



The Open University

MT264/Rev

Module Examination 20??

Designing applications with Visual Basic

Someday nth Somemonth 20?? xx am – yy pm

Time allowed: 3 hours

There are **TWO** parts to this examination. You should attempt **BOTH**.

You should attempt **ALL** the questions in Part 1.

You should attempt **two** of the three questions in Part 2. If you attempt all three questions then only your best **two** answers will count towards your examination score.

You are advised to spend about 100 minutes on Part 1 and 70 minutes on Part 2, and to leave yourself about 10 minutes for checking. Part 1 carries 64% of the total marks and Part 2 carries 36%.

You are advised to show all your working and to give reasons for all your answers unless a question is explicitly phrased 'Write down ...'. You should begin each answer on a new page of the answer book.

At the end of the examination

Check that you have written your personal identifier and examination number on **each** answer book used. **Failure to do so will mean that your paper cannot be identified.**

Put all your used answer books together with your signed desk record on top. Fasten them in the top left corner with the round paper fastener. Attach this question paper to the back of the answer books with the flat paper clip.

The use of calculators is not permitted in this examination.

Part 1

You should attempt **ALL** the questions in Part 1 and are advised to spend about **100 minutes** on it. This part carries 64% of the marks for the whole examination.

The marks for each question are given beside the question.

Question 1A (8 marks)

- (a) Write design code to make the Width property of a button, aButton, have the value 40;
- (b) Suppose that aNumUpDn is a NumericUpDown control. What is the effect of executing the following design code?
Set aNumUpDn.Value To 10
- (c) Write down design code for the following tasks:
- ▶ declare a string variable, aString;
 - ▶ initialise aString to the value "Name";
 - ▶ set the Text property of a label control, aLabel, to aString.
- (d) What property of a ComboBox control allows the developer to specify a list of items from which a user can select one at run time? When would you use such a ComboBox control rather than a TextBox control into which the user would type a value? Give an example.

Question 1B (8 marks)

- (a) Write design code to set the MaxSelectionCount property of a MonthCalendar control, called aMonthCal, so that a single date can be selected;
- (b) Suppose that aListBox is a ListBox control on a form. What is the effect of executing the following design code?
Set aListBox.Visible To True
- (c) Write down design code for the following tasks:
- ▶ declare a variable aColor of type Color;
 - ▶ initialise aColor to Color.Yellow;
 - ▶ set the BackColor property of a label control, aLabel, to aColor.
- (d) What property of a PictureBox allows the specification of a picture for display on the control?

Question 2A (8 marks)

Suppose that there is a class `Trophy` with the following description.

Class table: Trophy

Properties

Style **As** String

Finish **As** String

HeightInInches **As** Integer

Inscription **As** String

Constructor

New

Methods

isEngraved() **As** Boolean

costOfInscription(costPerLetter **As** Integer) **As** Integer

Constructor New

'An object of type Trophy is created with Style, Finish and Inscription set to the empty string, and HeightInInches set to 0.

End Constructor

Method isEngraved() **As** Boolean

'Preconditions: none

'Postconditions: If Inscription is not the empty string, then True is returned. Otherwise False is returned.

End Method

Method costOfInscription(costPerLetter **As** Integer) **As** Integer

'Preconditions: costPerLetter is not negative.

'Postconditions: Returns the cost of engraving the inscription as an integer number of pence, calculated as the total cost for each non-space in the inscription string.

End Method

- (a) What is the method identifier of the method with no parameter given above?
- (b) Does this method return a value? If so, what is the type of the return value?
- (c) The other method has a parameter. What type is it?
- (d) Is it possible that either method changes the state of the object on which it is invoked? Briefly explain your answer.
- (e) (i) Write design code to declare and create an object `aTrophy` of type `Trophy` so that its `Style` property is "Cup", its `Finish` property is "Silver", its `HeightInInches` property is 10 and its `Inscription` property is "The Winner, 2009"
- (ii) What value has the variable `cost` after execution of the following design code fragment?

