Министерство образования и науки РФ

Институт информационных технологий и управления в технических системах

Севастопольский государственный университет

Лабораторная работа №4

по дисциплине: «Платформа .Net»

Выполнил:

ст.гр. ИСб/22о

Воронин И.Ю.

Проверил:

Шишкевич В.Е.

г. Севастополь

2016 г.

1. Цель работы

Изучить особенности разработки MDI-приложений в Visual Studio .Net;

Изучить способы сохранения данных в файл и загрузки из файла;

Освоить механизм сериализации и десериализации объектов.

1. Вариант задания

1. Создать учебные примеры и разобрать принцип их работы. Поместить примеры работы программ и их коды с комментариями в отчет.

2. Создать текстовый редактор NotepadC#, добавив недостающие пункты меню и функции.

3. На основании лабораторной работы 3 создать MDI-приложение. Информация в окне должна отображаться в виде таблицы. Иметь возможность делать выборку данных по различным критериям. Переносить данные из одной формы в другую.

4. Добавить формы для ввода дополнительной информации об объекте и фото объекта.

5. Добавить пункты меню для сохранения объектов в файл и загрузки. При сохранении использовать стандартные диалоговые окна и механизм сериализации. В класс добавить поле «Дата создания объекта». Поле не сериализовать, а при десериализации заново устанавливать по системной дате.

3.Код программы

Форма Form1:

namespace sharp\_lab3

{

public partial class Form1 : Form

{

public static int WindowsCounter = 0;

public List<Smartphone> ListBuffer = new List<Smartphone>();

public Form1()

{

InitializeComponent();

}

//File->New

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

{

Table t = new Table();

t.MdiParent = this;

t.Text = "Table " + ++WindowsCounter;

t.Show();

}

//File->Save

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

{

//exeption

if (WindowsCounter == 0)

{

MessageBox.Show("Необходимо создать форму!", "Предупреждение",

MessageBoxButtons.OK, MessageBoxIcon.Hand);

return;

}

SaveFileDialog sfd = new SaveFileDialog();

Form t = this.ActiveMdiChild;

if (((Table)t).list.Count > 0)

sfd.FileName = "" + ((Table)t).list[0].dateTime.TimeOfDay.Hours + " " +

((Table)t).list[0].dateTime.TimeOfDay.Minutes + " " +

((Table)t).list[0].dateTime.TimeOfDay.Seconds + " " +

((Table)t).list[0].dateTime.Date.Day + "\_" +

((Table)t).list[0].dateTime.Date.Month + "\_" +

((Table)t).list[0].dateTime.Date.Year;

sfd.Filter = "dat-файл(\*.dat)|\*.dat";

if (sfd.ShowDialog() == DialogResult.OK)

{

Serialization.Write(((Table)t).list, sfd.FileName);

}

}

//File->Open

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

{

//exeption

if (WindowsCounter == 0)

{

MessageBox.Show("Необходимо создать форму!", "Предупреждение",

MessageBoxButtons.OK, MessageBoxIcon.Hand);

return;

}

OpenFileDialog ofd = new OpenFileDialog();

if (ofd.ShowDialog() == DialogResult.OK)

{

Form t = this.ActiveMdiChild;

try

{

Serialization.Read(ref(((Table)t).list), ofd.FileName);

}

catch (System.NullReferenceException)

{

return;

}

((Table)t).FillList();

}

}

//Edit->Add

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

{

if (WindowsCounter == 0)

{

return;

}

Form t = this.ActiveMdiChild;

((Table)t).Add();

}

//Edit->Edit

private void editToolStripMenuItem1\_Click(object sender, EventArgs e)

{

if (WindowsCounter == 0)

{

return;

}

Form t = this.ActiveMdiChild;

((Table)t).Edit();

}

//Edit->Delete

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

{

if (WindowsCounter == 0)

{

return;

}

((Table)this.ActiveMdiChild).Delete();

}

//Insert->Copy

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

{

if (WindowsCounter == 0)

{

return;

}

ListBuffer.Clear();

Form t = this.ActiveMdiChild;

foreach (Smartphone s in ((Table)t).list){

Smartphone sNew = new Smartphone(s.getName(),s.getCode(),s.getAmount(),

s.getPackage(),s.recomendation,s.Image);

ListBuffer.Add(sNew);

}

}

//Insert->Paste

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

{

if (WindowsCounter == 0)

{

return;

}

Form t = this.ActiveMdiChild;

foreach (Smartphone s in ListBuffer)

{

Smartphone sNew = new Smartphone(s.getName(), s.getCode(), s.getAmount(),

s.getPackage(), s.recomendation, s.Image);

