Міністерство освіти і науки, молоді та спорту України

Національний лісотехнічний університет України

*Кафедра обчислювальної техніки та*

*моделювання технологічних процесів*

Лабораторна робота №9

на тему

«Потоки файлового введення-виведення.

Серіалізація об’єктів»

з дисципліни

«**ТРіПО**»

Виконав:

студент групи КНС-11/1

*Бежнар Я.С.*

Перевірив:

асистент Сикала О.П.

Львів – 2014

**МЕТА:** Навчитися працювати з потоками файлового введення-виведення та створювати серіалізовані об’єкти.

**Завдання:**

Розробити клас, відповідно до індивідуального завдання, який повинен містити наступні складові:

* Конструктори (за замовчування та параметризований) з обробкою виключень;
* Властивості з обробкою виключень при некоректних значеннях;
* Відповідні методи для реалізації заданої функціональності;
* Реалізовані методи стандартних інтерфейсів для реалізації закладеної функціональності;

Розробити клас-обгортку для узагальненої колекції (за вибором студента), в якому повинна бути реалізована наступна функціональність:

* Формування колекції об’єктів класу відповідно до завдання;
* Відображення колекції об’єктів;
* Здійснення сортування та вибірки з колекції;
* Запис колекції у файл (об’єкта колекції);
* Зчитування колекції з файлу(об’єкта колекції).

# Варіант №2

Розробити клас для представлення комплектуючих ПК, який повинен містити наступну інформацію: назва,серійний номер, виробки, країна, ціна. Серійний номер є унікальним і не повинен співпадати. Надати можливість виводити комплектуючі за назвою або країною-виробником. Відсортувати комплектуючі за серійним номером.

**Код програми**

Complectation.cs

namespace lab\_9

{

[Serializable]

class Complectation: IComparable

{

string name;

int number;

string model;

string country;

double price;

public string Name

{

get { return name; }

set { name = value; }

}

public int Number

{

get { return number; }

set { number = value; }

}

public string Model

{

get { return model; }

set { model = value; }

}

public string Country

{

get { return country; }

set { country = value; }

}

public double Price

{

get { return price; }

set { price = value; }

}

public Complectation() { }

public Complectation(string name, int number, string model, string country, double price)

{

try

{

Name = name;

Number = number;

Model = model;

Country = country;

Price = price;

}

catch (InvalidCastException e)

{

}

}

public override bool Equals(object obj)

{

Complectation cob = (Complectation)obj;

return this.Number == cob.Number;

}

public int CompareTo(object ob)

{

Complectation cob = (Complectation)ob;

if (this.Number > cob.Number)

return 1;

else if (this.Number < cob.Number)

return -1;

else return 0;

}

}

}

PCCollection.cs

namespace lab\_9

{

class PCCollection

{

List<Complectation> PC;

public PCCollection()

{ PC = new List<Complectation>();

}

public void Add(string name, int number, string model, string country, double price)

{

try

{ PC.Add(new Complectation(name, number, model, country, price));

}

catch(InvalidCastException e)

{

}

}

public void Add(Complectation ob)

{

if(!PC.Contains(ob))

PC.Add(ob);

}

public Complectation Get(int index)

{

return PC[index];

}

public void Remove(int index)

{

PC.RemoveAt(index);

}

public void Sort()

{

PC.Sort();

}

public int Count()

{

return PC.Count;

}

public void SaveFile(string file)

{

BinaryFormatter binForm = new BinaryFormatter();

using(Stream fstream = new FileStream(file, FileMode.Create, FileAccess.Write, FileShare.Write))

{

binForm.Serialize(fstream, PC);

}

}

public void OpenFile(string file)

{

BinaryFormatter binFormat = new BinaryFormatter();

using(Stream fstream = new FileStream(file, FileMode.Open, FileAccess.Read, FileShare.Read))

{

PC = (List<Complectation>)binFormat.Deserialize(fstream);

}

}

}

}

Form1.cs

namespace lab\_9

{

public partial class Form1 : Form

{

PCCollection PC = new PCCollection();

public Form1()

{

InitializeComponent();

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AllowUserToDeleteRows = false;

}

private void DataGridShow()

{

dataGridView1.Rows.Clear();

comboBox2.Items.Clear();

comboBox3.Items.Clear();

for (int i = 0; i < PC.Count(); i++)

{

dataGridView1.Rows.Add();

dataGridView1.Rows[i].HeaderCell.Value = (i + 1).ToString();

dataGridView1.Rows[i].Cells["Column1"].Value = PC.Get(i).Name.ToString();

dataGridView1.Rows[i].Cells["Column2"].Value = PC.Get(i).Number.ToString();

dataGridView1.Rows[i].Cells["Column3"].Value = PC.Get(i).Model.ToString();

dataGridView1.Rows[i].Cells["Column4"].Value = PC.Get(i).Country.ToString();

dataGridView1.Rows[i].Cells["Column5"].Value = PC.Get(i).Price.ToString();

if (!comboBox2.Items.Contains(PC.Get(i).Name.ToString()))

comboBox2.Items.Add(PC.Get(i).Name.ToString());

if (!comboBox3.Items.Contains(PC.Get(i).Country.ToString()))

comboBox3.Items.Add(PC.Get(i).Country.ToString());

}

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

if (Convert.ToString(comboBox1.Text).Trim() == string.Empty || Convert.ToString(textBox2.Text).Trim() == string.Empty || Convert.ToString(textBox3.Text).Trim() == string.Empty || Convert.ToString(textBox4.Text).Trim() == string.Empty || Convert.ToString(textBox5.Text).Trim() == string.Empty)

