SQL

1нф – атомарность

2нф – частична функциональна зависимость (зависимость от части ключевого поля не ключевого роля)

3нф - отсуствие транзитивных зависимостей (зависимость не ключевых ключей)

Составные ключи – значание которых не повторяются

Nchar – префикс n – Юникод

Автоматом зополнять индентификатор – Свойство столбцов – Спецификация индентифицирующим столбцом ->ДА (прирост).

Выбрать все записи из таблици

select \* from dbo.Classes

Выбор двох полей таблицы

select [Name], RoomNumber from dbo.Classes;

Выборка с условием

select \* from dbo.Classes

where [Name] = '2-Б';

Вставка данных в таблицу

insert into dbo.Classes ([Name], RoomNumber)

values ('7-Г',250),('8-Г',259);

Обновить данные в таблице

update Classes set RoomNumber = 260 where [Name] ='8-Г';

Заменить по сходству

update Classes set RoomNumber = 000 where [Name] like '1%'

Удаление данных

delete from Classes where RoomNumber like 2;

select [Name], RoomNumber from dbo.Classes where [Name] like '5-%';

Сортировка

select [Name], RoomNumber from dbo.Classes order by [Name] desc

select [Name], RoomNumber from dbo.Classes where not (RoomNumber >200 and [Name] like '%-Г');

ВИБОРКА В IN

select [Name], RoomNumber from dbo.Classes where [Name] in ('2-Б','5-В');

BETWEEN

select [Name], RoomNumber from dbo.Classes where RoomNumber between 150 and 300;

Distinct – без повтора

select distinct RoomNumber from classes

Среднее значение записей поля

select avg(RoomNumber) from dbo.Classes

Провести связь меж таблицами – пкм – отношения – спецификация столбца

Виборка данных с двох таблиц

select [Classes].[Name], [Head].[Name] from Classes,Head where Head.ID = Classes.HeadID;

Другой способ

select [Classes].[Name], [Head].[Name] from Classes join Head on Head.ID = Classes.HeadID;

select [Classes].[Name], [Head].[Name] from Classes join Head on Head.ID = Classes.HeadID where RoomNumber < 250

select [Classes].[Name], [Head].[Name] from Classes join Head on Head.ID = Classes.HeadID where Head.[Name] like 'Шалу%'

Вложеный запрос

select Classes.[Name] from Classes where Classes.HeadID =

(select Head.ID from Head where Head.[Name] like 'Шалун%')

LeftJoin – выбераем все значение с левой таблицей

select Classes.[Name], Head.[Name] from Classes left join Head on Classes.HeadID = Head.ID

select Classes.[Name], Head.[Name] from Classes left join Head on Classes.HeadID = Head.ID where Classes.HeadID is null

UNION

select Classes.[Name] from Classes

union

select Head.[Name] from Head

Назвать колонку

select Classes.[Name] as azaza, Head.[Name] as Axaxa from Classes, Head where Classes.HeadID = Head.ID

select \* from Classes where Classes.HeadID in (select Head.ID from Head )

Создание БД с Visual Studio

VS – ServerExploler

Хранимые процедуры

Выполняются вместе с БД.

CREATE PROCEDURE [dbo].[AddArticleleData]

(

@categ nvarchar(20),

@tit nvarchar(140),

@tex text

)

AS

INSERT INTO dbo.Article (Article.Category,Article.Title,Article.[Text],Article.AvgRating,Article.CountRated,Article.AddedDate)

VALUES (@categ, @tit, @tex, 5, 1, CURRENT\_TIMESTAMP)

RETURN 0

Пейджинг

CREATE PROCEDURE [dbo].[SelectPartialData]

(

@cat nvarchar(20),

@pageSize int = 5,

@pageIndex int = 0

)

AS

Select \* From

(

SELECT Id, Category, Title, [Text], AvgRating/CountRated as Rating, AddedDate, ROW\_NUMBER() OVER (Order By AddedDate Desc) as RowNum

FROM dbo.Article Where Article.Category = @cat

) content

WHERE content.RowNum BETWEEN (@pageIndex\*@pageSize +1) and ((@pageIndex +1)\* @pageSize) Order By AddedDate Desc

