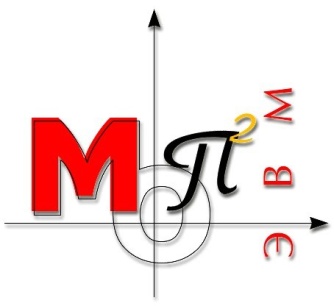
Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждения высшего образования   
«Южный федеральный университет»

Инженерно-технологическая академия

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

**

Лабораторная работа №5

по дисциплине

"ООП"

на тему

". Классы, объекты,

наследование в С#"

**Выполнил:**

студент группы КТбо2-6

Пустовой А.В

**Проверил:**

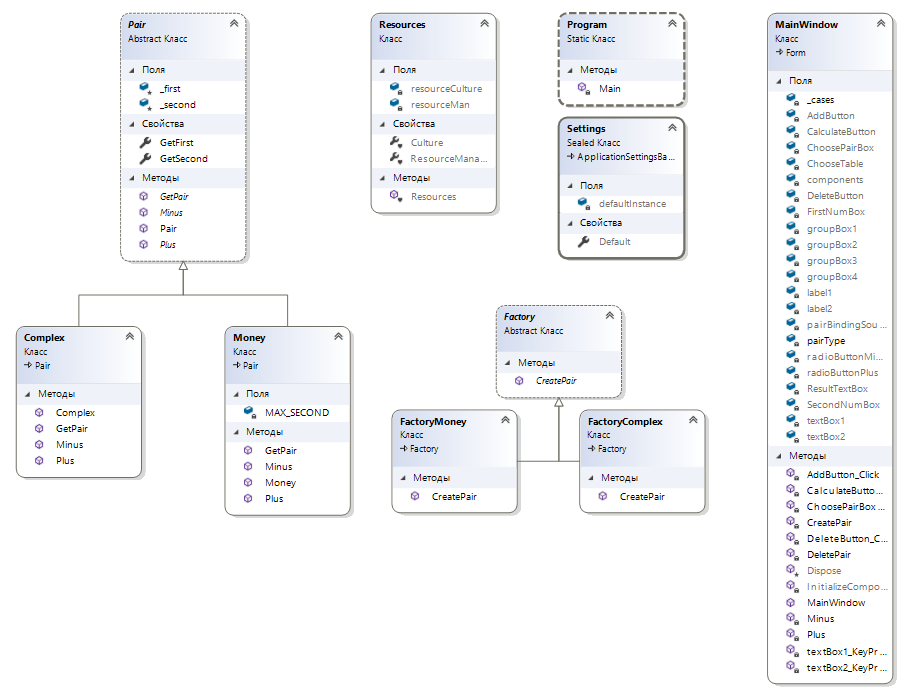
Таганрог, 2020

**Вариант**

**Задание:**

Выполните одну из написанных ранее работ на языке C#. В качестве задания следует выбрать уже готовый вариант на С++.

**Диаграмма классов**



**Листинг программы**

**Complex.cs**

|  |
| --- |
| using System; |
|  | using System.Collections.Generic; |
|  | using System.Text; |
|  |  |
|  | namespace lab6 |
|  | { |
|  | class Complex : Pair |
|  | { |
|  |  |
|  | public Complex(int first, int second) : base(first, second) { } |
|  |  |
|  | public override string GetPair() |
|  | { |
|  | return \_first + " + " + \_second + "i"; |
|  | } |
|  |  |
|  | public override Pair Minus(Pair source) |
|  | { |
|  | int resFirst = \_first; |
|  | int resSecond = \_second; |
|  |  |
|  | resFirst -= source.GetFirst; |
|  | resSecond -= source.GetSecond; |
|  | return new Complex(resFirst, resSecond); |
|  | } |
|  |  |
|  | public override Pair Plus(Pair source) |
|  | { |
|  | int resFirst = \_first; |
|  | int resSecond = \_second; |
|  |  |
|  | resFirst += source.GetFirst; |
|  | resSecond += source.GetSecond; |
|  | return new Complex(resFirst, resSecond); |
|  | } |
|  | } |
|  | } |

