Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф.Уткина»

Кафедра ЭВМ

Отчет по лабораторной работе №12

на тему

«Разработка клиентского приложения»

по дисциплине  
**«Визуальное программирование»**

Выполнили:

Студенты группы №140

Бригада 5

Сафаров Д. А.

Тимохин Е. С.

Проверили:  
ст. преп. Хизриева Н.И.

ст. преп. Бастрычкин А.С.

**Цель работы**

Получение навыков написания клиентского приложения на языке C#

**Задание**

Требуется доработать приложение для работы с БД Supply со следующими требованиями:

• на одной форме должны располагаться все три таблицы, каждая таблица в отдельной вкладке;

• для каждой из таблиц необходимо предусмотреть обновление, добавление, изменение и удаление данных через контекстное меню;

• данные из подчиненной таблицы Supply нужно выводить в виде:  
*дата поставки, наименование поставщика, артикул товара, наименование товара, объем поставки, общая стоимость поставки*.

• при редактировании данных нельзя менять значение первичных ключей;

• при вставке данных в подчиненную таблицу пользователь не должен вручную вводить ItemID и SupplierID. Нужно предусмотреть выбор нужного поставщика и товара через выпадающие списки с наименованиями;

• для ввода телефона поставщика использовать MaskedTextBox.

**Практическая часть**

Код программы представлен в приложении А. Результат её работы представлен на рисунке 1.

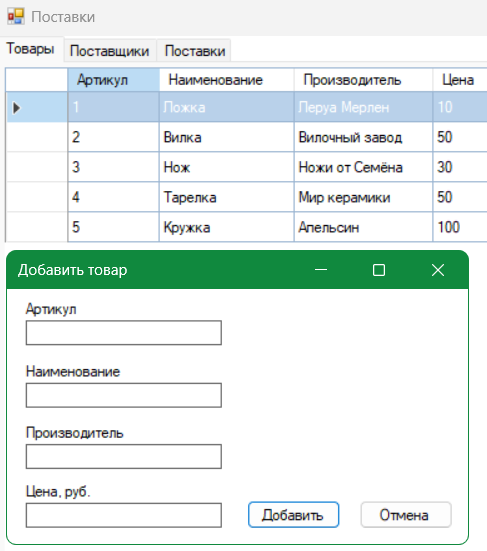
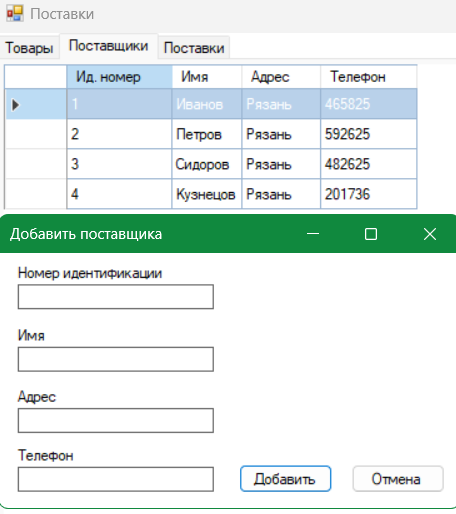
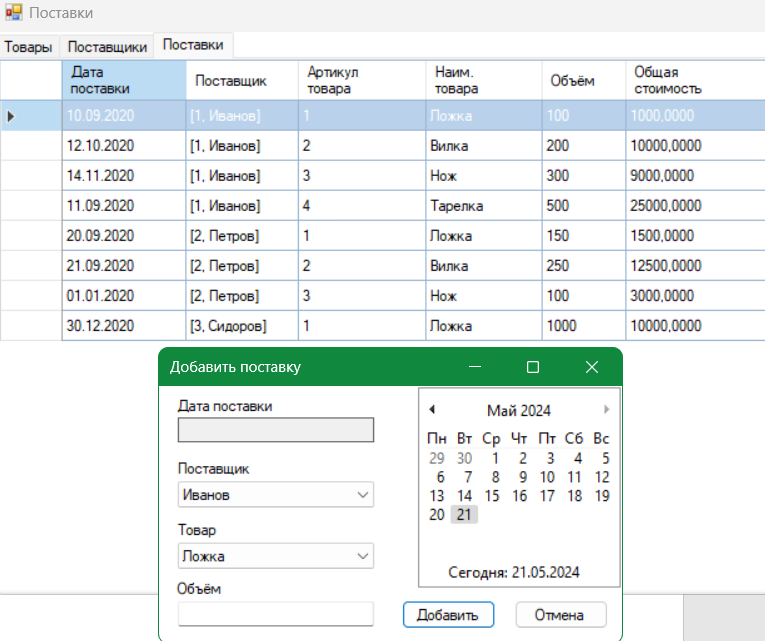