RETURN 0

Entire System (уровни)

1. Уровень представление данных dal (dal знает только о базе данных);
2. Уровень бизнес – логики bll (bll общается и з тем и з тем);
3. Уровень доступа к данным presentable (знает о bll, но не знает о dal);

Если изпользуется sql аунтификация,то uid=dsdsds;pwd=sdsdsds; вместо Integrated

using System.Data.SqlClient;

using System.Data.Common;

string connection = @"Data Source=KEINELUST-PC\SQLEXPRESS; Initail Catalog=BANK; Integrated Security=true";

\*\*\*

Вытянуть данные из БД

using System.Data.SqlClient;

using System.Data.Common;

namespace BankRetail

{

class DAL

{

string connectionString = @"Data Source=KEINELUST-PC;Initial Catalog=Bank;Integrated Security=True";

public ArrayList GetAllDebitors()

{

ArrayList allDebitors = new ArrayList();

SqlConnection conn = new SqlConnection(connectionString);

SqlCommand com = new SqlCommand("SELECT \*FROM dbo.Debitors", conn);

conn.Open();

SqlDataReader dr = com.ExecuteReader();

if (dr.HasRows)

foreach (DbDataRecord result in dr)

{

allDebitors.Add(result);

}

conn.Close();

return allDebitors;

}

ПОМЕСТИТЬ

DAL dal = new DAL();

public MainForm()

{

InitializeComponent();

ArrayList allDebitors = dal.GetAllDebitors();

dgv\_debitors.DataSource = allDebitors;

}

Настройки DatGridView

public void SettingDGV\_Debitors()

{

try

{

dgv\_debitors.Columns["ID"].Visible = false;

dgv\_debitors.Columns["PostNumber"].Visible = false;

dgv\_debitors.Columns["PhoneNumber"].Visible = false;

dgv\_debitors.TopLeftHeaderCell.Value = "#";

}

catch { }

}

}

AutoSizeColumns -> Fill // Заполнить все пространство Грида

\*\*\*

using (SqlConnection conn = new SqlConnection(connectionString)) //закрытие соиденения

\*\*\*

Событие выбора записи из ГридВью в TextBox при выборе в гриде

GridView ->Events -> CellEnter

private void dgv\_debitors\_CellEnter(object sender, DataGridViewCellEventArgs e)

{

txbx\_debitorID.Text = dgv\_debitors.Rows[e.RowIndex].Cells[0].Value.ToString();

txbx\_debitorName.Text = dgv\_debitors.Rows[e.RowIndex].Cells[1].Value.ToString();

txbx\_debitorPostNumber.Text = dgv\_debitors.Rows[e.RowIndex].Cells[2].Value.ToString();

string phone = dgv\_debitors.Rows[e.RowIndex].Cells[3].Value.ToString();

txbx\_phoneNumber.Text = (phone == String.Empty) ? "Нет данных" : phone;

}

\*\*\*

Оптимизация запросов

Для каждого CellEnter подписыватся в FormLoad

Datagridview отписать от события cellenter, для формы сделать событие FormLoad и подписатся на CellEnter.

private void dgv\_debitors\_CellEnter(object sender, DataGridViewCellEventArgs e)

{

txbx\_debitorID.Text = dgv\_debitors.Rows[e.RowIndex].Cells[0].Value.ToString();

txbx\_debitorName.Text = dgv\_debitors.Rows[e.RowIndex].Cells[1].Value.ToString();

txbx\_debitorPostNumber.Text = dgv\_debitors.Rows[e.RowIndex].Cells[2].Value.ToString();

string phone = dgv\_debitors.Rows[e.RowIndex].Cells[3].Value.ToString();

txbx\_phoneNumber.Text = (phone == String.Empty) ? "Нет данных" : phone;

dgv\_credits.DataSource = dal.GetAllCreditsForDebitors(dgv\_debitors.CurrentRow.Cells["ID"].Value.ToString());

}

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

{

dgv\_debitors.CellEnter += new DataGridViewCellEventHandler(dgv\_debitors\_CellEnter);