`cost As Integer`

'The design code you have written for (i)

Set `cost` to `aTrophy.costOfInscription(10)`

Question 2B (8 marks)

Suppose that there is a class `CollectingTin` with the following description.

Class table: CollectingTin

Properties

Charity **As** String

Collector **As** String

AmountCollected **As** Integer

Constructor

New

Method

addDonation(donation **As** Integer)

Constructor New

'An object of type `CollectingTin` is created with `Charity` and `Collector` set to the empty string, and `AmountCollected` set to 0.

End Constructor

Method addDonation(donation **As** Integer)

'Preconditions: donation is not negative.

'Postconditions: donation is added to the current `AmountCollected`.

End Method

- (a) What is the method identifier of the only method given above?
- (b) Does this method return a value? If so, what is the type of the return value?
- (c) Does this method have any parameters? If so, what type are they?
- (d) Is it possible that the method changes the state of the object on which it is invoked? Briefly explain your answer.
- (e) (i) Write design code to declare and create an object `aTin` of type `CollectingTin` so that its `Charity` property is "Help the Aged", its `Collector` property is "Sue" and `AmountCollected` property is 0.
- (ii) What value has the variable `totalCollected` after execution of the following design code fragment?

received **As** Integer

totalCollected **As** Integer

Set received **To** 100

'The design code you have written for (i)

`aTin.addDonation(received)`

Set totalCollected **To** `aTin.AmountCollected`

Question 3A (8 marks)

Suppose that the following design code has been executed.

```
aDictionary As Dictionary(Of String, Integer)
Set aDictionary To New Dictionary(Of String, Integer)
total As Integer
Set total To 0
'code to add pairs of strings (representing names) and numbers
'(representing ages) to aDictionary
For Each key As String In aDictionary.Keys
    If key < "N" Then
        Set total To total + aDictionary.Item(key)
    End If
End For
```

- (a) Explain what this segment of design code calculates.
- (b) If the purpose was instead to find the number of entries in the dictionary whose key (string) began with one of the letters 'A' to 'M' then suggest what change(s) you would make to the code above.

Question 3B (8 marks)

Suppose that the following design code has been executed.

```
aList As List(Of Integer)
total As Integer
Set aList To New List(Of Integer)
Set total To 0
For index As Integer From 1 To 10
    aList.add(index)
End For
aList.remove(4)
aList.removeAt(5)
For Each number As Integer In aList
    Set total To total + number
End For
```

- (a) What value would be returned by aList.contains(5)?
- (b) What value would be returned by aList.Item(2)?
- (c) What value is returned by aList.Count?
- (d) Write down the value of total and explain how you calculated it.

Question 4A (8 marks)

This question concerns the following enumerated type, SubjectType, and two classes Photo and DigitalPhoto.

```
Enum SubjectType
    Landscape
    Portrait
    Action
    StillLife
End Enum
```

Class table: Photo

Properties

```
Subject As SubjectType
Colour As Boolean
Album As String
```

Constructors

```
Protected New
Protected New(aSubject As SubjectType)
```

Method

```
isThumbnail() As Boolean
```

Constructor New

'Creates a Photo object with Subject set to Nothing, Album set to the 'empty string and Colour set to False.

End Constructor

Constructor New(aSubject As SubjectType)

'Creates a Photo object with Subject set to aSubject, Album set to the 'empty string and Colour set to False.

End Constructor

Protected Method isThumbnail() As Boolean

'Not implemented

End Method

Class table: DigitalPhoto Inherits Photo

Properties

```
Size As Integer
```

Constructor

```
New
```

Constructor New

'Creates a DigitalPhoto object with Subject set to Nothing, Album set to the 'empty string and Colour set to False and Size set to 0.

End Constructor

Overrides Method isThumbnail() As Boolean

'Precondition: none

'Postcondition: if Size is less than 25600 then True is returned.

'Otherwise False is returned.