{

MessageBox.Show("\tПомилка.\n\nНе всі поля заповнені даними");

return;

}

if (Convert.ToInt32(textBox5.Text) > 0)

{

PC.Add(new Complectation(Convert.ToString(comboBox1.Text), Convert.ToInt32(textBox2.Text), Convert.ToString(textBox3.Text), Convert.ToString(textBox4.Text), Convert.ToDouble(textBox5.Text)));

PC.Sort();

DataGridShow();

}

else MessageBox.Show("\tПомилка.\n\nЦіна повинна бути додатнім числом");

}

catch

{

MessageBox.Show("Поле 'Номер' та 'Ціна' повинні бути числовим значенням");

}

}

private void button2\_Click(object sender, EventArgs e)

{

DataGridShow();

}

private void button4\_Click(object sender, EventArgs e)

{

try

{

PC.Remove(dataGridView1.CurrentCell.RowIndex);

dataGridView1.Rows.RemoveAt(dataGridView1.CurrentCell.RowIndex);

DataGridShow();

}

catch

{

MessageBox.Show("Виберить елемент для видалення");

}

}

private void ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Close();

}

private void зберегтиЯкToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFileDialog saveFile = new SaveFileDialog();

saveFile.Filter = "file (\*.dat)|\*.dat|All files (\*.\*)|\*.\*";

saveFile.FileName = "PC Complectation";

saveFile.DefaultExt = "dat";

saveFile.ShowDialog();

if (saveFile.FileName != string.Empty)

PC.SaveFile(saveFile.FileName);

}

private void відкритиToolStripMenuItem\_Click(object sender, EventArgs e)

{

OpenFileDialog file = new OpenFileDialog();

file.Filter = "file (\*.dat)|\*.dat|All files (\*.\*)|\*.\*";

file.ShowDialog();

if (file.FileName != string.Empty)

PC.OpenFile(file.FileName);

DataGridShow();

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

MessageBox.Show("\tТРіПО\nЛабораторна робота №9");

}

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

dataGridView1.Rows.Clear();

for (int i = 0; i < PC.Count(); i++)

{

if (comboBox2.Items[comboBox2.SelectedIndex].ToString() == PC.Get(i).Name.ToString())

{

dataGridView1.Rows.Add();

dataGridView1.Rows[dataGridView1.RowCount - 1].HeaderCell.Value = (dataGridView1.RowCount).ToString();

dataGridView1.Rows[dataGridView1.RowCount - 1].Cells["Column1"].Value = PC.Get(i).Name.ToString();

dataGridView1.Rows[dataGridView1.RowCount - 1].Cells["Column2"].Value = PC.Get(i).Number.ToString();

dataGridView1.Rows[dataGridView1.RowCount - 1].Cells["Column3"].Value = PC.Get(i).Model.ToString();

dataGridView1.Rows[dataGridView1.RowCount - 1].Cells["Column4"].Value = PC.Get(i).Country.ToString();

dataGridView1.Rows[dataGridView1.RowCount - 1].Cells["Column5"].Value = PC.Get(i).Price.ToString();

}

}

}

catch

{

MessageBox.Show("Не вибрано жодного елемента");

}

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

dataGridView1.Rows.Clear();

for (int i = 0; i < PC.Count(); i++)

{

if (comboBox3.Items[comboBox3.SelectedIndex].ToString() == PC.Get(i).Country.ToString())

{

dataGridView1.Rows.Add();

dataGridView1.Rows[dataGridView1.RowCount - 1].HeaderCell.Value = (dataGridView1.RowCount).ToString();

dataGridView1.Rows[dataGridView1.RowCount - 1].Cells["Column1"].Value = PC.Get(i).Name.ToString();

dataGridView1.Rows[dataGridView1.RowCount - 1].Cells["Column2"].Value = PC.Get(i).Number.ToString();

dataGridView1.Rows[dataGridView1.RowCount - 1].Cells["Column3"].Value = PC.Get(i).Model.ToString();

dataGridView1.Rows[dataGridView1.RowCount - 1].Cells["Column4"].Value = PC.Get(i).Country.ToString();

dataGridView1.Rows[dataGridView1.RowCount - 1].Cells["Column5"].Value = PC.Get(i).Price.ToString();

}

}

}

catch

{

MessageBox.Show("Не вибрано жодного елемента");

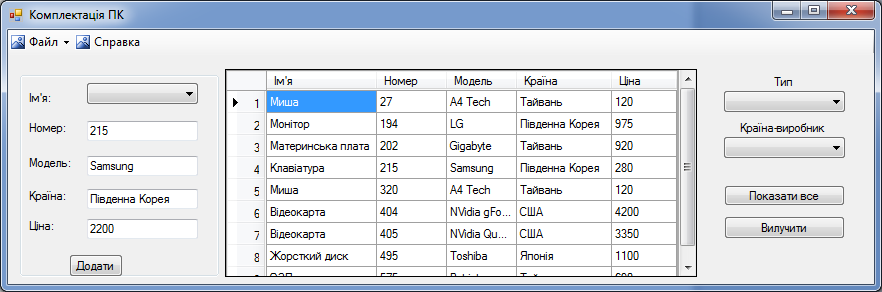
}

}

}

}

**Результат виконання програми**



**Висновок:** на лабораторній роботі я працював з потоками файлового введення-виведення та створював серіалізовані об’єкти.