



MT264 Solutions to the Specimen Examination Paper (2009)

Note: Any text marked **Comment** does not form part of the solution, but has been included for information. Please note that there are many other valid alternative solutions, but it was infeasible to include them all.

Part 1

Question 1

- (a) **Set** aTextBox.Text **To** "Hello world"
- (b) The background colour of the button aButton will turn red.
- (c) maxSize **As** Integer
Set maxSize **To** 100
Set aNumUpDn.Maximum **To** maxSize
- (d) The Interval property.

Comment: VB code instead of design code is acceptable. If you prefer to write VB code, then your solution should use the symbol = instead of **Set ... To**, and maxSize should be declared as a local variable with the code **Dim** maxSize **As** Integer.

Question 2

- (a) salesOver
- (b) The method has one parameter, of type Integer.
- (c) The method returns a value of type Boolean.
- (d) No, the method does not change the state of the object on which it is invoked.
- (e) (i) aBook **As** Book
Set aBook **To** New Book
Set aBook.Title **To** "A new book"
Set aBook.Author **To** "A.N.On"
(ii) False

Question 3

- (a) 11
- (b) False
- (c) True
- (d) "k0", "k1", "k2", "k3", "k4"

Question 4

- (a) **Method** howMany(word **As** String) **As** Integer
 result **As** Integer
 len **As** Integer
 Set result **To** 0
 Set len **To** word.Length
 For position **As** Integer **From** 0 **To** Text.Length – len
 If Text.substring(position, len) = word **Then**
 Set result **To** result + 1
 End If
 End For
 Return result
End Method

Comment: Note that if Text is the empty string, then Text.Length – len is less than 0 and the **For** loop does not get executed. In this case the return value is 0.

- (b) Any three of the following tests would be acceptable.
- ▶ word: "be" as this occurs several times in the text.
 - ▶ word: "." to check that the last character is included.
 - ▶ word: "N" to check that the first character is included.
 - ▶ word: "zz" to check that there is no problem if the word does not occur at all.

Comment: Note that it would be invalid to test the empty string, as this test would not satisfy the preconditions of howMany.

Question 5

- (a) The effect of declaring the method timeNeededToHarvest **as Protected** is that it can only be invoked within its class implementation or within an implementation of a subclass.
- (b) (i) Yes, since it is a public method of PeaField.
(ii) Yes, since it is a public method of PeaField.
- (c) **Constructor** New
 MyBase.New
 Set Soil **To** SoilType.Neutral
 End Constructor

Comment: Invoking MyBase.New initialises Area and Shape.

Question 6

- (a) **Set** openFileDialog.Filter **To** "Image files (*.jpg)|*.jpg"
If openFileDialog.ShowDialog() = DialogResult.OK **Then**
 Try
 Set pictureBox.Image **To** Image.FromFile(openFileDialog.FileName)
 Catch ex **As** Exception
 'Do something to indicate that there is a problem loading the file.'
 End Try
End If

Comment: Alternatively, you could explain how to set the Filter property in the property list at design time and use a message box instead of a comment in the **Catch** part.

- (b) centreX **As** Integer
centreY **As** Integer
Set centreX **To** canvasPanel.Width \ 2
Set centreY **To** canvasPanel.Height \ 2
e.Graphics.drawRectangle(Pens.Black, centreX - 50, centreY - 50, 100, 100)
e.Graphics.drawString("Hello", fCurrentFont, Brushes.Blue, centreX - 50, centreY - 50)

Comment: An alternative to using e.Graphics is to create canvasPanel's Graphics object. In that case you should dispose of it at the end of your code.

Question 7

- (a) SELECT RecipeCode, Name
FROM Recipe
WHERE Type = 'vegetarian';
- (b) This will return the recipe codes of all recipes containing an ingredient that has a non-zero sugar content.

Comment: Although this sentence does not specifically say that the quantity of the ingredient is non-zero, this can be inferred.

Question 8

- (a) If an application has been designed using an MVC pattern, then it should be possible to alter the GUI without altering the model. Only those aspects of the GUI that involve user interaction will need altering.
- (b) The TabIndex property should be set so that controls gain focus on tabbing in the order in which they are most likely to be used, i.e. TabIndex should be set to 0 for the input text box and to 1 for the button. The output text box should have TabStop set to False, so that it will not gain focus. The input text box should have focus initially. The button should have a keyboard shortcut or mnemonic, or should be set up as the form's AcceptButton.

Part 2

Question 9

- (a) **Method** `isCurrentCode(code As String) As Boolean`
Return `fStockItems.ContainsKey(code)`
'Alternatively: Return StockList.contains(code)
End Method
- Method** `numberInStock(code As String) As Integer`
`aStock As Stock`
Set `aStock To fStockItems.Item(code)`
Return `aStock.Quantity`
'Alternatively, just: Return fStockItems.Item(code).Quantity
End Method
- Method** `addStock(code As String, aNumber As Integer)`
`aStock As Stock`
Set `aStock To fStockItem(code)`
Set `aStock.Quantity To aStock.Quantity + aNumber`
'Alternatively:
'Set fStockItem(code).Quantity To fStockItem(code).Quantity + aNumber
End Method
- Method** `newStockItem(code As String, aName As String)`
`aStock As Stock`
Set `aStock To New Stock`
Set `aStock.Name To aName`
`fStockItems.add(code, aStock)`
End Method
- (b) (i) It should add a new stock item to `fStockAdmin`, using the code and name entered by the user into the two text boxes.
- (ii)
- ```
code As String
Set code To codeTextBox.Text
If Not (code = "") And Not (fStockAdmin.isCurrentCode(code)) Then
 fStockAdmin.newStockItem(code, nameTextBox.Text)
Else
 MessageBox.Show("Your choice of code is invalid.")
End If
codeTextBox.Text = ""
nameTextBox.Text = ""
```
- Comment:** Some feedback that the new item has been added should be given. In the code above this is done by clearing the two text boxes.
- (c) (i) ► A `ComboBox` or `ListBox` control called `codeBox` should be filled with the valid codes given by `fStockAdmin.StockList`.
- A `NumericUpDown` control called `quantNumUpDn` should have its `Minimum` property set to 0 and its `Maximum` property set to the current quantity held for the stock item whose code is selected in `codeBox`.
- There should also be a button called `removeButton`.

```
(ii) removeButton clicked
 OnClick
 If codeBox.SelectedIndex >= 0 Then
 fStockAdmin.removeStock(codeBox.SelectedItem, quantNumUpDn.Value)
 End If
 Set quantNumUpDn.Value To 0
 Set codeBox.SelectedIndex To -1 'or some other feedback
```