**ComplexInterator.cs**

|  |
| --- |
| using System; |
|  |  |
|  | namespace lab6 |
|  | { |
|  | class ConsoleInteractor |
|  | { |
|  | public ConsoleInteractor() |
|  | { |
|  | Console.Write("Enter space for Pairs\n>> "); |
|  | \_count = Convert.ToInt32(Console.ReadLine()); |
|  | \_names = new string[\_count]; |
|  | \_cases = new Pair[\_count]; |
|  | } |
|  |  |
|  | public void Init() |
|  | { |
|  | int code, arg; string name, name2; |
|  | bool flag = true; |
|  | while (flag) |
|  | { |
|  | Console.Write("Enter \"3\" - Help\n"); |
|  | Console.Write(">> "); |
|  | code = Convert.ToInt32(Console.ReadLine()); |
|  |  |
|  | switch (code) |
|  | { |
|  | case 1: |
|  | CreatePair(); |
|  | break; |
|  | case 2: |
|  | DelPair(); |
|  | break; |
|  | case 3: |
|  | Help(); |
|  | break; |
|  | case 4: |
|  | Plus(); |
|  | break; |
|  | case 5: |
|  | Minus(); |
|  | break; |
|  | case 6: |
|  | PrintPair(); |
|  | break; |
|  | case 7: |
|  | flag = false; |
|  | break; |
|  | } |
|  | } |
|  | } |
|  |  |
|  |  |
|  | private int GetIndex(string name) |
|  | { |
|  | for (int i = 0; i < \_count; i++) |
|  | { |
|  | if (name == \_names[i]) |
|  | { |
|  | return i; |
|  | } |
|  | } |
|  | return -1; |
|  | } |
|  |  |
|  | private void Minus() |
|  | { |
|  | string name, name2; |
|  | Console.Write("Enter name of first pair\n>> "); |
|  | name = Console.ReadLine(); |
|  | Console.Write("Enter name of second pair\n>> "); |
|  | name2 = Console.ReadLine(); |
|  |  |
|  | int indexName1 = GetIndex(name); |
|  | int indexName2 = GetIndex(name2); |
|  |  |
|  | if (indexName1 != -1 && indexName2 != -1) |
|  | { |
|  | Console.Write(\_cases[indexName1].GetPair() + "\n-\n"); |
|  | Console.Write(\_cases[indexName2].GetPair() + "\n"); |
|  | Console.Write(\_cases[indexName1].Minus(\_cases[indexName2]).GetPair() + "\n"); |
|  | } |
|  | else |
|  | { |
|  | Console.Write("Wrond name\n"); |
|  | } |
|  | } |
|  |  |
|  | private void Plus() |
|  | { |
|  | string name, name2; |
|  | Console.Write("Enter name of first pair\n>> "); |
|  | name = Console.ReadLine(); |
|  | Console.Write("Enter name of second pair\n>> "); |
|  | name2 = Console.ReadLine(); |
|  |  |
|  | int indexName1 = GetIndex(name); |
|  | int indexName2 = GetIndex(name2); |
|  |  |
|  | if (indexName1 != -1 && indexName2 != -1) |
|  | { |
|  | Console.Write(\_cases[indexName1].GetPair() + "\n+\n"); |
|  | Console.Write(\_cases[indexName2].GetPair() + "\n"); |
|  | Console.Write(\_cases[indexName1].Plus(\_cases[indexName2]).GetPair() + "\n"); |
|  | } |
|  | else |
|  | { |
|  | Console.Write("Wrond name\n"); |
|  | } |
|  | } |
|  |  |
|  | private void Help() |
|  | { |