End Method

- (a) Suppose that the class `Photo` is implemented according to the description given above. Briefly explain why `Photo` might be considered to be an abstract class.
- (b) `Photo` has more than one constructor. Is this an example of overloading or overriding? Explain your answer.
- (c) Write design code to implement the constructor of `DigitalPhoto`. There is no need to include a comment.
- (d) Write a fragment of design code that
- ▶ declares an instance, `aDigitalPhoto`, of `DigitalPhoto`;
 - ▶ creates the instance as a digital photo with `Subject` set to `Portrait`, `Album` set to `"Family"`, `Colour` set to `true` and `Size` set to `125600`.

Question 4B (8 marks)

This question concerns the following two classes Pet and Cat. [Other classes that might be added in the future include Tortoise and Fish].

Class table: Pet

Properties

DateOfBirth **As** String
Colour **As** String
Name **As** String
FavouriteFood **As** String

Constructor

New

Method

isYoungster() **As** Boolean Protected, Overridable

Constructor New

'Creates a Pet object with DateOfBirth, Colour, FavouriteFood and Name set to the empty string.'

End Constructor

Protected Overridable Method isYoungster() **As** Boolean

'Preconditions: none'

'Postconditions: True is returned if the pet is young, based on its DateOfBirth.'

'False is returned otherwise.'

End Method

Class table: Cat Inherits Pet

Property

VaccinationRecord **As** Dictionary(Of String, String)

Constructor

New

Methods

isTimeForVaccination() **As** Boolean
isYoungster() **As** Boolean Protected, Overrides

Constructor New

'Creates a Cat object with DateOfBirth, Colour, FavouriteFood and Name set to the empty string, and VaccinationRecord set to a new empty instance of Dictionary(Of String, String).'

End Constructor

Protected Overrides Method isYoungster() **As** Boolean

'Preconditions: none'

'Postconditions: True is returned if the cat is less than 2 years old,

'based on its DateOfBirth. False is returned otherwise.'

End Method

Method isTimeForVaccination() **As** Boolean

'Preconditions: none'

'Postconditions: If VaccinationRecord is empty, or the latest entry is over 1 year old then True is returned. False is returned otherwise.'

End Method

- (a) Can the method `isYoungster` be invoked on an instance of `Cat`? Briefly explain your answer.
- (b) Can the method `isTimeForVaccination` be invoked on an instance of `Cat`? Briefly explain your answer.
- (c) Write design code to implement the constructor of `Cat`. There is no need to include a comment.
- (d) Write a fragment of design code that
 - ▶ declares an instance, `moggy`, of `Cat`;
 - ▶ creates the instance as a cat with `DateOfBirth` set to "01/01/2000", `Colour` set to "Black", `Name` set to "Moggy" and `VaccinationRecord` set to a new empty instance of `Dictionary(Of String, String)`.
- (e) What is the effect of declaring the method `isYoungster` as `Overridable` in `Pet`?

Question 5A (8 marks)

- (a) Write design code for the body of an event handler that enables selection of a font. The code should allow a user to choose a specific font and this choice should be recorded using the variable `newFont`, but with the style always recorded as bold. You may assume that a `FontDialog` control, `aFontDialog`, has been added to the form, and that a variable `newFont` of type `Font` has been declared. You may find it helpful to use a local variable `aFont` of type `Font`.
There is no need to include a comment for the event handler.
- (b) Complete the following fragment of design code as indicated by the comments. You should assume that `fileName` is an initialised string variable, giving a file name.

```

aReader As StreamReader
aLine As String
Try
    'create aReader as a stream reader that reads from fileName
    'read one line of text from fileName and assign it to aLine
Catch ex As Exception
    'display a message box warning that the file could not be read
Finally
    'close aReader, if necessary
End Try

```

Question 5B (8 marks)