}

Закрытие приложения

На форме юзать событые FormClosinfg

private void MainForm\_FormClosing(object sender, FormClosingEventArgs e)

{

if(MessageBox.Show("Хотите выйти?", "Bank Manager", MessageBoxButtons.OKCancel) == DialogResult.Cancel){

e.Cancel = true;

}

else{

e.Cancel = false;

}

}

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

{

this.Close();

//

Application.Exit();

}

Добавление данных в базу

public bool SaveNewDebitor(string Id, string Name, string PostNumber, string PhoneNumber)

{

bool flag = false;

string query = String.Format("Insert Into Debitors" +

"([Id], [Name], [PostNumber], [PhoneNumber])" +

"Values ('{0}', '{1}', '{2}', '{3}')", Id, Name, PostNumber, (PhoneNumber == String.Empty)? null:PhoneNumber);

using (SqlConnection con = new SqlConnection(connectionString))

{

SqlCommand com = new SqlCommand(query, con);

try

{

con.Open();

if (com.ExecuteNonQuery() == 1)

flag = true;

}

catch { };

}

return flag;

}

\*\*\*

public NewDebitorForm()

{

InitializeComponent();

txbx\_debitorID.Text = Guid.NewGuid().ToString();//Генерация 16 бит ключа

}

private void btn\_AddNewDebitor\_Click(object sender, EventArgs e)//Событие нажатия на кнопку добавления

{

if (dal.SaveNewDebitor(txbx\_debitorID.Text.Trim(), txbx\_debitorName.Text.Trim(), txbx\_debitorPostNumber.Text.Trim(), txbx\_phoneNumber.Text.Trim()))

this.DialogResult = DialogResult.OK;

else

this.DialogResult = DialogResult.Cancel;

}

В коде главной формы

private void createNewDebitorToolStripMenuItem\_Click(object sender, EventArgs e)//Меню по открытью формы добавления дебитора

{

NewDebitorForm NewDebitor = new NewDebitorForm();

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

{

dgv\_debitors.DataSource = dal.GetAllDebitors();//Обновити дані

MessageBox.Show("Новый дебитор уcпешно создан", "Банк", MessageBoxButtons.OK);

}

else

MessageBox.Show("Новый дебитор не создан!!!", "Банк", MessageBoxButtons.OK);

}

После добавления не забыть обновить данные в GridView

Уровень DAL не должен конвертировать данные в другие типы, этим занимается уровент DLL

Дата – DateTimePicker

Выпадающий список - listBox

**Робота над формой кредитов**

Правильный метод DAL

public bool NewCredit(Guid ID, Guid debitorId, int amount, int balance, DateTime OpenDate)

{

bool flag = false;

string query = String.Format("Insert Into dbo.Credits ID, DebitorId,Amount,Balance, OpenDate VALUES ('{0}','{1}','{2}','{3}','{4}')",ID,debitorId,amount,balance,OpenDate);

using (SqlConnection con = new SqlConnection(connectionString))

{

SqlCommand com = new SqlCommand(query, con);

try

{

con.Open();

if (com.ExecuteNonQuery() == 1)

flag = true;

}

catch (Exception)

{ }

}

return flag;

}

\*\*\*

Выпадающий список

ListBox->ValueMember->ID;lISTbOX->DisplayMember->ID

ListBox2->ValueMember->ID;ListBox->DisplayMember->Name

public partial class NewCredit : Form

{

ArrayList allDebitors;

DAL dal = new DAL();

public NewCredit()

{

InitializeComponent();

txbx\_creditsID.Text = Guid.NewGuid().ToString();

allDebitors = dal.GetAllDebitors();

ltbx\_debitorid.DataSource = allDebitors;

ltbx\_debitorname.DataSource = allDebitors;

}

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

{

NewCredit ncredit = new NewCredit();

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

{

dgv\_credits.DataSource = dal.GetAllCreditsForDebitors(dgv\_debitors.CurrentRow.Cells["ID"].Value.ToString());

MessageBox.Show("Новый rhtlbn уcпешно создан", "Банк", MessageBoxButtons.OK);

}

else

MessageBox.Show("Новый кредит не создан!!!", "Банк", MessageBoxButtons.OK);

