50 points Due: Specified on Canvas

Assignment Purpose

This assignment will give you practice regular expressions and defining application-specific exceptions.

Mandatory Instructions

~~The main for of the application should display no contents.~~

~~The user can select the File or Edit menu. The File -> Merge File menu will allow the user to open a text file containing names and phone numbers. The file contents will be added to the list box.~~

~~The first name should be auto selected, and the corresponding phone number should be shown in the text box.~~

~~The user can then select the Edit -> Insert menu item, which will allow adding a new name/phone number.~~

Graphical user interface, application

Description automatically generated

~~Users can also start by inserting a new name/phone pair via the Edit -> Insert menu. Then the user can merge the contents of the data file.~~

~~Users can edit the phone number of the selected list box item.~~

~~Graphical user interface, application

Description automatically generatedUsers can also remove name/phone pair from the list box.~~

Graphical user interface, application

Description automatically generated

**~~Part 1: Define MyFieldException and FieldExceptionForm classes~~**

1. ~~To get started, download the File called PhoneApp SEED.zip from Canvas.~~
2. ~~The standard ApplicationException class might not provide all of the functionality you need in every situation. Define a new class called MyFieldException that inherits from ApplicationException.~~
3. ~~There should~~ ~~be three arguments to the constructor~~.

* ~~The first is a message that should be passed to the base class constructor (e.g., "Invalid format").~~
* ~~The second is a string holding the name of the field in error (e.g., "Name").~~
* ~~The third is a string giving the incorrect value (e.g., "12\*34").~~
* ~~Save the last two strings in private data, and define read-only properties allowing access to the information.~~
* ~~This is the type of object we'll throw when a formatting problem is found.~~

1. ~~Add a form to the project called FieldExceptionForm. The constructor should have one argument, an object of type MyFieldException. The form should display components of the error.~~

Graphical user interface, application

Description automatically generated

**~~Part 2: Modify the Phone class~~**

1. ~~Add two static functions to the Phone class: ValidateName and ValidatePhoneNumber.~~ 
   * ~~Each takes a string argument and returns a string.~~
   * ~~These functions need several regular expressions.~~
   * ~~You should define these as static Regex data members and initialize the Regex references in a static constructor.~~
2. ~~ValidateName takes a string argument and returns a string.~~ 
   * ~~It checks to verify that the string argument is a valid name.~~
   * ~~Use regular expressions to remove any leading or trailing whitespace from the argument. Then use regular expressions to verify that the name is valid~~.
   * ~~A valid name begins with valid characters, followed by a comma, followed by one or more valid characters and/or spaces.~~
   * ~~The valid characters for a name are letters, apostrophes, dashes, and spaces. Return the name as the value of the function~~.
   * ~~Throw MyFieldException if the name is invalid~~.
3. ~~ValidatePhoneNumber should use regular expressions to ensure that the phone number is formatted properly~~.
   * ~~The phone number can be entered in any of these four formats:~~ 
     1. ~~3722337, 372-2337, 4193722337, 419-372-2337.~~
   * ~~Regardless of how the number is originally entered, the string that's returned should be in the format 4193722337.~~
   * ~~Assume the 419 area code if none is supplied.~~
   * ~~Throw MyFieldException if an error occurs.~~
   * ~~Do not accept any other phone area codes besides 419.~~
4. ~~Modify the set method of the PhoneNumber property to call the ValidatePhoneNumber function and save the value returned as the value of the private data~~.
5. ~~Modify the set method of the Name property to call ValidateName and save the value returned as the value of the private data.~~
6. ~~Implement a read-only property called PhoneNumberDisplay that returns the phone number in the format 419-372-2337. Note that this is not the format in which the number is stored.~~ 
   * ~~Use a regular expression to build this format.~~

**~~Part 3: Modify PhoneForm~~**

1. ~~Add a button called Cancel.~~ ~~This button should set NewPhone to null and close the~~ ~~form.~~
2. Add data validation for all fields on the input form.
   * ~~Use the validation functions defined in the Phone class to validate the data.~~
   * ~~Add an ErrorProvider to the form and guarantee that errors must be fixed before going to a new field.~~
   * ~~However, it is always possible to press Cancel or to close the form, even if there is an error on the form.~~