((Table)t).list.Add(sNew);

}

((Table)t).FillList();

}

//Insert->Cut

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

{

if (WindowsCounter == 0)

{

return;

}

ListBuffer.Clear();

Form t = this.ActiveMdiChild;

foreach (Smartphone s in ((Table)t).list)

{

ListBuffer.Add(s);

}

((Table)t).list.Clear();

if(((Table)t).pictureBox1.Visible)

((Table)t).pictureBox1.Hide();

((Table)t).FillList();

}

//Windows->Close

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

{

if (WindowsCounter == 0) return;

((Table)this.ActiveMdiChild).Close();

}

//Windows->Close all

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

{

if (WindowsCounter == 0) return;

int counter = Form1.WindowsCounter;

for (int i = 0; i < counter; i++)

{

closeToolStripMenuItem\_Click(sender, e);

}

}

//Windows->View->Cascade

private void toolStripMenuItem2\_Click(object sender, EventArgs e)

{

if (WindowsCounter == 0)

{

return;

}

this.LayoutMdi(System.Windows.Forms.MdiLayout.Cascade);

}

//Windows->View->Horizontal

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

{

if (WindowsCounter == 0)

{

return;

}

this.LayoutMdi(System.Windows.Forms.MdiLayout.TileHorizontal);

}

//Windows->View->Vertical

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

{

if (WindowsCounter == 0)

{

return;

}

this.LayoutMdi(System.Windows.Forms.MdiLayout.TileVertical);

}

//Sort->Name

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

{

if (WindowsCounter == 0)

{

return;

}

Form t = this.ActiveMdiChild;

((Table)t).SortName();

}

//Sort->Code

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

{

if (WindowsCounter == 0)

{

return;

}

Form t = this.ActiveMdiChild;

((Table)t).SortCode();

}

//Sort->Amount

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

{

if (WindowsCounter == 0)

{

return;

}

Form t = this.ActiveMdiChild;

((Table)t).SortAmount();

}

//help->Information

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

{

Form4 f4 = new Form4();

f4.Show();

}

}

}

Форма Form2:

namespace sharp\_lab3

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

textBox2.Text = textBox3.Text = "0";

try

{

pictureBox1.Image = Image.FromFile(ImageName);

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

pictureBox1.Show();

}catch(System.ArgumentException){

}

}

public int Key;

public String ImageName = "";

private void label1\_Click(object sender, EventArgs e)

{

}

public String getName()

{

return textBox1.Text;

}

public String getCode()

{

return textBox2.Text;

}

public String getAmount()

{

return textBox3.Text;

}

public String getPackage()

{

return textBox4.Text;

}

public String getRecomendation()

{

return richTextBox1.Text;

}

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

{

Key = 1;

this.Close();

}

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

{

Key = 2;

this.Close();

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar <= 47 || e.KeyChar >= 58) && e.KeyChar != 8)

e.Handled = true;

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar <= 47 || e.KeyChar >= 58) && e.KeyChar != 8)

e.Handled = true;

}

//двойное нажатие на картинку

private void pictureBox1\_Click(object sender, EventArgs e)

{

OpenFileDialog ofd = new OpenFileDialog();

if (ofd.ShowDialog() == DialogResult.OK)

{

ImageName = ofd.FileName;

try

{

pictureBox1.Image = Image.FromFile(ImageName);

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

}

catch (System.OutOfMemoryException) { return; }

}

}

//удаление изображения

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

ImageName = "";

pictureBox1.Image = null;

}

}

}

Форма Table:

namespace sharp\_lab3

{

public partial class Table : Form

{

//создание контейнера

public List<Smartphone> list = new List<Smartphone>();

public Table()

{

InitializeComponent();

listView1.View = View.Details;

// инициализируем заголовок таблицы

listView1.Columns.Add("Наименование", 160, HorizontalAlignment.Left);

listView1.Columns.Add("Код", 80, HorizontalAlignment.Left);

listView1.Columns.Add("Количество", 73, HorizontalAlignment.Left);

listView1.Columns.Add("Комплектация", 160, HorizontalAlignment.Left);

listView1.Columns.Add("Отзывы", 170, HorizontalAlignment.Left);

listView1.Items.Clear();

}

public void FillList()

{

//очистка списка

listView1.Items.Clear();

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

{

listView1.Items.Add(new ListViewItem(new string[] {

list[i].getName(),

"" + list[i].getCode(),

"" + list[i].getAmount(),

list[i].getPackage(),

list[i].recomendation

}

));

}

}

public void Add()