Рисунок 1 – Результаты работы программы

Приложение А. Текст программы

**AddItemForm.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace SupplyApp

{

public partial class AddItemForm : Form

{

private int \_id;

private string \_name;

private string \_manufacturer;

private decimal \_price;

public AddItemForm()

{

InitializeComponent();

}

public AddItemForm(Item item)

{

InitializeComponent();

this.Text = "Изменить товар";

txtId.ReadOnly = true;

btnAdd.Text = "Изменить";

txtId.Text = item.Id.ToString();

txtName.Text = item.Name;

txtManufacturer.Text = item.Manufacturer;

txtPrice.Text = ((double)item.Price).ToString();

}

private void txtId\_Validating(object sender, CancelEventArgs e)

{

string input = txtId.Text.Trim();

if (Regex.IsMatch(input, @"(?<=\s|^)\d+(?=\s|$)"))

{

errorProvider.SetError(txtId, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(txtId, "Ошибка!");

e.Cancel = true;

}

}

private void txtId\_Validated(object sender, EventArgs e)

{

\_id = Convert.ToInt32(txtId.Text.Trim());

}

private void txtName\_Validating(object sender, CancelEventArgs e)

{

string input = txtName.Text.Trim();

if(Regex.IsMatch(input, @"^[А-Яа-яЁё\s\-]+$"))

{

errorProvider.SetError(txtName, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(txtName, "Ошибка!");

e.Cancel = true;

}

}

private void txtName\_Validated(object sender, EventArgs e)

{

\_name = txtName.Text.Trim();

}

private void txtManufacturer\_Validating(object sender, CancelEventArgs e)

{

string input = txtManufacturer.Text.Trim();

if (Regex.IsMatch(input, @"^[А-Яа-яЁё\s\-]+$"))

{

errorProvider.SetError(txtManufacturer, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(txtManufacturer, "Ошибка!");

e.Cancel = true;

}

}

private void txtManufacturer\_Validated(object sender, EventArgs e)

{

\_manufacturer = txtManufacturer.Text.Trim();

}

private void txtPrice\_Validating(object sender, CancelEventArgs e)

{

string input = txtPrice.Text.Trim();

if (Regex.IsMatch(input, @"^\d+([.,]\d{1,2})?$"))

{

errorProvider.SetError(txtPrice, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(txtPrice, "Ошибка!");

e.Cancel = true;

}

}

private void txtPrice\_Validated(object sender, EventArgs e)

{

\_price = Convert.ToDecimal(txtPrice.Text.Trim().Replace('.',','));

}

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

{

DialogResult = ValidateChildren() ? DialogResult.OK : DialogResult.None;

if (DialogResult == DialogResult.OK)

{

AddItem();

this.Close();

}

else

{

MessageBox.Show("Введите корректные данные!", "Ошибка",

MessageBoxButtons.OK);

}

}

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

{

DialogResult = DialogResult.None;

this.Close();

}

private void AddItem()

{

try

{

using (var db = new SupplyModel())

{

if (txtId.ReadOnly)

{

Item old = db.Item.Find(\_id);

db.Item.Remove(old);

}

db.Item.Add(new Item

{

Id = \_id,

Name = \_name,

Manufacturer = \_manufacturer,

Price = \_price

});

db.SaveChanges();