- (a) Suppose that there is a text box, a `TextBox`, in a dialog box `aDialog` of type `ExamDialog`. Consider the following code fragment:

```
aDialog As ExamDialog
aString As String
finished As Boolean
todaysDate As String
'code to initialise todaysDate to today's date in the form "dd/mm/yyyy"
Set aDialog To New ExamDialog
Set aString To ""
Set finished To False
While Not finished
    Set aDialog.aTextBox.Text To "Enter the date of your birth in the form dd/mm/yyyy"
    If aDialog.ShowDialog() = DialogResult.OK Then
        If aDialog.aTextBox.Text = todaysDate Then
            Set aString To "Today " + aDialog.aTextBox.Text + _
                " is your birthday!"
        Else
            Set aString To "Not yet :( "
        End If
    Else
        Set finished To True
    End If
End While
aDialog.dispose()
```

- (i) What is the effect of the code in the loop body?
(ii) What is the loop control and how is its value changed?
(iii) Briefly describe the overall effect of the code fragment as observed by a user of the application.
- (b) Consider the following fragment of code and answer the questions below. You should assume that `wanted` is an initialised `String` variable.

```
aReader As StreamReader
nextLine As String
count As Integer
Set count To 0
Try
    Set aReader To StreamReader("names.txt")
    While Not aReader.EndOfStream
        If (wanted = aReader.ReadLine()) Then
            Set count To count + 1
        End If
    End While
Catch ex As Exception
    MessageBox.Show("File could not be read")
Finally
    If aReader IsNot Nothing Then
        aReader.close()
    End If
End Try
```

- (i) Why has the line `aReader.close()` been placed inside the **Finally** part of **Try Catch**?
(ii) Why does the code check **If** `aReader IsNot Nothing`?
(iii) What does the variable `count` count?

Question 6A (8 marks)

- (a) Write design code to declare and create an instance `aPlayer` of `SoundPlayer`, set its `SoundLocation` property to `"music.wav"` and attempt to play this file repeatedly in the background. If the latter fails, a message box should be displayed, warning that the music file could not be played.
- (b) Write design code for the `OnPaint` event handler of a panel, `canvasPanel`. The code should draw at the bottom of the panel a red filled rectangle of height 50, filling the whole width of the panel. With its top left corner in the centre of the rectangle, the word "Stop" should be written in black, using the font `newFont`.

There is no need to include a comment for the event handler.

Question 6B (8 marks)

- (a) Write design code using the `Audio` object `My.Computer.Audio` to attempt to play the file `"music.wav"` repeatedly in the background. If this fails, a message box should be displayed, warning that the music file could not be played.
- (b) (i) Sketch or describe the effect of the following design code. You should assume that the code is in the `OnPaint` event handler of a panel, `canvasPanel`, and that `canvasPanel` is sufficiently large to contain the drawing. `aFont` is a standard font. `anImage` is some image. `fGraphics` has been previously declared and initialised as the `Graphics` object of `canvasPanel`.

```
fGraphics.drawLine(Pens.Black, 0, 0, 200, 200)
fGraphics.drawImage(anImage, 200, 200)
fGraphics.drawString("My Picture", aFont, Brushes.Blue, 100, 100)
```

- (ii) What line of code would be required to initialise `fGraphics`?

Question 7A (8 marks)

Suppose that the class `Dragonfly` inherits from `BaseSprite`. There is to be a method `hoverUp` of `Dragonfly` that sets the unit of movement to 0 in the x-direction and -10 in the y-direction.

- (a) Give the design code implementation of the method `hoverUp`. There is no need to include a comment.
- (b) Write a fragment of design code that declares and creates an instance `dragon` of `Dragonfly`, sets it to hover up and makes it move (hover up) twice.
- (c) Write down a design code **If** condition to check that hovering up once more will not move the top of `dragon` move above the top edge of a panel on which it is displayed.

Hint: The top edge of the panel will have y-coordinate equal to 0.