{

Form2 f2 = new Form2();

f2.ShowDialog();

String temp = f2.ImageName;

if (f2.Key == 1)

{

Int32 i1 = 0;

Int32 i2 = 0;

if (f2.getCode().Length != 0)

{

if (f2.getCode().Length > 10) i1 = (int)(Convert.ToInt64(f2.getCode().Remove(10)) % 999999999);

else i1 = (int)(Convert.ToInt64(f2.getCode()) % 999999999);

}

else i1 = 0;

if (f2.getAmount().Length != 0)

{

if (f2.getCode().Length > 10) i2 = (int)(Convert.ToInt64(f2.getAmount().Remove(10)) % 999999999);

else i2 = (int)(Convert.ToInt64(f2.getAmount()) % 999999999);

}

else i2 = 0;

list.Add(new Smartphone(

f2.getName(),

i1,

i2,

f2.getPackage(),

f2.getRecomendation(),

temp));

// MessageBox.Show("Элемент добавлен", "Внимание", MessageBoxButtons.OK);

}

FillList();

}

public void Edit()

{

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

{

if (listView1.Items[i].Selected == true)

{

Form2 f2 = new Form2();

f2.textBox1.Text = list[i].getName();

f2.textBox2.Text = "" + list[i].getCode();

f2.textBox3.Text = "" + list[i].getAmount();

f2.textBox4.Text = list[i].getPackage();

f2.richTextBox1.Text = list[i].recomendation;

f2.ImageName = list[i].Image;

f2.ShowDialog();

list[i].setAll(

f2.getName(),

Convert.ToInt32(f2.getCode()),

Convert.ToInt32(f2.getAmount()),

f2.getPackage(),

f2.getRecomendation(),

f2.ImageName

);

FillList();

}

}

}

public void Delete()

{

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

{

if (listView1.Items[i].Selected == true)

{

if (MessageBox.Show("Удалить элемент " + list[i].getName() +"?","Удалить",

MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

list.RemoveAt(i);

}

FillList();

}

}

}

private void listView1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

if (pictureBox1.Visible) pictureBox1.Hide();

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

{

if (listView1.Items[i].Selected == true)

{

Form2 f2 = new Form2();

f2.textBox1.Text = list[i].getName();

f2.textBox2.Text = "" + list[i].getCode();

f2.textBox3.Text = "" + list[i].getAmount();

f2.textBox4.Text = list[i].getPackage();

f2.richTextBox1.Text = list[i].recomendation;

f2.ImageName = list[i].Image;

try

{

f2.pictureBox1.Image = Image.FromFile(list[i].Image);

f2.pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

}

catch (System.ArgumentException) { }

catch (System.OutOfMemoryException) {

list[i].Image = "";

f2.pictureBox1.Image = null;

}

f2.ShowDialog();

list[i].setAll(

f2.getName(),

Convert.ToInt32(f2.getCode()),

Convert.ToInt32(f2.getAmount()),

f2.getPackage(),

f2.getRecomendation(),

f2.ImageName

);

FillList();

}

}

}

//вывод картинки у записи

private void listView1\_MouseClick(object sender, MouseEventArgs e)

{

if (pictureBox1.Visible)

{

pictureBox1.Hide();

return;

}

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

{

if (listView1.Items[i].Selected == true)

{

if (list[i].Image == null || list[i].Image.Length == 0) return;

if (pictureBox1.Visible)

{

pictureBox1.Hide();

return;

}

else

{

try

{

if (list[i].Image.Length != 0)

{

pictureBox1.Image = Image.FromFile(list[i].Image);

}

}

catch (System.ArgumentException)

{

return;

}

catch (System.OutOfMemoryException)

{

return;

}

pictureBox1.Bounds = new Rectangle(e.Location.X+20, e.Location.Y, 120, 120);

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

pictureBox1.Show();

}

}

}

}

private void listView1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (pictureBox1.Visible) pictureBox1.Hide();

}

public void SortName()

{

if (list == null || list.Count == 0) return;

list.Sort(SortByName.Sort);

FillList();

}

public void SortCode()

{

if (list == null || list.Count == 0) return;

list.Sort(SortByCode.Sort);

FillList();

}

public void SortAmount()

{

if (list == null || list.Count == 0) return;

list.Sort(SortByAmount.Sort);

FillList();

}

}

}

4.Выполнение программы

Было написано приложение Windows Form на языке С# в среде MS Visual Studio 2010. Данное приложение было написано использованием MDI.

При запуске программы на экране появляется окно, в котором есть следующие основные пункты: File, Edit, Insert, Windows, Sort, Help.