|  | Console.Write("\"1\" - Create new pair\n\"2\" - Delete pair\n\"3\" - Help\n\"4\" - Plus\n\"5\" - Minus\n\"6\" - Print pair\n\"7\" - Exit\n"); |
|  | } |
|  |  |
|  | private void CreatePair() |
|  | { |
|  | int firstNum, secondNum; |
|  | Console.Write("Enter first num:\n>> "); |
|  | firstNum = Convert.ToInt32(Console.ReadLine()); |
|  |  |
|  | Console.Write("Enter second num:\n>> "); |
|  | secondNum = Convert.ToInt32(Console.ReadLine()); |
|  |  |
|  | Console.Write("Enter name of Pair:\n>> "); |
|  | string name; |
|  | name = Console.ReadLine(); |
|  |  |
|  | int type; |
|  | Console.Write("Is it Complex or Money (1 or 2):\n>> "); |
|  | type = Convert.ToInt32(Console.ReadLine()); |
|  |  |
|  | bool isEnoughtSpace = false; |
|  | int index = -1; |
|  | for (int i = 0; i < \_count && !isEnoughtSpace; i++) |
|  | { |
|  | if (\_names[i] == null) |
|  | { |
|  | isEnoughtSpace = true; |
|  | index = i; |
|  | } |
|  | } |
|  |  |
|  | if (isEnoughtSpace) |
|  | { |
|  | FactoryComplex factoryComplex = new FactoryComplex(); |
|  | FactoryMoney factoryMoney = new FactoryMoney(); |
|  | Pair newElement = null; |
|  | if (type == 1) |
|  | { |
|  | newElement = factoryComplex.CreatePair(firstNum, secondNum); |
|  | } |
|  | else if (type == 2) |
|  | { |
|  | try |
|  | { |
|  | newElement = factoryMoney.CreatePair(firstNum, secondNum); |
|  | } |
|  | catch (ArgumentOutOfRangeException err) |
|  | { |
|  | Console.Write(err.Message); |
|  | return; |
|  | } |
|  | } |
|  |  |
|  | \_names[index] = name; |
|  | \_cases[index] = newElement; |
|  | } |
|  | else |
|  | { |
|  | Console.Write("You don't have enought space, delete something\n"); |
|  | } |
|  | } |
|  |  |
|  | private void DeletePair(string name) |
|  | { |
|  | int index = GetIndex(name); |
|  |  |
|  | if (index == -1) |
|  | { |
|  | Console.Write("Can't find pair with this name\n"); |
|  | } |
|  | else |
|  | { |
|  | \_names[index] = null; |
|  | \_cases[index] = null; |
|  | } |
|  | } |
|  |  |
|  | private void PrintPair() |
|  | { |
|  | string name; |
|  | Console.Write("Enter name of pair to print\n>> "); |
|  | name = Console.ReadLine(); |
|  | int index = GetIndex(name); |
|  | if (index != -1) |
|  | { |
|  | Console.Write(\_cases[index].GetPair() + "\n"); |
|  | } |
|  | else |
|  | { |
|  | Console.Write("Wrong name\n"); |
|  | } |
|  | } |
|  |  |
|  | private void DelPair() |
|  | { |
|  | string name; |
|  | Console.Write("Enter name of pair to delete\n>> "); |
|  | name = Console.ReadLine(); |
|  | DeletePair(name); |
|  | Console.Write("Complete\n"); |
|  |  |
|  | } |
|  |  |
|  | private Pair[] \_cases = null; |
|  | private string[] \_names = null; |
|  | private int \_count = 0; |
|  | } |
|  | } |