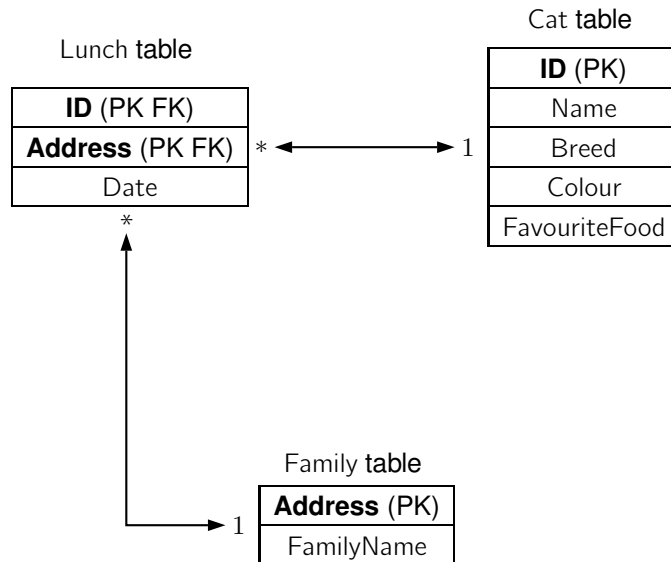
Question 7B (8 marks)

Suppose that the class `Dragonfly` inherits from `BaseSprite`. There is to be a method `zoomRight` of `Dragonfly` that sets the unit of movement to 15 in the x-direction and 0 in the y-direction.

- (a) Give the design code implementation of the method `zoomRight`. There is no need to include a comment.
- (b) Write down a design code **If** statement that will display a message box saying 'Game over' if the instance `dragon` of `Dragonfly` cannot fully be displayed within the game panel, `aPanel` (i.e. when `dragon` has moved partially out of the panel area).
Hint: You should use `dragon`'s `BoundingBox` property.
- (c) Rewrite your **If** condition given in part (b) so that the 'Game over' message gets shown only when all of the `dragon` as displayed on screen has left the panel.

Question 8A (8 marks)

(a) This part is about a database with the following ER diagram.



Some sample data is given in the tables below.

Cat table

ID	Name	Breed	Colour	FavouriteFood
C1	Biddy	Persian	White	Tuna
C2	Fluff	Manx	Black	Chicken

Lunch table

ID	Address	Date
C1	AA10 2BB	1/1/09
C1	AA10 3CC	2/1/09
C2	AA10 2BB	2/1/09

Family table

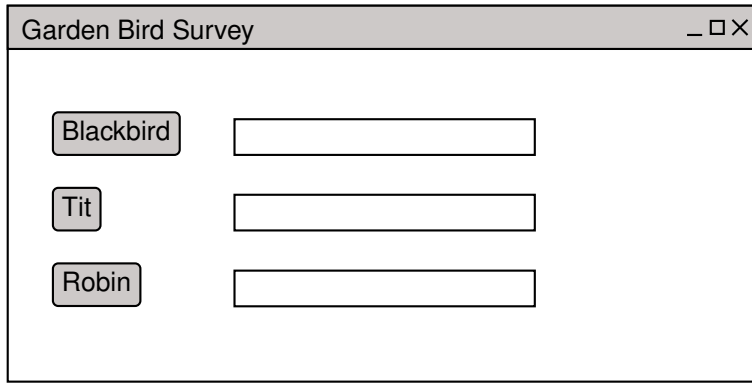
Address	FamilyName
AA10 2BB	Smith
AA10 3CC	Jones

- (i) Write an SQL statement that returns a table giving the names of all the cats that are black and whose favorite food is tuna.
- (ii) Describe in general terms the result table that will be returned by the following SQL statement.

```

SELECT Family.FamilyName
FROM Family INNER JOIN Lunch
ON Family.Address = Lunch.Address
WHERE Lunch.Date = '2/1/09';
    
```

- (b) The following sketch and property table show a graphical user interface for a simple garden bird survey application. Note that the user is expected to press the buttons to record any sightings of the birds.