В меню файл можно создавать новые формы, загружать данные из файла и записывать в файл.

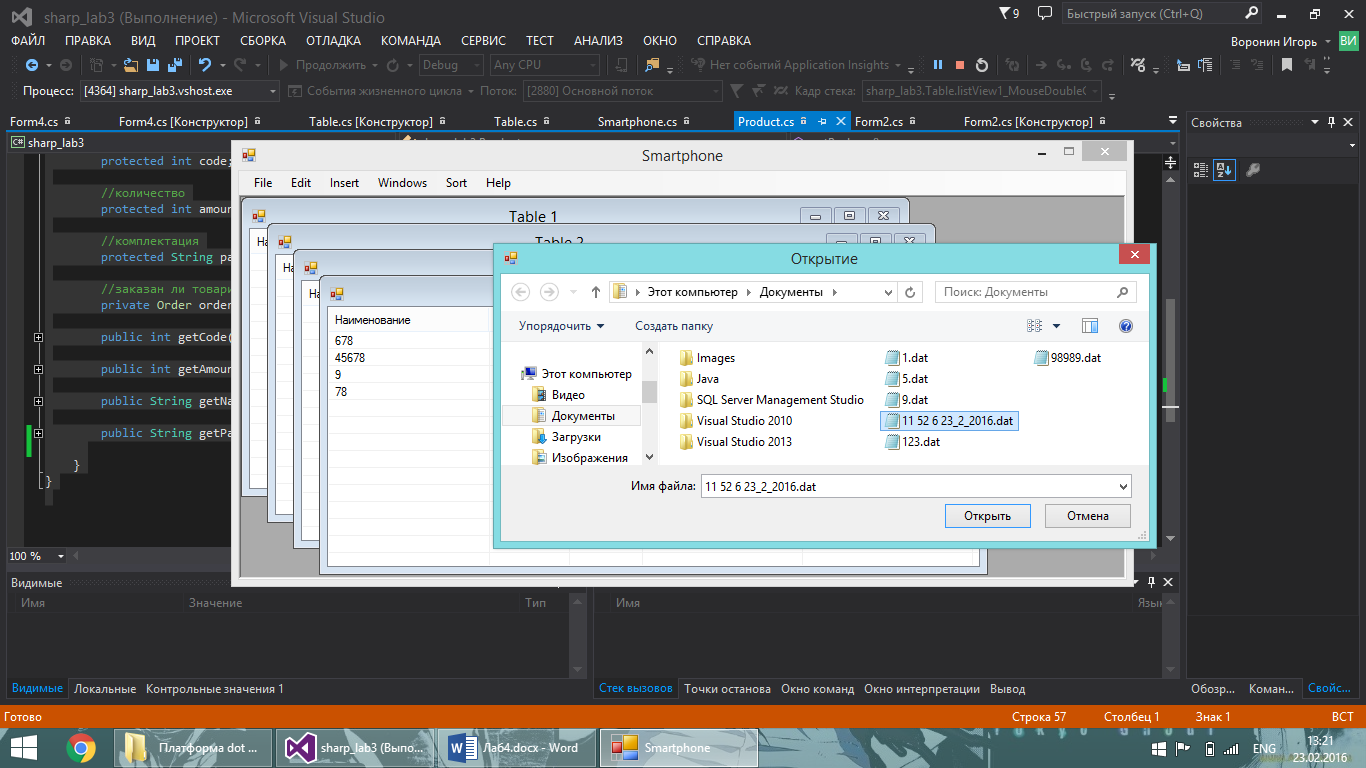


Рисунок 4.1 – Загрузка данных из формы.

При помощи пункта Edit можно добавлять элементы в таблицу, которая выделена в данный момент, редактировать элементы (также можно двойным нажатием мыши на строку в таблице), при выделении элемента можно удалить строку.