**Factory.cs**

|  |
| --- |
| using System; |
|  |  |
|  | namespace lab6 |
|  | { |
|  | abstract class Factory |
|  | { |
|  | public abstract Pair CreatePair(int first, int second); |
|  | } |
|  |  |
|  | class FactoryMoney : Factory |
|  | { |
|  | public override Pair CreatePair(int first, int second) |
|  | { |
|  | return new Money(first, second); |
|  | } |
|  | } |
|  |  |
|  | class FactoryComplex : Factory |
|  | { |
|  | public override Pair CreatePair(int first, int second) |
|  | { |
|  | return new Complex(first, second); |
|  | } |
|  | } |
|  | } |

**MainWindow.Designer.cs**

|  |
| --- |
| namespace lab6 |
|  | { |
|  | partial class MainWindow |
|  | { |
|  | /// <summary> |
|  | /// Required designer variable. |
|  | /// </summary> |
|  | private System.ComponentModel.IContainer components = null; |
|  |  |
|  | /// <summary> |
|  | /// Clean up any resources being used. |
|  | /// </summary> |
|  | /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param> |
|  | protected override void Dispose(bool disposing) |
|  | { |
|  | if (disposing && (components != null)) |
|  | { |
|  | components.Dispose(); |
|  | } |
|  | base.Dispose(disposing); |
|  | } |
|  |  |
|  | #region Windows Form Designer generated code |
|  |  |
|  | /// <summary> |
|  | /// Required method for Designer support - do not modify |
|  | /// the contents of this method with the code editor. |
|  | /// </summary> |
|  | private void InitializeComponent() |
|  | { |
|  | this.components = new System.ComponentModel.Container(); |
|  | this.groupBox1 = new System.Windows.Forms.GroupBox(); |
|  | this.textBox2 = new System.Windows.Forms.TextBox(); |
|  | this.textBox1 = new System.Windows.Forms.TextBox(); |
|  | this.AddButton = new System.Windows.Forms.Button(); |
|  | this.ChoosePairBox = new System.Windows.Forms.ComboBox(); |
|  | this.groupBox2 = new System.Windows.Forms.GroupBox(); |
|  | this.ChooseTable = new System.Windows.Forms.ListBox(); |
|  | this.DeleteButton = new System.Windows.Forms.Button(); |
|  | this.groupBox3 = new System.Windows.Forms.GroupBox(); |
|  | this.CalculateButton = new System.Windows.Forms.Button(); |
|  | this.radioButtonMinus = new System.Windows.Forms.RadioButton(); |
|  | this.radioButtonPlus = new System.Windows.Forms.RadioButton(); |
|  | this.label2 = new System.Windows.Forms.Label(); |
|  | this.label1 = new System.Windows.Forms.Label(); |
|  | this.SecondNumBox = new System.Windows.Forms.ComboBox(); |
|  | this.FirstNumBox = new System.Windows.Forms.ComboBox(); |
|  | this.groupBox4 = new System.Windows.Forms.GroupBox(); |
|  | this.ResultTextBox = new System.Windows.Forms.RichTextBox(); |
|  | this.pairBindingSource = new System.Windows.Forms.BindingSource(this.components); |
|  | this.groupBox1.SuspendLayout(); |
|  | this.groupBox2.SuspendLayout(); |
|  | this.groupBox3.SuspendLayout(); |
|  | this.groupBox4.SuspendLayout(); |
|  | ((System.ComponentModel.ISupportInitialize)(this.pairBindingSource)).BeginInit(); |
|  | this.SuspendLayout(); |
|  | // |
|  | // groupBox1 |
|  | // |
|  | this.groupBox1.Controls.Add(this.textBox2); |
|  | this.groupBox1.Controls.Add(this.textBox1); |
|  | this.groupBox1.Controls.Add(this.AddButton); |
|  | this.groupBox1.Controls.Add(this.ChoosePairBox); |
|  | this.groupBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.9F); |
|  | this.groupBox1.Location = new System.Drawing.Point(12, 12); |
|  | this.groupBox1.Name = "groupBox1"; |
|  | this.groupBox1.RightToLeft = System.Windows.Forms.RightToLeft.No; |
|  | this.groupBox1.Size = new System.Drawing.Size(218, 133); |
|  | this.groupBox1.TabIndex = 0; |
|  | this.groupBox1.TabStop = false; |
|  | this.groupBox1.Text = "Choose pair type"; |
|  | // |
|  | // textBox2 |
|  | // |
|  | this.textBox2.Location = new System.Drawing.Point(112, 59); |
|  | this.textBox2.Name = "textBox2"; |
|  | this.textBox2.Size = new System.Drawing.Size(100, 26); |
|  | this.textBox2.TabIndex = 3; |
|  | this.textBox2.KeyPress += new System.Windows.Forms.KeyPressEventHandler(this.textBox2\_KeyPress); |
|  | // |
|  | // textBox1 |
|  | // |
|  | this.textBox1.Location = new System.Drawing.Point(5, 59); |
|  | this.textBox1.MaxLength = 10; |
|  | this.textBox1.Name = "textBox1"; |
|  | this.textBox1.Size = new System.Drawing.Size(100, 26); |
|  | this.textBox1.TabIndex = 2; |
|  | this.textBox1.KeyPress += new System.Windows.Forms.KeyPressEventHandler(this.textBox1\_KeyPress); |
|  | // |
|  | // AddButton |
|  | // |