**Comment:** Because the question did not ask just for the body of the event handler, the event handler header (in design language) was included in the above solution.

## Question 10

```
(a) Private Method updateImage()
 g As Graphics
 Set g To Graphics.FromImage(GameArea)
 g.clear(Color.White)
 fJack.draw(g)
 g.fillRect(Brushes.Black, fBox.Left, fBox.Top, fBox.Width, fBox.Height)
 g.dispose()
 End Method

 Method startJump()
 Try
 Audio.play(fJack.NoiseFile, AudioPlayMode.Background)
 Catch
 'Do nothing.
 End Try
 Set fMoving To True
 fJack.goUp(5)
 Set fJack.GoingUp To True
 updateImage()
 End Method

 Method move()
 If Moving Then
 fJack.move()
 If (fJack.GoingUp) And Not (fJack.BoundingBox.intersectsWith(fBox)) Then
 Set fMoving To False
 End If
 If Not (fJack.GoingUp) And (fJack.Position.Y > GameArea.Height - fBox.Height) Then
 Set fMoving To False
 End If
 updateImage()
 End If
 End Method
```

**Comment:** In updateImage, first the jack and then the box need to be drawn. As the sound file is part of the application, **Try–Catch** is not really necessary in startJump, as it could be argued that the developer should ensure that a valid sound file exists.

```
(b) (i) Private Method updateView()
 'The display of the jack–in–the–box is updated.
 Set gamePictureBox.Image To fJackAdmin.GameArea
 End Method
```

**Comment:** Note that the method header and comment need to be given as part of this solution.

```
(ii) fJackAdmin.startJump()
 Set gameTimer.Enabled To True
 updateView()
```

**Comment:** updateView can be omitted from this code provided that it is included in the code given in part (iii).

```
(iii) fJackAdmin.move()
 updateView()
 If Not fJackAdmin.Moving Then
 Set gameTimer.Enabled To False
 End If
```

## Question 11

### (a) Postcode application – components

|                       | Name                             | Property   | Initial value         |
|-----------------------|----------------------------------|------------|-----------------------|
| <b>Components</b>     |                                  |            |                       |
| DataSet               | postcodeDataDataSet              |            |                       |
| BindingSource         | postcodeBindingSource            | DataSource | postcodeDataDataSet   |
|                       |                                  | DataMember | Postcode              |
| PostcodeTableAdapter  | postcodeTableAdapter             |            |                       |
| BindingSource         | fkResidencePostcodeBindingSource | DataSource | postcodeBindingSource |
|                       |                                  | DataMember | FKResidencePostcode   |
| ResidenceTableAdapter | residenceTableAdapter            |            |                       |

**Comment:** The details of the second binding source should indicate that it is the foreign key binding source, so including 'fk' in the names is important here.

### (b) (i) Postcode application – extract

|                 | Name                  | Property      | Initial value                                                          |
|-----------------|-----------------------|---------------|------------------------------------------------------------------------|
| <b>Controls</b> |                       |               |                                                                        |
| ComboBox        | postcodeComboBox      | DataSource    | postcodeBindingSource                                                  |
|                 |                       | DisplayMember | Postcode                                                               |
| TextBox         | placeNameTextBox      | DataBindings  | <b>Binding:</b> Text, postcodeBindingSource, PlaceName                 |
| TextBox         | internetTextBox       | DataBindings  | <b>Binding:</b> Text, postcodeBindingSource, InternetLevel             |
| DataGridView    | residenceDataGridView | DataSource    | fkResidencePostcodeBindingSource                                       |
| TextBox         | connectionTextBox     | DataBindings  | <b>Binding:</b> Text, fkResidencePostcodeBindingSource, ConnectionType |

(ii) The combo box would have its `DropDownStyle` property set to `DropDownList`, the text box `placeNameTextBox` would be made read-only, and the editing permissions for users would be set to `False` for the data grid view.

(c) (i) `postcodeBindingSource.EndEdit()`  
`fkResidencePostcodeBindingSource.EndEdit()`

**Try**

```
postcodeTableAdapter.Update(postcodeDataDataSet.Postcode)
residenceTableAdapter.Update(postcodeDataDataSet.Residence)
MessageBox.Show("Update succeeded.")
```

**Catch** `ex As Exception`

```
MessageBox.Show("There has been a problem with updating the database.")
```

**End Try**

(ii) `postcodeBindingSource.EndEdit()`  
`fkResidencePostcodeBindingSource.EndEdit()`

(iii) The event handler for the form closing event should be linked to the event handler for the *Save* menu item click event.