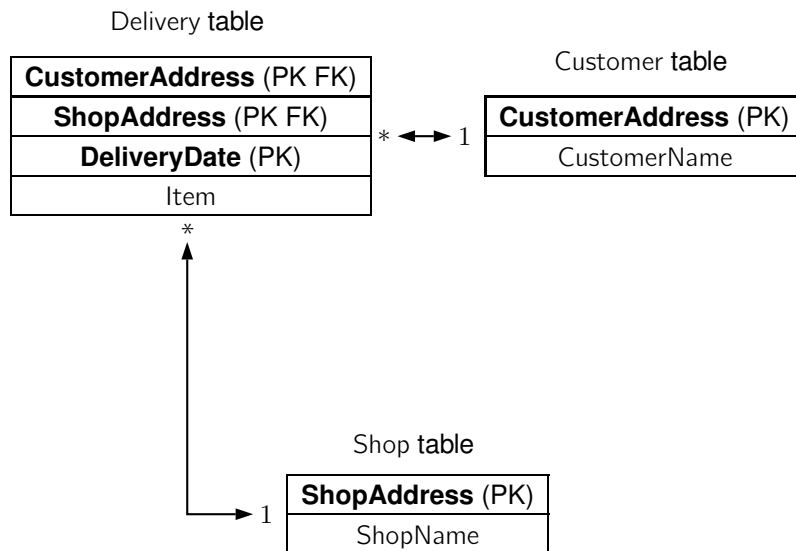
Property table: Garden Bird Survey application

	Name	Property	Initial value
Container			
Form	MainForm	Text	Garden Bird Survey application
Control			
Button	blackbirdButton	Text	Blackbird
TextBox	blackbirdTextBox	Text	
Button	titButton	Text	Tit
TextBox	titTextBox	Text	
Button	robinButton	Text	Robin
TextBox	robinTextBox	Text	

- (i) Identify and briefly discuss a problem with the design of this application, relating it to one of the design principles. You should name the design principle involved.
- (ii) Identify and briefly discuss how one of the design principles has been successfully considered in this application design. You should relate your answer to and name the design principle involved.
- (iii) Identify one usability goal which you believe is not achieved. For this usability goal, briefly explain how the graphical user interface could be modified to improve the usability.

Question 8B (8 marks)

(a) This part is about a database with the following ER diagram.



Some sample data is given in the tables below.

Customer table

CustomerAddress	CustomerName
1 The Crescent	Mr and Mrs Smith
54 London Road	Mr Jones

Delivery table

CustomerAddress	ShopAddress	Date	Item
1 The Crescent	2 High Street	1/1/09	Patio swing chair
1 The Crescent	2 High Street	1/4/09	Gas powered barbecue
54 London Road	35 Main Road	2/11/08	10 bags cement

Shop table

ShopAddress	ShopName
2 High Street	The Hardware Store
35 Main Road	Local Builders Merchant

- (i) Write an SQL statement that returns a table giving the names of all the shops who have made a delivery to the address "1 The Crescent".
- (ii) Describe in general terms the result table that will be returned by the following SQL statement.

```

SELECT Customer.CustomerName
FROM Customer INNER JOIN Delivery
ON Customer.CustomerAddress = Delivery.CustomerAddress
WHERE Delivery.DeliveryDate = '1/4/09';
    
```

- (b) (i) Give two examples of situations arising that could require changes to be made to a GUI, where the underlying model does not require to be changed.
- (ii) Explain one difficulty that may arise if an application has not been designed with separation of the view and the underlying model.
- (iii) Explain one advantage (not the opposite of the difficulty discussed in (b)) of separating the view and the underlying model.

Part 2

In Part 2, marks will be given for your best **two** answers, and you are advised to spend about **70 minutes** on it.

Each question carries 18% of the marks for the whole examination, and an indication of the allocation of marks within each question is given beside the question.

Question 9 (18 marks)

This question is about a model for an application to support maintenance of records of pets in a veterinary surgery.

The descriptions of the two classes to be used are given below. An object of type PetAdmin has a Pets property that gives the list of recorded pets.

Class table: Pet

Properties

Name **As** String
Species **As** String
Description **As** String
Colour **As** String

Constructor

New

Constructor New

'Creates a Pet object with all four fields set to the empty string.'

