|  |  |
| --- | --- |
| Chapter 5 Lecture Terms / Questions  Interface Design Philosophy  1.       What is the design philosophy for proper application interface?  Keep it simple  2.       Describe the first consideration  Determine what is needed  Inputs & outputs  3.       What are the possible functions of a data base interface?  Delete. Edit. Add. Search. Sort. Print  4.       Why do you have to worry about user errors?  You have to check for possible errors  Make it so users can’t make many errors  5.       What’s an example given of preventing a user error  Limiting to specific data types  Dropdowns  6.       What does using “point and click” tools mean here?  Preexisting options are chosen by users  7.       Why is it useful?  Does not allow for mistakes to be made by the program  8.       How should you make your code readable & traceable?  Use efficient naming methods  Use comments to explain your code  Debug as much as possible  9.       List the types of validation is done on the form  Disable Controls  Popups  Restricting Datatype  Cursor appears in specific areas  Tab/Enter to move on | 10.   How does the programmer make the code understandable?  Using Comments throughout the code  Form Control  11.   List and describe the form properties  AcceptButton- gets/sets the button that is clicked when the user presses ENTER  CancelButton- gets/sets the button that is clicked when the user presses ESC  ControlBox- whether a control box is displayed on the form  Enabled- if form is disabled, all controls are disabled  MaximizeButton- button exist in control box?  MinimizeButton- button exist in control box?  BackColor- Get or sets the form background color  Font- Gets or sets font name, style, size.  ForeColor- Gets or sets color of text or graphics.  FormBorderStyle- Sets the form border to be fixed or sizeable.  Height- Height of form in pixels.  Help- Gets or sets a value indicating whether a Help button should be displayed in the caption box of the form.  Icon- Gets or sets the icon for the form.  Left- Distance from left of screen to left edge of form, in pixels.  StartPosition- Gets or sets the starting position of the form when the application is running.  Text- Gets or sets the form window title.  Top- Distance from top of screen to top edge of form, in pixels.  Width- Width of form in pixels.  12.   List & describe the form methods  Close  Focus  Hide- makes form invisible  Refresh  Show- makes form visible |
| 13.   List/describe form events  a.       Activated- when form is activated by code or user  b.       Click- when form is clicked by user  c.       FormClosing- when form is closing  d.       DoubleClick- when form double clicked  e.       Load- when form loads  f.        Paint- when form is redrawn  14.   What do we usually do to set up a form  a.       Use this.methodName();  b.       If using multiple forms, specify the form name rather than this   Button Control  15.   List and describe the button properties  a.       Enabled- btnPrevious.Enabled = true;  b.       Name- Gets or sets the name of the button (three letter prefix for button name is btn).  c.       BackColor- Get or sets the button background color.  d.       Font- Gets or sets font name, style, size.  e.       ForeColor- Gets or sets color of text or graphics.  f.        Image- Gets or sets the image that is displayed on a button control.  g.       Text- Gets or sets string displayed on button.  h.       TextAlign- Gets or sets the alignment of the text on the button control.  16.   List & describe the button methods  a.       Focus-  sets focus to btn                                                                 i.      btnDone.Focus();  b.       PerformClick- creates aclick event for a btn  17.   List/describe button events  a.       Click  18.   What do we usually do to set up a button  b.       Clear- clears all text in textbx  c.       Focus- places cursor in a textbx  d.       SelectAll- selects all text in box  e.       Undo    i.      txtAuthorName.Clear();  25.   List/describe textbox events  a.       KeyDown- whena key is pressed  b.       KeyPressed-  when a key is pressed; knows what key is pressed  c.       Leave- leave the text box, tab away  d.       TextChanged-  when text property is changed  26.   What do we usually do to set up a textbox   CheckBox Control  27.   List and describe the checkbox properties  a.       Checked  28.   List & describe the checkbox methods  29.   List/describe checkbox events  a.       Click  b.       CheckedChanged  30.   What do we usually do to set up a checkbox  RadioButton Control  31.   List and describe the radiobutton properties  a.       Name- Gets or sets the name of the radio button (three letter prefix for radio button name is rdo).  b.       AutoSize- Gets or sets a value indicating whether the radio button is automatically resized to display its entire contents.  c.       BackColor- Get or sets the radio button background color.  d.       Checked- Gets or sets a value indicating whether the radio button is checked.  e.       Font- Gets or sets font name, style, size.  f.        ForeColor- Gets or sets color of text or graphics.  g.       TextAlign- Gets or sets the alignment of text of the radio button.  Panel Control  39.   What is a panel?  