}

if(txtId.ReadOnly)

MessageBox.Show("Данные изменены!", "Обновлено", MessageBoxButtons.OK);

else

MessageBox.Show("Данные добавлены!", "Добавлено", MessageBoxButtons.OK);

}

catch (Exception exc)

{

MessageBox.Show("Ошибка в данных!", "Ошибка", MessageBoxButtons.OK);

}

}

}

}

**AddSupplierForm.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Net;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace SupplyApp

{

public partial class AddSupplierForm : Form

{

private int \_id;

private string \_name;

private string \_address;

private string \_phone;

public AddSupplierForm()

{

InitializeComponent();

}

public AddSupplierForm(Supplier supplier)

{

InitializeComponent();

this.Text = "Редактировать поставщика";

txtId.ReadOnly = true;

btnAdd.Text = "Изменить";

txtId.Text = supplier.Id.ToString();

txtName.Text = supplier.Name;

txtAddress.Text = supplier.Address;

txtPhone.Text = supplier.Phone;

}

private void txtId\_Validating(object sender, CancelEventArgs e)

{

string input = txtId.Text.Trim();

if (Regex.IsMatch(input, @"(?<=\s|^)\d+(?=\s|$)"))

{

errorProvider.SetError(txtId, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(txtId, "Ошибка!");

e.Cancel = true;

}

}

private void txtId\_Validated(object sender, EventArgs e)

{

\_id = Convert.ToInt32(txtId.Text.Trim());

}

private void txtName\_Validating(object sender, CancelEventArgs e)

{

string input = txtName.Text.Trim();

if (Regex.IsMatch(input, @"^[А-Яа-яЁё\s\-]+$"))

{

errorProvider.SetError(txtName, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(txtName, "Ошибка!");

e.Cancel = true;

}

}

private void txtName\_Validated(object sender, EventArgs e)

{

\_name = txtName.Text.Trim();

}

private void txtAddress\_Validating(object sender, CancelEventArgs e)

{

string input = txtAddress.Text.Trim();

if (Regex.IsMatch(input, @"^[А-Яа-яЁё]+(?: [\s-][А-Яа-яЁё]+)\*(?:\sул\.\s[А-Яа-яЁё]+(?: [\s-][А-Яа-яЁё]+)\*\s\d+)?$"))

{

errorProvider.SetError(txtAddress, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(txtAddress, "Ошибка!");

e.Cancel = true;

}

}

private void txtAddress\_Validated(object sender, EventArgs e)

{

\_address = txtAddress.Text.Trim();

}

private void txtPhone\_Validating(object sender, CancelEventArgs e)

{

string input = txtPhone.Text.Trim();

if (Regex.IsMatch(input, @"^\d{5,7}$"))

{

errorProvider.SetError(txtPhone, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(txtPhone, "Ошибка!");

e.Cancel = true;

}

}

private void txtPhone\_Validated(object sender, EventArgs e)

{

\_phone = txtPhone.Text.Trim();

}

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

{

DialogResult = ValidateChildren() ? DialogResult.OK : DialogResult.None;

if (DialogResult == DialogResult.OK)

{

AddSupplier();

this.Close();

}

else

{

MessageBox.Show("Введите корректные данные!", "Ошибка",

MessageBoxButtons.OK);

}

}

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

{

DialogResult = DialogResult.None;

this.Close();

}

private void AddSupplier()

{

try

{

using (var db = new SupplyModel())

{

if (txtId.ReadOnly)

{

Supplier old = db.Supplier.Find(\_id);

db.Supplier.Remove(old);

}

db.Supplier.Add(new Supplier

{

Id = \_id,

Name = \_name,

Address = \_address,

Phone = \_phone

});

db.SaveChanges();

}

if (txtId.ReadOnly)

MessageBox.Show("Данные изменены!", "Обновлено", MessageBoxButtons.OK);

else

MessageBox.Show("Данные добавлены!", "Добавлено", MessageBoxButtons.OK);

}

catch (Exception exc)