**~~Part 4: Modify Edit Menu~~**

1. ~~You will need to add another menu item to the “Edit” menu~~, ~~namely “Modify…”~~ ~~which should bring up a form similar to the insert phone form, prefilled with the selected Phone object.~~
   * ~~The modify menu item should be grayed out when no phones are in the list box.~~ ~~All the data validation should also be performed in the Edit phone form.~~

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

~~When the user edits the name, let’s say “Smith, Amy” and presses the OK button. In the main Form, you will need to remove the selected item from the list (use lstPhones.RemoveAt(i)),~~ ~~then re-add the modified Phone object (lstPhone.Insert(i, phObj)). Be sure that the same items remains selected in the list box after the edit.~~

~~The Edit form can store the object passed to its constructor as a private data member, accessed via a read-only property from the main Form~~

**~~Part 5: Modify Form1~~**

1. ~~Remove both buttons and add two menus.~~ ~~The File menu contains menu items "Merge File...", "Save As...", and "Exit". The Phone menu contains "Edit...", "Insert...", "Remove...".~~ ~~The Merge File, Exit, and Insert items should initially be enabled, and the others should be disabled. Be sure to give meaningful names to the menu items (e.g., mnuMerge). Implement all of these commands.~~
2. ~~A file should be merged into the list box only after the Merge File menu item is used to select the File. No file should be loaded automatically when the application begins. Show the open file dialog and use a filter to display only text (\*.txt) files. After merging a file or manually inserting a new phone number, all menu items should be enabled.~~ **~~Portion~~** ~~of code needed below.~~
   * ~~Set the filter for the open file dialog this.dlgOpenFile.Filter = DataAccess.FileFilter;~~
   * ~~Get the name of the input file~~

~~if (this.dlgOpenFile.ShowDialog() == DialogResult.OK)~~

~~fileName = this.dlgOpenFile.FileName;~~

~~else~~

~~{~~

~~Program.errorFlag = true;~~

~~this.Close();~~

}

1. ~~For the Remove menu item, you should allow the user to change his/her mind before actually deleting the data. Display a message box to ask, "Are you sure you want to delete ...?" and provide Yes and No buttons.~~
2. ~~Add input file and save file dialogs to choose the filename. Allow text files (\*.txt) only.~~
3. ~~Reading a file merges with the names already there. Just add the new names to the list box~~.
4. ~~For Exit, if the data has changed since the last save, ask, "Do you want to save your data before quitting?" This dialog should show buttons for Yes, No, and Cancel. Do something reasonable for each alternative. Do the same thing if the form is closed using the close button in the form's upper-right corner.~~
5. ~~The output file name is not necessarily the same as the input filename. PhoneApp should be modified so that the output filename is passed as an argument to WritePhones.~~
6. ~~When an item in the list box is selected, display the phone number in the text box using PhoneNumberDisplay.~~

General

1. ~~Make sure to set AcceptButton and CancelButton properties for every form.~~
2. ~~Make sure to provide keyboard accelerators for every button and text box label.~~
3. ~~Make sure to provide keyboard accelerators for menus.~~
4. ~~Use Location and StartPosition properties to center the phone add, phone edit, and exception dialogs on Form1 parent. Place Form1 at 100,100 of the screen coordinates.~~
5. ~~Be sure tab order is set so that keyboard only navigation works.~~
6. ~~Be sure all menus are grayed out if their action does not apply, i.e., Save As when the phone list is empty, remove when the phone list is empty, etc.~~

Code Documentation

Each of your source files should contain a documentation header with description, author name, class information (CS3160, Spring/Fall 20xx). Each function should have a documentation header with a minimally function name, description, parameters, return value. Use proper program style at all times, which means indentation, alignment, and whitespace. Utilize self-documenting code which means use mnemonic, i.e., meaningful variable/function/namespace/class names, etc.

What to turn in?

Submit a compressed (.zip) solution folder via Canvas.

Ensure that all necessary files to compile/link/execute your projects are provided in your solution folder.

Here are some files that may be required:

1. All required C# source files (.cs)
2. makefile (if the solution requires it)
3. The entire Visual Studio 2019 solution folder (if the solution requires it)