|  | this.AddButton.Location = new System.Drawing.Point(63, 91); |
|  | this.AddButton.Name = "AddButton"; |
|  | this.AddButton.Size = new System.Drawing.Size(88, 35); |
|  | this.AddButton.TabIndex = 1; |
|  | this.AddButton.Text = "Add"; |
|  | this.AddButton.UseVisualStyleBackColor = true; |
|  | this.AddButton.Click += new System.EventHandler(this.AddButton\_Click); |
|  | // |
|  | // ChoosePairBox |
|  | // |
|  | this.ChoosePairBox.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList; |
|  | this.ChoosePairBox.Items.AddRange(new object[] { |
|  | "Money", |
|  | "Complex number"}); |
|  | this.ChoosePairBox.Location = new System.Drawing.Point(5, 25); |
|  | this.ChoosePairBox.Name = "ChoosePairBox"; |
|  | this.ChoosePairBox.Size = new System.Drawing.Size(207, 28); |
|  | this.ChoosePairBox.TabIndex = 0; |
|  | this.ChoosePairBox.SelectedIndexChanged += new System.EventHandler(this.ChoosePairBox\_SelectedIndexChanged); |
|  | // |
|  | // groupBox2 |
|  | // |
|  | this.groupBox2.Controls.Add(this.ChooseTable); |
|  | this.groupBox2.Controls.Add(this.DeleteButton); |
|  | this.groupBox2.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.8F); |
|  | this.groupBox2.Location = new System.Drawing.Point(12, 161); |
|  | this.groupBox2.Name = "groupBox2"; |
|  | this.groupBox2.Size = new System.Drawing.Size(218, 282); |
|  | this.groupBox2.TabIndex = 1; |
|  | this.groupBox2.TabStop = false; |
|  | this.groupBox2.Text = "Pairs"; |
|  | // |
|  | // ChooseTable |
|  | // |
|  | this.ChooseTable.FormattingEnabled = true; |
|  | this.ChooseTable.ItemHeight = 20; |
|  | this.ChooseTable.Location = new System.Drawing.Point(6, 25); |
|  | this.ChooseTable.Name = "ChooseTable"; |
|  | this.ChooseTable.Size = new System.Drawing.Size(207, 204); |
|  | this.ChooseTable.TabIndex = 2; |
|  | // |
|  | // DeleteButton |
|  | // |
|  | this.DeleteButton.Location = new System.Drawing.Point(6, 238); |
|  | this.DeleteButton.Name = "DeleteButton"; |
|  | this.DeleteButton.Size = new System.Drawing.Size(207, 38); |
|  | this.DeleteButton.TabIndex = 0; |
|  | this.DeleteButton.Text = "Delete"; |
|  | this.DeleteButton.UseVisualStyleBackColor = true; |
|  | this.DeleteButton.Click += new System.EventHandler(this.DeleteButton\_Click); |
|  | // |
|  | // groupBox3 |
|  | // |
|  | this.groupBox3.Controls.Add(this.CalculateButton); |
|  | this.groupBox3.Controls.Add(this.radioButtonMinus); |
|  | this.groupBox3.Controls.Add(this.radioButtonPlus); |
|  | this.groupBox3.Controls.Add(this.label2); |
|  | this.groupBox3.Controls.Add(this.label1); |
|  | this.groupBox3.Controls.Add(this.SecondNumBox); |
|  | this.groupBox3.Controls.Add(this.FirstNumBox); |
|  | this.groupBox3.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.8F); |
|  | this.groupBox3.Location = new System.Drawing.Point(262, 12); |
|  | this.groupBox3.Name = "groupBox3"; |
|  | this.groupBox3.Size = new System.Drawing.Size(350, 254); |
|  | this.groupBox3.TabIndex = 2; |
|  | this.groupBox3.TabStop = false; |
|  | this.groupBox3.Text = "Operations"; |
|  | // |
|  | // CalculateButton |
|  | // |
|  | this.CalculateButton.Location = new System.Drawing.Point(6, 192); |
|  | this.CalculateButton.Name = "CalculateButton"; |
|  | this.CalculateButton.Size = new System.Drawing.Size(338, 54); |
|  | this.CalculateButton.TabIndex = 6; |
|  | this.CalculateButton.Text = "Calculate"; |
|  | this.CalculateButton.UseVisualStyleBackColor = true; |
|  | this.CalculateButton.Click += new System.EventHandler(this.CalculateButton\_Click); |
|  | // |
|  | // radioButtonMinus |
|  | // |
|  | this.radioButtonMinus.AutoSize = true; |
|  | this.radioButtonMinus.Location = new System.Drawing.Point(11, 65); |
|  | this.radioButtonMinus.Name = "radioButtonMinus"; |
|  | this.radioButtonMinus.Size = new System.Drawing.Size(75, 24); |
|  | this.radioButtonMinus.TabIndex = 5; |
|  | this.radioButtonMinus.TabStop = true; |
|  | this.radioButtonMinus.Text = "Minus"; |
|  | this.radioButtonMinus.UseVisualStyleBackColor = true; |
|  | // |
|  | // radioButtonPlus |
|  | // |
|  | this.radioButtonPlus.AutoSize = true; |
|  | this.radioButtonPlus.Location = new System.Drawing.Point(11, 35); |
|  | this.radioButtonPlus.Name = "radioButtonPlus"; |
|  | this.radioButtonPlus.Size = new System.Drawing.Size(63, 24); |
|  | this.radioButtonPlus.TabIndex = 4; |
|  | this.radioButtonPlus.TabStop = true; |
|  | this.radioButtonPlus.Text = "Plus"; |
|  | this.radioButtonPlus.UseVisualStyleBackColor = true; |
|  | // |
|  | // label2 |