End Constructor

Class table: PetAdmin

Property

Pets **As** List(Of Pet) ReadOnly

Constructor

New

Methods

addPet(addName **As** String, spec **As** String, descr **As** String, col **As** String)
isPet(isName **As** String) **As** Boolean
removePet(removeName **As** String)
speciesCount(thisSpecies **As** String) **As** Integer

Constructor New

'Creates a PetAdmin object with an empty list of pets.'

End Constructor

Method addPet(addName **As** String, spec **As** String, descr **As** String, col **As** String)

'Preconditions: addName is not already the name of a pet in the list.'

'Postconditions: A new entry for this pet, with these fields, is added to the list.'

End Method

Method isPet(isName **As** String) **As** Boolean

'Preconditions: The list of pets is not empty.'

'Postconditions: If isName is a name of a pet already in the list, then True is returned, else False is returned.'

End Method

Method removePet(removeName **As** String)

'Preconditions: removeName is the name of a pet already in the list.

'Postconditions: The Pet object with name removeName is removed from the list.

End Method

Method speciesCount(thisSpecies **As** String) **As** Integer

'Preconditions: none

'Postconditions: The number of pets in the list with species thisSpecies is returned.

End Method

- (a) Write the design code for the method addPet in the class PetAdmin. There is no need to include a comment.
- (b) Write the design code for the method speciesCount. There is no need to include a comment.
- (c) Explain the purpose of each line in the following design code for the method isPet. The lines have been numbered for your convenience.
- (1) thisPet **As** Pet
 - (2) index **As** Integer
 - (3) **Set** index **To** 0
 - (4) **Set** thisPet **To** Pets.Item(index)
 - (5) **While Not** thisPet.Name = isName And index < Pets.Count – 1
 - (6) **Set** index **To** index + 1
 - (7) **Set** thisPet **To** Pets.Item(index)
 - (8) **End While**
 - (9) **Return** (thisPet.Name = isName)
- (d) Suppose that a form in the graphical user interface of the application has text boxes for the display of the name, species, description and colour of a pet, and that there is a button which when clicked will delete a pet of the name on display. What checks would you have to put in the event handler for this button before invoking removePet on the PetAdmin object?

Question 10 (18 marks)

This question concerns the class `TextSearcher` with the following description.

Class table: TextSearcher**Property**

Text **As** String

Constructor

New

Methods

doesItContain(word **As** String) **As** Boolean

everyOther() **As** String

Constructor New

'Creates a TextSearcher object with Text set to the empty string.

End Constructor**Method** doesItContain(word **As** String) **As** Boolean

'Preconditions: word is not the empty string.

'Postconditions: If word appears somewhere in Text, then True is returned.

'Otherwise False is returned.

End Method**Method** everyOther() **As** String

'Preconditions: none

'Postconditions: If Text is the empty string then the empty string

'is returned. Otherwise a string is returned consisting of the first

'and then every other character in Text.

End Method

- (a) Write design code for the method `doesItContain`. There is no need to include a comment.

You must not use the String method `indexOf(wanted As Char) As Integer`.

You may wish to use the String method

`substring(position As Integer, len As Integer)`, which returns the string that is formed from the string on which it has been invoked by taking the substring starting at index position and which has length len.

- (b) Assume that `Text` is "Baa baa, black sheep, have you any wool?". Give three examples of `word` with which to test your code. You should choose the examples such that each will test a significantly different aspect of your code. Briefly explain what aspect each example is intended to test.
- (c) Write design code for the method `everyOther`. There is no need to include a comment.
- (d) Give three examples of `Text` with which to test your code. You should choose the examples such that each will test a significantly different aspect of your code. Briefly explain what aspect each example is intended to test.
- (e) The class `TextSearcher` is to be extended by the following method.

