windows.Forms.DialogResult.Ignore</td>
<td>user chose the Ignore button</td>
</tr>
<tr>
<td>6</td>
<td>Windows.Forms.DialogResult.Yes</td>
<td>user chose the Yes button</td>
</tr>
<tr>
<td>7</td>
<td>Windows.Forms.DialogResult.No</td>
<td>user chose the No button</td>
</tr>
</tbody>
</table>

**Example 1**

```vbnet
Dim dlgButton As DialogResult
dlgButton = MessageBox.Show("Delete this record?", "Payroll", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Button2)
If dlgButton = Windows.Forms.DialogResult.Yes Then
    ' instructions to delete the record
End If
```
Using Radio Buttons in an Interface
(cont’d.)

Work Together:
Tutorial #2, pgs. 266-271
Charleston Cable Company
Using the KeyPress Event

• A control’s **KeyPress event** occurs each time the user presses a key while the control has the focus
  • The character corresponding to the pressed key is sent to the event’s **e parameter**
  • To prevent a text box from accepting an inappropriate character, you first use the e parameter’s **KeyChar property** to determine the pressed key
  • You then use the e parameter’s **Handled property** to cancel the pressed key if it is an inappropriate one
Using the KeyPress Event

- You **cancel the key** by **setting the Handled property to True**, like this: `e.Handled = True`
- The KeyPress event procedure determines whether the value stored in the KeyChar property is inappropriate for the text box
- You refer to the **Backspace key** on your keyboard using Visual Basic’s `ControlChars.Back` constant
- The KeyPress event automatically allows the use of the Delete key for editing
Using the KeyPress Event (cont’d)

**HOW TO** Use the KeyPress Event to Control the Characters Accepted by a Text Box

**Version 1**

```csharp
Private Sub boxesTextBox_KeyPress(sender As Object, e As KeyPressEventArgs) Handles boxesTextBox.KeyPress
    ' accept only numbers and the Backspace key
    If (e.KeyChar < "0" OrElse e.KeyChar > "9") AndAlso e.KeyChar <> ControlChars.Back Then
        e.Handled = True
    End If
End Sub
```
Using the KeyPress Event (cont’d.)

- Work Together: CD Warehouse, pgs. 271-273
Summary

• You can nest a selection structure in either the True or False path of another selection structure
• The primary decision is always made by an outer selection structure. The secondary decision is always made by a nested selection structure
• You can code a multiple-alternative selection structure using either If...Then...Else statements or the Select Case statement
• In a flowchart, a diamond is used to represent the condition in a multiple-alternative selection structure
  • The diamond has several flow lines leading out of the symbol, with each flow line representing a possible path
In a Select Case statement, the data type of the expressions in the Case clauses should match the data type of the statement’s selectorExpression.

A Case clause in a Select Case statement can contain more than one expression.

- The selectorExpression needs to match only one of the expressions for the instructions in that Case to be processed.

- Use the keyword To in a Case clause’s expressionList when you know both the upper and lower values of the range you want to specify.
• Radio buttons allow you to limit the user to only one choice from a group of two or more related but mutually exclusive choices

• If you need to include two groups of radio buttons in an interface, at least one of the groups must be placed within a container, such as a group box, panel, or table layout panel

• It is customary to have one radio button in each group of radio buttons selected when the interface first appears

  • The selected radio button is the default radio button
Summary (cont’d)

• If a radio button is selected, its Checked property contains the Boolean value True; otherwise, it contains the Boolean value False

• The MessageBox.Show method allows an application to communicate with the user while the application is running

• The MessageBox.Show method displays a message box that contains text, one or more buttons, and an icon
Summary (cont'd)

• Use sentence capitalization for the text argument in the MessageBox.Show method, but book title capitalization for the caption argument

• You can code a text box’s KeyPress event procedure to prevent the text box from accepting an inappropriate character
  • The character is stored in the e parameter’s KeyChar property
  • To cancel the character, you set the e parameter’s Handled property to True