|  | // |
|  | this.label2.AutoSize = true; |
|  | this.label2.Location = new System.Drawing.Point(217, 126); |
|  | this.label2.Name = "label2"; |
|  | this.label2.Size = new System.Drawing.Size(102, 20); |
|  | this.label2.TabIndex = 3; |
|  | this.label2.Text = "Second num"; |
|  | // |
|  | // label1 |
|  | // |
|  | this.label1.AutoSize = true; |
|  | this.label1.Location = new System.Drawing.Point(7, 126); |
|  | this.label1.Name = "label1"; |
|  | this.label1.Size = new System.Drawing.Size(104, 20); |
|  | this.label1.TabIndex = 2; |
|  | this.label1.Text = "First number"; |
|  | // |
|  | // SecondNumBox |
|  | // |
|  | this.SecondNumBox.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList; |
|  | this.SecondNumBox.FormattingEnabled = true; |
|  | this.SecondNumBox.Location = new System.Drawing.Point(220, 149); |
|  | this.SecondNumBox.Name = "SecondNumBox"; |
|  | this.SecondNumBox.Size = new System.Drawing.Size(124, 28); |
|  | this.SecondNumBox.TabIndex = 1; |
|  | // |
|  | // FirstNumBox |
|  | // |
|  | this.FirstNumBox.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList; |
|  | this.FirstNumBox.FormattingEnabled = true; |
|  | this.FirstNumBox.Location = new System.Drawing.Point(6, 149); |
|  | this.FirstNumBox.Name = "FirstNumBox"; |
|  | this.FirstNumBox.Size = new System.Drawing.Size(121, 28); |
|  | this.FirstNumBox.TabIndex = 0; |
|  | // |
|  | // groupBox4 |
|  | // |
|  | this.groupBox4.Controls.Add(this.ResultTextBox); |
|  | this.groupBox4.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.8F); |
|  | this.groupBox4.Location = new System.Drawing.Point(262, 272); |
|  | this.groupBox4.Name = "groupBox4"; |
|  | this.groupBox4.Size = new System.Drawing.Size(350, 165); |
|  | this.groupBox4.TabIndex = 7; |
|  | this.groupBox4.TabStop = false; |
|  | this.groupBox4.Text = "Result"; |
|  | // |
|  | // ResultTextBox |
|  | // |
|  | this.ResultTextBox.AccessibleName = "ResultTextBox"; |
|  | this.ResultTextBox.Cursor = System.Windows.Forms.Cursors.IBeam; |
|  | this.ResultTextBox.Location = new System.Drawing.Point(3, 22); |
|  | this.ResultTextBox.Name = "ResultTextBox"; |
|  | this.ResultTextBox.ReadOnly = true; |
|  | this.ResultTextBox.RightToLeft = System.Windows.Forms.RightToLeft.No; |
|  | this.ResultTextBox.Size = new System.Drawing.Size(341, 137); |
|  | this.ResultTextBox.TabIndex = 0; |
|  | this.ResultTextBox.Text = ""; |
|  | // |
|  | // pairBindingSource |
|  | // |
|  | this.pairBindingSource.DataSource = typeof(lab6.Pair); |
|  | // |
|  | // MainWindow |
|  | // |
|  | this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F); |
|  | this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font; |
|  | this.ClientSize = new System.Drawing.Size(627, 454); |
|  | this.Controls.Add(this.groupBox4); |
|  | this.Controls.Add(this.groupBox3); |
|  | this.Controls.Add(this.groupBox2); |
|  | this.Controls.Add(this.groupBox1); |
|  | this.Name = "MainWindow"; |
|  | this.Text = "Laba5-6"; |
|  | this.groupBox1.ResumeLayout(false); |
|  | this.groupBox1.PerformLayout(); |
|  | this.groupBox2.ResumeLayout(false); |
|  | this.groupBox3.ResumeLayout(false); |
|  | this.groupBox3.PerformLayout(); |
|  | this.groupBox4.ResumeLayout(false); |
|  | ((System.ComponentModel.ISupportInitialize)(this.pairBindingSource)).EndInit(); |
|  | this.ResumeLayout(false); |
|  |  |
|  | } |
|  |  |
|  | #endregion |
|  | private System.Windows.Forms.BindingSource pairBindingSource; |
|  | private System.Windows.Forms.GroupBox groupBox1; |
|  | private System.Windows.Forms.ComboBox ChoosePairBox; |
|  | private System.Windows.Forms.Button AddButton; |
|  | private System.Windows.Forms.GroupBox groupBox2; |
|  | private System.Windows.Forms.Button DeleteButton; |
|  | private System.Windows.Forms.ListBox ChooseTable; |
|  | private System.Windows.Forms.GroupBox groupBox3; |
|  | private System.Windows.Forms.Label label2; |
|  | private System.Windows.Forms.Label label1; |
|  | private System.Windows.Forms.ComboBox SecondNumBox; |
|  | private System.Windows.Forms.ComboBox FirstNumBox; |
|  | private System.Windows.Forms.RadioButton radioButtonMinus; |
|  | private System.Windows.Forms.RadioButton radioButtonPlus; |
|  | private System.Windows.Forms.Button CalculateButton; |
|  | private System.Windows.Forms.GroupBox groupBox4; |
|  | private System.Windows.Forms.RichTextBox ResultTextBox; |
|  | private System.Windows.Forms.TextBox textBox2; |
|  | private System.Windows.Forms.TextBox textBox1; |
|  | } |
|  | } |