Method editedText() **As** String

result **As** String

i **As** Integer

nextChar **As** String

Set result **To** "" *'empty string*

Set i **To** 0

While i < Text.Length - 2

Set nextChar **To** Text.Chars(i).ToString()

Select Case nextChar

Case ".", "!", "?"

If Text.substring(i + 1, 2) = " " **Then** *'checking for two spaces*

```

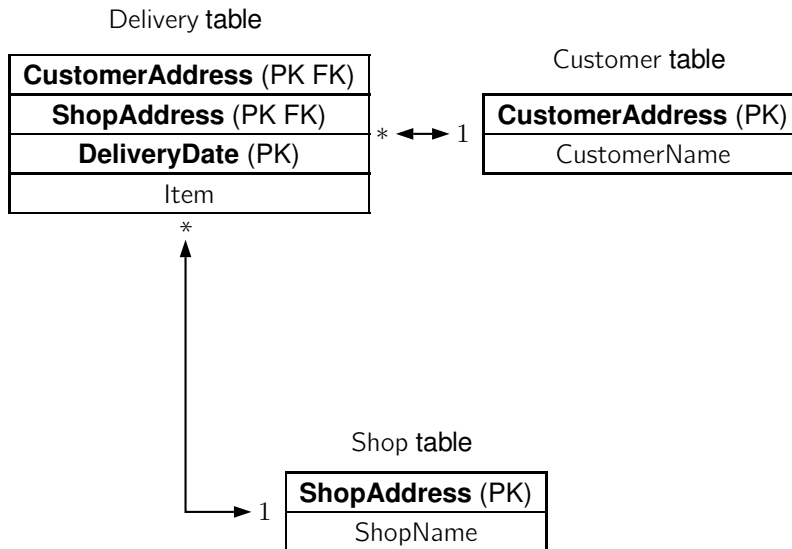
        Set result To result + nextChar + " " 'one space
        Set i To i + 3
    Else
        Set result To result + nextChar
        Set i To i + 1
    End If
Case Else
    Set result To result + nextChar
    Set i To i + 1
End Select
End While
While i < Text.Length
    Set result To result + Text.Chars(i).ToString()
    Set i To i + 1
End While
Return result
End Method

```

Briefly describe what this method does. Note that the comments in the code explain that the strings in question are the empty string, a string consisting of two space characters and a string consisting of one space character.

Question 11 (18 marks)

This question is about an application that gives access to a database concerning shop deliveries to customers. The ER diagram is shown below.



Some sample data is given in the tables below.

Customer table

CustomerAddress	CustomerName
1 The Crescent	Mr and Mrs Smith
54 London Road	Mr Jones

Delivery table

CustomerAddress	ShopAddress	Date	Item
1 The Crescent	2 High Street	1/1/09	Patio swing chair
1 The Crescent	2 High Street	1/4/09	Gas powered barbecue
54 London Road	35 Main Road	2/11/08	10 bags cement

Shop table

ShopAddress	ShopName
2 High Street	The Hardware Store
35 Main Road	Local Builders Merchant

The specification for the application is given below.

Project specification: Shop Deliveries

The application is intended to allow the user to view partial information from the shop deliveries database and make some edits to deliveries. The application should possess:

- ▶ a way to view the name and address for each shop, and a way to select a shop;
- ▶ a way to view the delivery data from the Delivery table for the currently selected shop;
- ▶ a way to edit the delivery data for the currently selected shop (including adding and deleting of deliveries);
- ▶ a way to save the updates to the database;

The application should not allow any Shop (or Customer) data to be edited.

- (a) Draw a labelled sketch of a possible GUI for this application. The labels should give the types of controls used. [4]
- (b) Explain how you would ensure that the user's editing permissions are correctly set. [2]
- (c) Draw up a property table for any non visual components. You may assume that the data set is called `shopDeliveryDataSet`. [5]
- (d) Draw up an extract of the main property table showing the data bindings (naming the relevant control) that can be set at design time. [3]
- (e) State which event handler will save the updates to the database. Write the design code body for this event handler. [4]

[END OF QUESTION PAPER]