# 650Лабораторная работа №9

## Создание элементов управления

### Вариант 1

Цель работы: приобретение практических навыков создания визуальных элементов интерфейса приложения Windows Forms.

### Задание к работе

Создать компонент, служащий для ввода в текстовое поле вещественных чисел в заданном диапазоне. Компонент должен содержать следующие свойства: диапазон допустимых значений, текущее значение, целая часть текущего значения, дробная часть текущего значения, строковое представление текущего значения, шаг изменения при нажатии клавиш вверх и вниз.

# Решение

Для реализации компонента выбран способ создания составного элемента управления, поскольку в задании сказано, что работать он должен только с элементом класса TextBox. Компонентом перехватываются следующие события:

1. PreviewKeyDown — для регистрации нажатия клавиш «Стрелочка вверх» и «Стрелочка вниз» и последующего изменения текстового поля.
2. KeyPress — для фильтрации ввода.

## Текст программы

|  |  |
| --- | --- |
| 001 | **using** System; |
| 002 | **using** System.Collections.Generic; |
| 003 | **using** System.ComponentModel; |
| 004 | **using** System.Drawing; |
| 005 | **using** System.Data; |
| 006 | **using** System.Linq; |
| 007 | **using** System.Text; |
| 008 | **using** System.Threading.Tasks; |
| 009 | **using** System.Windows.Forms; |
| 010 |  |
| 011 | **namespace** ctrlFloatNumbersLib |
| 012 | { |
| 013 | **public** partial class ctrlFloatNumbers: UserControl |
| 014 | { |
| 015 | **public** ctrlFloatNumbers() |
| 016 | { |
| 017 | InitializeComponent(); |
| 018 | textBox.Text = "0"; |
| 019 | StepUp = 1; |
| 020 | StepDown = 1; |
| 021 | MinValue = **double**.MinValue; |
| 022 | MaxValue = **double**.MaxValue; |
| 023 | Checker(); |
| 024 | } |
| 025 |  |
| 026 | **public** **double** MinValue { **get**; **set**; } |
| 027 | **public** **double** MaxValue { **get**; **set**; } |
| 028 | **public** **double** StepUp { **get**; **set**; } |
| 029 | **public** **double** StepDown { **get**; **set**; } |
| 030 |  |
| 031 | **public** **double** Value |
| 032 | { |
| 033 | **get** |
| 034 | { |
| 035 | **return** Convert.ToDouble(textBox.Text); |
| 036 | } |
| 037 | **set** |
| 038 | { |
| 039 | textBox.Text = value.ToString(); |
| 040 | textBox.SelectionStart = textBox.Text.Length; |
| 041 | } |
| 042 | } |
| 043 |  |
| 044 | **public** **int** Integer |
| 045 | { |
| 046 | **get** |
| 047 | { |
| 048 | **return** Convert.ToInt32(Math.Truncate(Value)); |
| 049 | } |
| 050 | **set** |
| 051 | { |
| 052 | Value = value + Fractional; |
| 053 | } |
| 054 | } |
| 055 |  |
| 056 | **public** **string** String |
| 057 | { |
| 058 | **get** |
| 059 | { |
| 060 | **return** textBox.Text; |
| 061 | } |
| 062 | **set** |
| 063 | { |
| 064 | textBox.Text = value; |
| 065 | } |
| 066 | } |
| 067 |  |
| 068 | **public** **double** Fractional |
| 069 | { |
| 070 | **get** |
| 071 | { |
| 072 | **return** Value - Integer; |
| 073 | } |
| 074 | **set** |
| 075 | { |
| 076 | Value = value + Integer; |
| 077 | } |
| 078 | } |
| 079 |  |
| 080 | **private** **void** textBox\_PreviewKeyDown(**object** sender,  PreviewKeyDownEventArgs e) |
| 081 | { |
| 082 | **if** ((e.KeyCode == Keys.Up) && ((Value + StepUp) <= MaxValue)) |
| 083 | { |
| 084 | Value += StepUp; |
| 085 | } |
| 086 |  |
| 087 | **if** ((e.KeyCode == Keys.Down) &&   ((Value - StepDown) >= MinValue)) |
| 088 | { |
| 089 | Value -= StepDown; |
| 090 | } |
| 091 | } |
| 092 |  |
| 093 | **private** **void** textBox\_KeyPress(**object** sender, KeyPressEventArgs e) |
| 094 | { |
| 095 | **if** (((e.KeyChar == '.' && !textBox.Text.Contains('.') &&  textBox.Text.Length >= 1) || Char.IsDigit(e.KeyChar)) ||   e.KeyChar == 8) |
| 096 | { |
| 097 | **base**.OnKeyPress(e); |
| 098 | Checker(); |
| 099 | } |
| 100 | **else** |
| 101 | { |
| 102 | e.Handled = **true**; |
| 103 | } |
| 104 | } |
| 105 |  |
| 106 | **private** **void** Checker() |
| 107 | { |
| 108 | **if** (Value > MaxValue) |
| 109 | { |
| 110 | Value = MaxValue; |
| 111 | } |
| 112 |  |
| 113 | **if** (Value < MinValue) |
| 114 | { |
| 115 | Value = MinValue; |
| 116 | } |
| 117 | } |
| 118 | } |
| 119 | } |