**MainWindow.recx**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> |
|  | <root> |
|  | <!-- |
|  | Microsoft ResX Schema |
|  |  |
|  | Version 2.0 |
|  |  |
|  | The primary goals of this format is to allow a simple XML format |
|  | that is mostly human readable. The generation and parsing of the |
|  | various data types are done through the TypeConverter classes |
|  | associated with the data types. |
|  |  |
|  | Example: |
|  |  |
|  | ... ado.net/XML headers & schema ... |
|  | <resheader name="resmimetype">text/microsoft-resx</resheader> |
|  | <resheader name="version">2.0</resheader> |
|  | <resheader name="reader">System.Resources.ResXResourceReader, System.Windows.Forms, ...</resheader> |
|  | <resheader name="writer">System.Resources.ResXResourceWriter, System.Windows.Forms, ...</resheader> |
|  | <data name="Name1"><value>this is my long string</value><comment>this is a comment</comment></data> |
|  | <data name="Color1" type="System.Drawing.Color, System.Drawing">Blue</data> |
|  | <data name="Bitmap1" mimetype="application/x-microsoft.net.object.binary.base64"> |
|  | <value>[base64 mime encoded serialized .NET Framework object]</value> |
|  | </data> |
|  | <data name="Icon1" type="System.Drawing.Icon, System.Drawing" mimetype="application/x-microsoft.net.object.bytearray.base64"> |
|  | <value>[base64 mime encoded string representing a byte array form of the .NET Framework object]</value> |
|  | <comment>This is a comment</comment> |
|  | </data> |
|  |  |
|  | There are any number of "resheader" rows that contain simple |
|  | name/value pairs. |
|  |  |
|  | Each data row contains a name, and value. The row also contains a |
|  | type or mimetype. Type corresponds to a .NET class that support |
|  | text/value conversion through the TypeConverter architecture. |
|  | Classes that don't support this are serialized and stored with the |
|  | mimetype set. |
|  |  |
|  | The mimetype is used for serialized objects, and tells the |
|  | ResXResourceReader how to depersist the object. This is currently not |
|  | extensible. For a given mimetype the value must be set accordingly: |
|  |  |
|  | Note - application/x-microsoft.net.object.binary.base64 is the format |
|  | that the ResXResourceWriter will generate, however the reader can |
|  | read any of the formats listed below. |
|  |  |
|  | mimetype: application/x-microsoft.net.object.binary.base64 |
|  | value : The object must be serialized with |
|  | : System.Runtime.Serialization.Formatters.Binary.BinaryFormatter |
|  | : and then encoded with base64 encoding. |
|  |  |
|  | mimetype: application/x-microsoft.net.object.soap.base64 |
|  | value : The object must be serialized with |
|  | : System.Runtime.Serialization.Formatters.Soap.SoapFormatter |
|  | : and then encoded with base64 encoding. |
|  |  |
|  | mimetype: application/x-microsoft.net.object.bytearray.base64 |
|  | value : The object must be serialized into a byte array |
|  | : using a System.ComponentModel.TypeConverter |
|  | : and then encoded with base64 encoding. |
|  | --> |
|  | <xsd:schema id="root" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"> |
|  | <xsd:import namespace="http://www.w3.org/XML/1998/namespace" /> |
|  | <xsd:element name="root" msdata:IsDataSet="true"> |
|  | <xsd:complexType> |
|  | <xsd:choice maxOccurs="unbounded"> |
|  | <xsd:element name="metadata"> |
|  | <xsd:complexType> |
|  | <xsd:sequence> |
|  | <xsd:element name="value" type="xsd:string" minOccurs="0" /> |
|  | </xsd:sequence> |
|  | <xsd:attribute name="name" use="required" type="xsd:string" /> |
|  | <xsd:attribute name="type" type="xsd:string" /> |
|  | <xsd:attribute name="mimetype" type="xsd:string" /> |
|  | <xsd:attribute ref="xml:space" /> |
|  | </xsd:complexType> |
|  | </xsd:element> |
|  | <xsd:element name="assembly"> |
|  | <xsd:complexType> |
|  | <xsd:attribute name="alias" type="xsd:string" /> |
|  | <xsd:attribute name="name" type="xsd:string" /> |
|  | </xsd:complexType> |
|  | </xsd:element> |
|  | <xsd:element name="data"> |
|  | <xsd:complexType> |
|  | <xsd:sequence> |
|  | <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1" /> |
|  | <xsd:element name="comment" type="xsd:string" minOccurs="0" msdata:Ordinal="2" /> |
|  | </xsd:sequence> |
|  | <xsd:attribute name="name" type="xsd:string" use="required" msdata:Ordinal="1" /> |
|  | <xsd:attribute name="type" type="xsd:string" msdata:Ordinal="3" /> |
|  | <xsd:attribute name="mimetype" type="xsd:string" msdata:Ordinal="4" /> |
|  | <xsd:attribute ref="xml:space" /> |
|  | </xsd:complexType> |
|  | </xsd:element> |
|  | <xsd:element name="resheader"> |
|  | <xsd:complexType> |
|  | <xsd:sequence> |
|  | <xsd:element name="value" type="xsd:string" minOccurs="0" msdata:Ordinal="1" /> |
|  | </xsd:sequence> |
|  | <xsd:attribute name="name" type="xsd:string" use="required" /> |
|  | </xsd:complexType> |
|  | </xsd:element> |
|  | </xsd:choice> |
|  | </xsd:complexType> |
|  | </xsd:element> |
|  | </xsd:schema> |
|  | <resheader name="resmimetype"> |
|  | <value>text/microsoft-resx</value> |
|  | </resheader> |
|  | <resheader name="version"> |
|  | <value>2.0</value> |
|  | </resheader> |
|  | <resheader name="reader"> |
|  | <value>System.Resources.ResXResourceReader, System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value> |
|  | </resheader> |
|  | <resheader name="writer"> |
|  | <value>System.Resources.ResXResourceWriter, System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</value> |
|  | </resheader> |
|  | <metadata name="pairBindingSource.TrayLocation" type="System.Drawing.Point, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"> |
|  | <value>930, 930</value> |
|  | </metadata> |
|  | <metadata name="$this.TrayHeight" type="System.Int32, mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"> |
|  | <value>29</value> |
|  | </metadata> |
|  | </root> |