{

MessageBox.Show("Ошибка в данных!", "Ошибка", MessageBoxButtons.OK);

}

}

}

}

**AddSupplyForm.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml.Linq;

namespace SupplyApp

{

public partial class AddSupplyForm : Form

{

private int \_supplier;

private int \_item;

private DateTime \_date;

private int? \_volume;

public AddSupplyForm()

{

InitializeComponent();

}

public AddSupplyForm(Supply supply)

{

InitializeComponent();

this.Text = "Редактировать поставку";

btnAdd.Text = "Изменить";

\_supplier = supply.Supplier;

\_item = supply.Item;

txtDate.Text = supply.Date.ToShortDateString();

txtDate.ReadOnly = true;

txtVolume.Text = supply.Volume.ToString();

}

private void AddSupplyForm\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "itemDataSet.Item". При необходимости она может быть перемещена или удалена.

this.itemTableAdapter1.Fill(this.itemDataSet.Item);

// TODO: данная строка кода позволяет загрузить данные в таблицу "supplierDataSet.Supplier". При необходимости она может быть перемещена или удалена.

this.supplierTableAdapter1.Fill(this.supplierDataSet.Supplier);

// TODO: данная строка кода позволяет загрузить данные в таблицу "supplyDataSet1.Item". При необходимости она может быть перемещена или удалена.

this.itemTableAdapter.Fill(this.supplyDataSet1.Item);

// TODO: данная строка кода позволяет загрузить данные в таблицу "supplyDataSet.Supplier". При необходимости она может быть перемещена или удалена.

this.supplierTableAdapter.Fill(this.supplyDataSet.Supplier);

}

private void AddSupplyForm\_Shown(object sender, EventArgs e)

{

if(\_supplier != 0)

{

boxSupplier.SelectedValue = \_supplier;

boxItem.SelectedValue = \_item;

boxItem.Enabled = false;

boxSupplier.Enabled = false;

}

Calendar.MaxDate = Calendar.TodayDate;

}

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

{

DialogResult = ValidateChildren() ? DialogResult.OK : DialogResult.None;

if (DialogResult == DialogResult.OK)

{

AddSupply();

this.Close();

}

else

{

MessageBox.Show("Введите корректные данные!", "Ошибка",

MessageBoxButtons.OK);

}

}

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

{

DialogResult = DialogResult.None;

this.Close();

}

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

{

(sender as MaskedTextBox).Select(0, 0);

}

private void AddSupply()

{

try

{

using (var db = new SupplyModel())

{

if (boxSupplier.Enabled == false)

{

Supply old = db.Supply.Find(\_supplier, \_item, \_date);

db.Supply.Remove(old);

}

db.Supply.Add(new Supply

{

Supplier = \_supplier,

Item = \_item,

Date = \_date,

Volume = \_volume

});

db.SaveChanges();

}

if (boxSupplier.Enabled == false)

MessageBox.Show("Данные изменены!", "Обновлено", MessageBoxButtons.OK);

else

MessageBox.Show("Данные добавлены!", "Добавлено", MessageBoxButtons.OK);

}

catch (Exception exc)

{

MessageBox.Show("Ошибка в данных!", "Ошибка", MessageBoxButtons.OK);

}

}

private void txtDate\_Validating(object sender, CancelEventArgs e)

{

if (!String.IsNullOrEmpty(txtDate.Text))

{

errorProvider.SetError(txtDate, String.Empty);

e.Cancel = false;

return;

}

errorProvider.SetError(txtDate, "Ошибка!");

e.Cancel = true;

}

private void txtDate\_Validated(object sender, EventArgs e)

{

\_date = Convert.ToDateTime(txtDate.Text.Trim());

}

private void boxSupplier\_Validating(object sender, CancelEventArgs e)

{

string input = boxSupplier.Text.Trim();

if (Regex.IsMatch(input, @"^[А-Яа-яЁё\s\-]+$"))

{

errorProvider.SetError(boxSupplier, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(boxSupplier, "Ошибка!");

e.Cancel = true;