The Panel control is another Visual C# grouping control.  40.   How is it different from a groupbox?  It is nearly identical to the GroupBox control in behavior. The panel control lacks a Text property (titling information), but has optional scrolling capabilities. Radio buttons in the panel control act as an independent group. Panel controls can also be used to display graphics (lines, curves, shapes, animations).  41.   List and describe the panel properties  Name Gets or sets the name of the panel (three letter prefix for panel name is pnl).  AutoScroll Gets or sets a value indicating whether the panel will allow the user to scroll to any controls placed outside of its visible boundaries.  BackColor Get or sets the panel background color.  BorderStyle Get or set the panel border style.  Enabled Gets or sets a value indicating whether the panel is enabled. If False, all controls in the panel are disabled.  Visible If False, hides the panel (and all its controls).  42.   What do we usually do to set up a panel   * Set Name property. * Place desired controls in panel control. * Monitor events of controls in panel using usual techniques.   PictureBox Control  43.   List and describe the picturebox properties  Name Gets or sets the name of the picture box (three letter prefix for picture box name is pic).  BackColor Get or sets the picture box background color.  BorderStyle Indicates the border style for the picture box.  Height Height of picture box in pixels.  Image Establishes the graphics file to display in the picture box (jpeg, gif, bmp files).  Left Distance from left edge of form to left edge of picture box, in pixels.  SizeMode Indicates how the image is displayed.  Top Distance bottom of form title bar area to top edge of picture box, in pixels.  Width Width of picture box in pixels.  44.   List/describe picturebox events  Click Triggered when a picture box is clicked.  54.   How do you specify the default button that has focus (give options and code example) MessageBoxDefaultButton.Member 55.   List and describe the possible values returned by the messagebox  Abort The Abort button was selected  Cancel The Cancel button was selected  Ignore The Ignore button was selected  No The No button was selected  OK The OK button was selected  Retry The Retry button was selected  Yes The Yes button was selected  56.   Give an example of a C# statement that includes the messagebox in an if statement.  Look at how each part is written,  a.      if (MessageBox.Show("This is an example of a message box", "Message Box Example", MessageBoxButtons.OKCancel, MessageBoxIcon.Information, MessageBoxDefaultButton.Button1) == DialogResult.OK)  {  // everything is OK  }  else  {  // cancel was pressed  }  57.   Give an example of using a messagebox where the program does not use any returned value from it. (It’s only used to show the user something.)  MessageBox.Show("Quick message for you.", "Hey You!");  58.   When should messageboxes be used?  Message boxes should be used whenever your application needs to inform the user of action or requires user feedback to continue. It is probably better to have too many message boxes, than too few. You always want to make sure your application is performing as it should and the more information you have, the better.  Application State  59.   What is application state  Application state implies knowing just what is currently being done within the interface. Are you adding a record, editing a record, deleting a record, or perhaps leaving the application? Once you know the state the application is in, you adjust the interface so that options needed for that particular state are available to the user. You also need to know when and how to transition from one state to another.  61.   How do you use enable/disable based on the application state  By setting Enabled to False, you disable a control, making it unavailable to the user  62.   How would you use the ReadOnly property based on the state  If a value in a text box is not to be edited, set ReadOnly to True. When editing is allowed (the state changes), toggle the ReadOnly property to False. For text boxes that are always read-only (used for display, not editing purposes), use color (red is good) to indicate they are not accessible. | Label Control  19.   List and describe the label properties  a.       Name- Gets or sets the name of the label (three letter prefix for label name is lbl).  b.       AutoSize- Gets or sets a value indicating whether the label is automatically resized to display its entire contents.  c.       BackColor- Get or sets the label background color.  d.       BorderStyle- Gets or sets the border style for the label.  e.       Font- Gets or sets font name, style, size.  f.        ForeColor- Gets or sets color of text or graphics.  g.       Text- Gets or sets string displayed on label.  h.       TextAlign- Gets or sets the alignment of text in the label.  20.   List & describe the label methods  a.       Refresh- Forces an update of the label control contents.  21.   List/describe label events  a.       Click- Event triggered when user clicks on a label.  b.       DblClick - Event triggered when user double-clicks on a label.  22.   What do we usually do to set up a label   TextBox Control  23.   