**Money.cs**

|  |
| --- |
| using System; |
|  |  |
|  | namespace lab6 |
|  | { |
|  | class Money : Pair |
|  | { |
|  |  |
|  | public Money(int first, int second) : base(first, second) |
|  | { |
|  | if (\_second > MAX\_SECOND) |
|  | { |
|  | \_first += \_second / MAX\_SECOND; |
|  | \_second = \_second % MAX\_SECOND; |
|  | } |
|  |  |
|  | if (\_first < 0 || \_second < 0) |
|  | { |
|  | throw new ArgumentOutOfRangeException("Less than zero\n"); |
|  | } |
|  | } |
|  |  |
|  | public override string GetPair() |
|  | { |
|  | string result = \_first + "." + \_second + " rub"; |
|  | return result; |
|  | } |
|  |  |
|  | public override Pair Minus(Pair source) |
|  | { |
|  | int firstInt = source.GetFirst; |
|  | int secondInt = source.GetSecond; |
|  |  |
|  | int resFirst = \_first; |
|  | int resSecond = \_second; |
|  |  |
|  | resFirst = (\_first > firstInt) ? \_first - firstInt : 0; |
|  |  |
|  | if (\_second >= secondInt) |
|  | { |
|  | resSecond = \_second - secondInt; |
|  | } |
|  | else if (\_first == 1) |
|  | { |
|  | resFirst = \_first - 1; |
|  | resSecond = \_second - secondInt + MAX\_SECOND; |
|  | } |
|  | else |
|  | { |
|  | resSecond = 0; |
|  | } |
|  |  |
|  | return new Money(resFirst, resSecond); |
|  | } |
|  |  |
|  | public override Pair Plus(Pair source) |
|  | { |
|  | int firstInt = source.GetFirst; |
|  | int secondInt = source.GetSecond; |
|  |  |
|  | int resFirst = \_first; |
|  | int resSecond = \_second; |
|  |  |
|  | resFirst += firstInt + (\_second + secondInt) / MAX\_SECOND; |
|  | resSecond = (\_second + secondInt) % MAX\_SECOND; |
|  |  |
|  | return new Money(resFirst, resSecond); |
|  | } |
|  |  |
|  | private readonly int MAX\_SECOND = 100; |
|  | } |
|  | } |

**Pair.cs**

|  |
| --- |
| using System; |
|  |  |
|  | namespace lab6 |
|  | { |
|  | abstract class Pair |
|  | { |
|  | protected int \_first = 0; |
|  | protected int \_second = 0; |
|  |  |
|  | public int GetFirst |
|  | { |
|  | get |
|  | { |
|  | return \_first; |
|  | } |
|  | } |
|  | public int GetSecond |
|  | { |
|  | get |
|  | { |
|  | return \_second; |
|  | } |
|  | } |
|  |  |
|  | public Pair(int first, int second) |
|  | { |
|  | \_first = first; |
|  | \_second = second; |
|  | } |
|  |  |
|  | abstract public Pair Plus(Pair source); |
|  | abstract public Pair Minus(Pair source); |
|  | abstract public string GetPair(); |
|  |  |
|  | } |
|  | } |

**Program.cs**

|  |
| --- |
| using System; |
|  | using System.Collections.Generic; |
|  | using System.Linq; |
|  | using System.Threading.Tasks; |
|  | using System.Windows.Forms; |
|  |  |
|  | namespace lab6 |
|  | { |
|  | static class Program |
|  | { |
|  | [STAThread] |
|  | static void Main() |
|  | { |
|  |  |
|  | Application.EnableVisualStyles(); |
|  | Application.SetCompatibleTextRenderingDefault(false); |
|  | Application.Run(new MainWindow()); |
|  | } |
|  | } |
|  | } |

**Вывод**

В данной лабораторной работе я ознакомился с возможностями утилиты Windows Forms и приобрел знания в настройке, отладке и конструировании своего GUI-интерфейса.