}

}

private void boxSupplier\_Validated(object sender, EventArgs e)

{

using (var db = new SupplyModel())

{

\_supplier = (int) boxSupplier.SelectedValue;

}

}

private void boxItem\_Validating(object sender, CancelEventArgs e)

{

string input = boxItem.Text.Trim();

if (Regex.IsMatch(input, @"^[А-Яа-яЁё\s\-]+$"))

{

errorProvider.SetError(boxItem, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(boxItem, "Ошибка!");

e.Cancel = true;

}

}

private void boxItem\_Validated(object sender, EventArgs e)

{

\_item = (int) boxItem.SelectedValue;

}

private void txtVolume\_Validating(object sender, CancelEventArgs e)

{

string input = txtVolume.Text.Trim();

if (Regex.IsMatch(input, @"^\d+$"))

{

errorProvider.SetError(txtVolume, String.Empty);

e.Cancel = false;

}

else

{

errorProvider.SetError(txtVolume, "Ошибка!");

e.Cancel = true;

}

}

private void txtVolume\_Validated(object sender, EventArgs e)

{

\_volume = Convert.ToInt32(txtVolume.Text.Trim());

}

private void Calendar\_DateSelected(object sender, DateRangeEventArgs e)

{

txtDate.Text = Calendar.SelectionStart.ToShortDateString();

}

}

}

**SupplyForm.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace SupplyApp

{

public partial class SupplyForm : Form

{

public SupplyForm()

{

InitializeComponent();

}

// Загрузка таблиц

private void SetItemGrid()

{

if (itemGrid.ColumnCount < 4)

{

itemGrid.Columns.Add("colId", "Артикул");

itemGrid.Columns.Add("colName", "Наименование");

itemGrid.Columns.Add("colManufacturer", "Производитель");

itemGrid.Columns.Add("colPrice", "Цена");

}

itemGrid.Rows.Clear();

using (var db = new SupplyModel())

foreach (var item in db.Item.ToList())

itemGrid.Rows.Add(item.Id, item.Name, item.Manufacturer, ((double)item.Price));

}

private void SetSupplierGrid()

{

if (supplierGrid.ColumnCount < 4)

{

supplierGrid.Columns.Add("colId", "Ид. номер");

supplierGrid.Columns.Add("colName", "Имя");

supplierGrid.Columns.Add("colAddress", "Адрес");

supplierGrid.Columns.Add("colPrice", "Телефон");

}

supplierGrid.Rows.Clear();

using (var db = new SupplyModel())

foreach (var supplier in db.Supplier.ToList())

supplierGrid.Rows.Add(supplier.Id, supplier.Name, supplier.Address, supplier.Phone);

}

private void SetSupplyGrid()

{

if (supplyGrid.ColumnCount < 6)

{

supplyGrid.Columns.Add("colDate", "Дата поставки");

supplyGrid.Columns.Add("colSupplier", "Поставщик");

supplyGrid.Columns.Add("colItem", "Артикул товара");

supplyGrid.Columns.Add("colName", "Наим. товара");

supplyGrid.Columns.Add("colVolume", "Объём");

supplyGrid.Columns.Add("colOverall", "Общая стоимость");

}

supplyGrid.Rows.Clear();

using (var db = new SupplyModel())

foreach (var supply in

db.Supply.Select(supply => new

{

supply.Date,

SupplierName = supply.Supplier1.Name,

SupplierId = supply.Supplier,

ItemId = supply.Item1.Id,

ItemName = supply.Item1.Name,

supply.Volume,

Overall = supply.Volume \* supply.Item1.Price

}).ToList()

)

supplyGrid.Rows.Add(supply.Date.ToShortDateString(), new KeyValuePair<int, string>(supply.SupplierId, supply.SupplierName), supply.ItemId, supply.ItemName, supply.Volume, supply.Overall);

}

// Supply Form

private void SupplyForm\_Load(object sender, EventArgs e)

{

SetItemGrid();

SetSupplierGrid();

SetSupplyGrid();

}

// Item context menu

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

{

SetItemGrid();