List and describe the textbox properties  a.       Hide Selection- whether selected text remains selected when control loses focus  b.       Lines- gets/sets lines of text in a txtbx  c.       MultiLine-  d.       ReadOnly-  e.       SelectedText- gets/ sets a value indicating the currently selected text in the control   i.      MessageBox.Show(txtAuthorName.SelectedText);  24.   List & describe the textbox methods  a.       AppendText-  appends text to the current textbox  32.   List & describe the radiobutton methods  a.       Focus- Moves focus to this radio button.  b.       PerformClick- Generates a Click event for the button, simulating a click by a user.  33.   List/describe radiobutton events  a.       CheckedChanged- Occurs when the value of the Checked property changes, whether in code or when a radio button is clicked.  b.       Click- Triggered when a button is clicked.  Checked property is automatically changed by Visual C#.  34.   What do we usually do to set up a radiobutton   * Establish a group of radio buttons. * For each button in the group, set the Name (give each button a similar name to identify them with the group) and Text property. You might also change the Font, BackColor and Forecolor properties. * Initialize the Checked property of one button to True. * Monitor the Click or CheckChanged event of each radio button in the group to determine when a button is clicked. The ‘last clicked’ button in the group will always have a Checked property of True.   GroupBox Control  35.   What is a groupbox?  Group boxes provide a way of grouping related controls on a form  36.   How does a groupbox affect radio buttons?  Radio buttons within a group box act independently of other radio buttons in an application.  37.   List and describe the groupbox properties  Name Gets or sets the name of the group box (three letter prefix for group box name is grp).  BackColor Get or sets the group box background color.  Enabled Gets or sets a value indicating whether the group box is enabled. If False, all controls in the group box are disabled.  Font Gets or sets font name, style, size.  ForeColor Gets or sets color of text.  Text Gets or sets string displayed in title region of group box.  Visible If False, hides the group box (and all its controls).  38.   What do we usually do to set up a groupbox   * Set Name and Text property (perhaps changing Font, BackColor and ForeColor properties). * Place desired controls in group box. Monitor events of controls in group box using usual techniques.   45.   What do we usually do to set up a picturebox   * Set the Name and SizeMode property (most often, StretchImage). * Set Image property, either in design mode or at run-time, remembering icon files are not automatically displayed.   MessageBox Object  46.   What is a messagebox?  Small window w/ message  47.   What are the items you can put on a messagebox  Text, Caption, Buttons, Icons, Default Buttons  48.   How do you display it  To use the MessageBox object, you decide what the Text of the message should be, what Caption you desire, what Icon and Buttons are appropriate, and which DefaultButton you want. To display the message box in code, you use the MessageBox Show method.  49.   List the different statements to show a messagebox  MessageBox.Show(Text);  MessageBox.Show(Text, Caption);  MessageBox.Show(Text, Caption, Buttons);  MessageBox.Show(Text, Caption, Buttons, Icon);  MessageBox.Show(Text, Caption, Buttons, Icon, DefaultButton);  50.   List and describe the different values for the Buttons argument  Member Description  AbortRetryIgnore Displays Abort, Retry and Ignore buttons  OK Displays an OK button  OKCancel Displays OK and Cancel buttons  RetryCancel Displays Retry and Cancel buttons  YesNo Displays Yes and No buttons  YesNoCancel Displays Yes, No and Cancel buttons  51.   How would you write one of these values in the code  MessageBoxButtons.Member  MessageBoxButtons.OKCancel  52.   List and describe the different values for the Icon argument  IconAsterisk Displays an information icon  IconInformation Displays an information icon  IconError Displays an error icon (white X in red circle)  IconHand Displays an error icon  IconNone Display no icon  IconStop Displays an error icon  IconExclamation Displays an exclamation point icon  IconWarning Displays an exclamation point icon  IconQuestion Displays a question mark icon  53.   How would you write one of these values in the code MessageBoxIcon.Member 63.   How can Focus be used  When editing in a text box, use the Focus method to place the cursor in the box, making it the active control (giving it focus) and saving the user a mouse click. The Focus method can also be used to programmatically move the user from one text box to the next in a desired order.  64.   How do TabIndex and TabStop help you guide the user through a form  Another mechanism for moving from one control to another in a prescribed order is the TabIndex property, in conjunction with TabStop. If TabStop is True, TabIndex defines the order controls become active (only one control can be active at a time) as the <Tab> key is pressed (the order is reversed when <Shift>-<Tab> is pressed). When controls are placed on a form at design time, they are assigned a TabIndex value with TabStop = true;. |