Level 2 Creating an event driven computer program using C# (7540-008)

Author: Bogdan Pasterak 06/03/19

Technical Documentation

The program is included in one “frmEstimate” class inheriting from “Form”. The class is divided into parts. My code is in the part containing the constructor.

**The class has six private variables:**

* exchangeRate: decimal - stores the exchange rate value from GBP to Euro. I limited the range from 0.5 to 3 and the precision to four decimal places.
* priceCurrentMaterial: decimal - includes the price per square meter of currently selected material. This value has acceptable values ​​from 1 to 1000 and the precision to two decimal places.
* materialName: string - stores the name of the currently selected material. Acceptable values ​​are Brick, Concrete, Tarmac and Gravel.
* foundations: decimal - here is one of the two values ​​used to calculate the final price of the service. 1 means 100% for standard foundations. 1.25 to 125% of the price, extra 25% for extra deep foundations.
* foundationName: string - stores the type of foundations currently selected using radioButton. . Acceptable values ​​are Standard or Extra Deep.
* pricesOfMaterials: decimal array - stores four prices per square meter for four types of materials. These values ​​like priceCurrentMaterial have acceptable values ​​from 1 to 1000, and accuracy to two decimal places.

**The methods in the program are listed in the order that appears in the code.**

* frmEstimate - constructor without parameters. Automatically included with the called InitializeComponent method. At this point, pricesOfMaterials array is initialized. In the comment there is a few lines of code for quick start. Fills the required fields with the values ​​given in the documentation.
* exitToolStripMenuItem\_Click - Event dispatched after clicking Exit on the File menu. It calls the MessageBox with two Yes and No buttons. If the user clicks Yes, the program ends.
* btnStart\_Click - event dispatched after clicking Start button. If all the prices of materials and the exchange rate are filled and appropriate, the change of the application's appearance changes. From the initial settings it goes to the operating mode. The panels pnlExchangeRate, pnlPrice and btnStart button are hidden. Shown are now hidden grpFoundations, pnlCalculate, btnCalculate and btnClear. The starting values ​​of variables and exchange rate are set in txtGBR and txtEuro. The focus is passed to txtLength.
* setCurrentPrice - auxiliary method. Sets the priceCurrentMaterial and materialName depending on which radioButton is selected in the material group. Uses the pricesOfMaterials array.
* validateNumericTextBox - auxiliary method. Check the correctness of data in four steps.

Parameters are:

* 1. textBox: TextBox – has a string that should be validated.
  2. value: decimal – the value is returned if the TextBox content matches the restrictions, otherwise 0.
  3. property: string – optional, describes the field in TextBox, serves in messages. The default value is Price.
  4. topRange: decimal – optional, sets the upper allowable value for this field. The default value is 1000.
  5. bottomRange decimal – optional, sets the lower allowable value for this field. The default value is 1.
  6. precision: int – optional, limit the number of decimal places. The default value is 2.

Steps:

1. Checking if the field is empty.
2. Checking whether the string can be converted to a numeric value.
3. Checking if the value is within limits.
4. Checking whether the number of decimal places does not exceed the admissible value. If so, correct the value, select it.

* reminderMessageBox - auxiliary method. It calls the MessageBox with the message passed. Sets the title to Reminder, OK button and exclamation icon. Then clear the field in TextBox and set the focus there.
* txtXXXXXXX\_Leave - seven similar methods responsible for the Focus Leave event. Depending on the selected TextBox, it validates the data with the appropriate parameters.
* rbMaterial\_CheckedChanged - Event called after clicking one of four radioButton to choose the material. Runs procedure setCurrentPrice.
* rbFoundations\_CheckedChanged - Event called after choosing one of the two foundation options. Sets the values ​​of foundations and foundationName variables.
* btnCalculate\_Click - event dispatched after clicking Calculate button. If the validateArea method returns, the truth calculates the service price in GBR, Euro and displays the corresponding message in lblOutput.
* validateArea - auxiliary method. Checks the values ​​of the Length and Width fields. If appropriate, calculates surfaces in square meters. Otherwise, it returns false and 0.
* btnClear\_Click - event dispatched after clicking Clear button. Clears or sets the appropriate fields. Restores the application's start-up status as after pressing the Start button.