}

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

{

AddItemForm add = new AddItemForm();

if (add.ShowDialog(this) == DialogResult.OK)

{

SetItemGrid();

}

}

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

{

DialogResult result = MessageBox.Show("Вы уверены, что хотите удалить этот элемент ? ",

"Удаление", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

if (itemGrid.SelectedCells.Count > 0)

{

var i = itemGrid.SelectedCells[0].OwningRow.Index;

int itemId = (int)itemGrid[0, i].Value;

using (var db = new SupplyModel())

{

Item item = db.Item.Where(x => x.Id == itemId).First();

db.Item.Remove(item);

db.SaveChanges();

}

}

}

SetItemGrid();

}

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

{

var i = itemGrid.SelectedCells[0].OwningRow.Index;

int itemId = (int)itemGrid[0, i].Value;

using (var db = new SupplyModel())

{

Item item = db.Item.Where(x => x.Id == itemId).First();

AddItemForm update = new AddItemForm(item);

if (update.ShowDialog(this) == DialogResult.OK)

{

SetItemGrid();

}

}

}

// Supplier context menu

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

{

SetSupplierGrid();

}

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

{

AddSupplierForm add = new AddSupplierForm();

if (add.ShowDialog(this) == DialogResult.OK)

{

SetSupplierGrid();

}

}

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

{

DialogResult result = MessageBox.Show("Вы уверены, что хотите удалить этот элемент ? ",

"Удаление", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

if (supplierGrid.SelectedCells.Count > 0)

{

var i = supplierGrid.SelectedCells[0].OwningRow.Index;

int supplierId = (int)supplierGrid[0, i].Value;

using (var db = new SupplyModel())

{

Supplier supplier = db.Supplier.Where(x => x.Id == supplierId).First();

db.Supplier.Remove(supplier);

db.SaveChanges();

}

}

}

SetSupplierGrid();

}

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

{

var i = supplierGrid.SelectedCells[0].OwningRow.Index;

int supplierId = (int)supplierGrid[0, i].Value;

using (var db = new SupplyModel())

{

Supplier supplier = db.Supplier.Where(x => x.Id == supplierId).First();

AddSupplierForm update = new AddSupplierForm(supplier);

if (update.ShowDialog(this) == DialogResult.OK)

{

SetSupplierGrid();

}

}

}

// Supply context menu

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

{

SetSupplyGrid();

}

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

{

AddSupplyForm add = new AddSupplyForm();

if (add.ShowDialog(this) == DialogResult.OK)

{

SetSupplyGrid();

}

}

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

{

DialogResult result = MessageBox.Show("Вы уверены, что хотите удалить этот элемент ? ",

"Удаление", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

if (supplyGrid.SelectedCells.Count > 0)

{

var i = supplyGrid.SelectedCells[0].OwningRow.Index;

int supplierId = ((KeyValuePair<int, string>)supplyGrid[1, i].Value).Key;

int itemId = (int)supplyGrid[2, i].Value;

DateTime date = Convert.ToDateTime(supplyGrid[0, i].Value);

using (var db = new SupplyModel())

{

Supply supply = db.Supply.Where(x => x.Supplier == supplierId

&& x.Item == itemId

&& x.Date == date).First();

db.Supply.Remove(supply);

db.SaveChanges();

}

}

}

SetSupplyGrid();

}

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

{

var i = supplyGrid.SelectedCells[0].OwningRow.Index;

int supplierId = ((KeyValuePair<int, string>)supplyGrid[1, i].Value).Key;

int itemId = (int)supplyGrid[2, i].Value;

DateTime date = Convert.ToDateTime(supplyGrid[0, i].Value);

using (var db = new SupplyModel())

{

Supply supply = db.Supply.Where(x => x.Supplier == supplierId

&& x.Item == itemId

&& x.Date == date).First();

AddSupplyForm update = new AddSupplyForm(supply);

if (update.ShowDialog(this) == DialogResult.OK)

{

SetSupplyGrid();

}

}

}

}

}