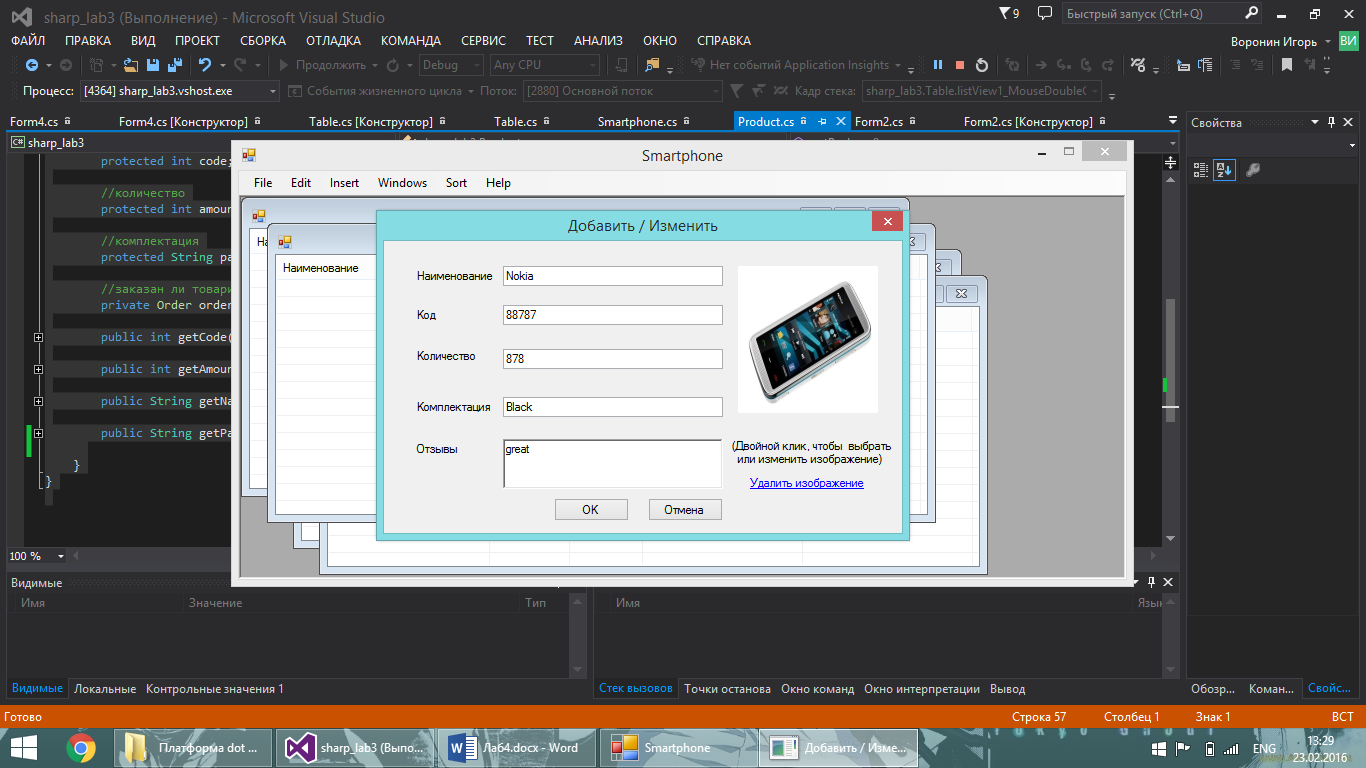


Рисунок 4.2 – Добавление элемента.

Используя функционал пункта Windows можно закрыть окно, закрыть все окна, менять вид открытых окон: каскад, вертикальное выравнивание, горизонтальное выравнивание.

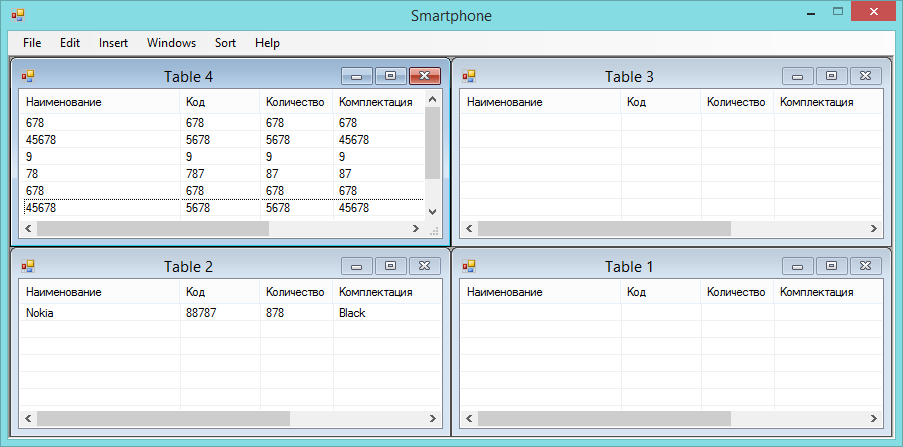


Рисунок 4.5 – Вертикальное выравнивание.

В пункте Sort можно сортировать по имени, кода и количеству товара.

В пункте Help можно вызвать функцию Information, где будет отображена информация о студенте.

ВЫВОДЫ

Было написано MDI приложение, в котором были реализованы: сериализация и десериализация, использование событий и свойств элементов панели Toolbox. Основные данные хранятся в контейнере List.