

Computer/Human Interaction

Lecture 10

Overview:

- More Visual Basic form elements
- More Visual Basic code elements
- Visual Basic File I/O

References:

- VBMS, VBCL, VBIO

Purple Pizza Parlor, v2

- Create a new VB project on local (C:) drive.
- Rename `Form1.vb` to `pizza2.vb`; save and close solution
- Download `pizza2.vb` and `pizza2.Designer.vb` from course webpage into project folder overwriting the original files
- Open project; right-click on `pizza2.vb` in solution tree, select View Code

Designer-Generated Code

- Application is a class. One instance is created when program is run
- Form elements are Friend data members of application class
- Generated code initializes data member properties and registers handlers for events

More Form Elements

- ComboBox: Drop-down menu of choices
 - `Items` (in Data section) holds choices; `Text` holds current choice
- ListBox: Box with all choices shown
 - also has `Items`; `SelectMode` can be One, Multi-Single, or Multi-Extend; selected elements in `<LBvar>.SelectedItems` array
- Default handler for both is `SelectedIndexChanged`

For Loop and Arrays

- For - To - Next counter-controlled loop
- Arrays are 0-based, indexed using `()`, have `Count` (number of elements) property. E.g.

```
count = _
    ToppingsListBox.SelectedItems.Count

For i = 0 to count-1
    ToppingsListBox.SelectedItems(i)
Next
```

Message Boxes

- Full version:

```
MsgBox (<message>, <type>, <title>)
```
- Type controls icon style and buttons. E.g.,
 - `vbExclamation`, `vbCritical`
 - `vbOKOnly`, `vbAbortRetryIgnore`
 - Added together to combine:
`vbQuestion+vbOkCancel`
- Returns value of button clicked. E.g.,
 - `vbOK`, `vbCancel`

Functions

- Returns a value
- Most will be Private

- Syntax:

```
Private Function <name>(<param list>) _  
    as <return type>  
    ` function code goes here  
End Function
```

- Parameter syntax:

```
ByVal <name> As <type>
```

Subroutines

- Do not return a value
- Also most will be Private

- Syntax:

```
Private Sub <name>(<param list>)  
    ` subroutine code goes here  
End Sub
```

File Output

- **System.IO.StreamWriter** type, declare stream variable as usual

- Open using

```
outStream = _  
    My.ComputerFileSystem.OpenTextFileWriter _  
    (<filename>, <append bool>)
```

- **Write** and **WriteLine** member functions overloaded for all built-in types

File Input

- **System.IO.StreamReader** type, declare stream variable as usual

- Open using

```
inStream = _  
    My.ComputerFileSystem.OpenTextFileReader _  
    (<filename>)
```

- **ReadLine** member function reads and returns a line of file as a string

More VB code notes

- Can import namespace names, so don't have to write them explicitly. E.g.

```
Import System.IO  
:  
Private Sub Submit ()  
    Dim outStream As StreamWriter
```

- **<type>.Parse (<string>)** functions return the value of **<type>** literal with **<string>** representation. E.g.

```
Dim i As Integer  
i = Integer.Parse ("1234") ` i = 1234
```

In-class Exercise

- Add sizes and/or toppings
- Separate the toppings into lists of meats and vegetables
- Add error checking to the Submit button handler for missing address data
- Add pricing information and display the computed total in the confirmation message box