}

private void btn\_OpenNewCredit\_Click(object sender, EventArgs e)

{

if (dal.NewCredit(new Guid(txbx\_creditsID.Text), new Guid(ltbx\_debitorid.SelectedValue.ToString()), Int32.Parse(txbx\_credit\_amount.Text), Int32.Parse(txbx\_creditbalance.Text), creditdate.Value))

{

this.DialogResult = DialogResult.OK;

}

else { this.DialogResult = DialogResult.No; }

}

\*\*\*

Динамическое заполнение из одного текст бокса в другой

textbox->TextChanged  
private void txbx\_credit\_amount\_TextChanged(object sender, EventArgs e)

{

txbx\_credit\_amount.Text = txbx\_credit\_amount.Text.Trim().Replace(" ", "");

txbx\_creditbalance.Text = txbx\_credit\_amount.Text;

}

\*\*\*

Обрабока ввода символов в TextBox

private void txbx\_credit\_amount\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar < 48 || e.KeyChar > 57)&& e.KeyChar!=8)

{

e.Handled = true;

}

}

\*\*\*

Событие Leave для TextBox в которои провиряется сумма кредита

private void txbx\_credit\_amount\_Leave(object sender, EventArgs e)

{

if (Int32.Parse(txbx\_credit\_amount.Text) < 100 || Int32.Parse(txbx\_credit\_amount.Text) > 1000000 || txbx\_credit\_amount.Text == string.Empty)

{

lbl\_CreditAmountMessage.Text = "Не допустимая сумма в поле";

lbl\_CreditAmountMessage.ForeColor = Color.Red;

btn\_OpenNewCredit.Enabled = false;

}

else

{

lbl\_CreditAmountMessage.Text = "ОК";

lbl\_CreditAmountMessage.ForeColor = Color.Red;

btn\_OpenNewCredit.Enabled = true;

}

Событие выбора в ListBox

void ltbx\_debitorid\_SelectedIndexChanged(object sender, EventArgs e)

{

allCredits = dal.GetAllCreditsForDebitors(ltbx\_debitorid.SelectedValue.ToString());

ltbx\_CreditAmount.DataSource = allCredits;

ltbx\_CreditId.DataSource = allCredits;

ltbx\_CreditBalance.DataSource = allCredits;

ltbx\_CreditId.DisplayMember = "ID";

ltbx\_CreditId.ValueMember = "ID";

ltbx\_CreditAmount.DisplayMember = "Amount";

ltbx\_CreditBalance.DisplayMember = "Balance";

}

Проверка на числовые данные

int res;

Int16.TryParse(e.KeyChar.ToString(),out res)

Транзкакция в DAL

internal bool SaveNewPayments(Guid ID, Guid CreditID, decimal p, DateTime dateTime)

{

bool flag = true;

using (SqlConnection con = new SqlConnection(connectionString))

{

con.Open();

SqlTransaction sqlTransaction = con.BeginTransaction();

SqlCommand com = con.CreateCommand();

com.Transaction = sqlTransaction;

try

{

string query = String.Format("Insert Into Payments(Id,CreditsId,Amount,PaymentDate) " +

" Values ('{0}','{1}',{2},'{3}')",ID,CreditID,p,dateTime);

com.CommandText = query;

com.ExecuteNonQuery();

query = String.Format("Update Credits Set Balance = (Balance - {0}) Where ID = {1}", p,CreditID);

com.CommandText = query;

com.ExecuteNonQuery();

sqlTransaction.Commit();

}

catch(Exception)

{

sqlTransaction.Rollback();

flag = false;

}

finally {

if (con.State == System.Data.ConnectionState.Open)

{ con.Dispose(); }

}

}

return flag;

}

\*\*\*

Параметризированные запросы

public bool NewCredit(Guid ID, Guid debitorId, int amount, int balance, DateTime OpenDate)

{

bool flag = false;

string query = String.Format("Insert Into Credits (id, DebitorId, Amount, Balance, OpendDate) VALUES (@id,@DebitorId,@Amount,@Balance,@OpenDate)");

using (SqlConnection con = new SqlConnection(connectionString))

{

SqlCommand com = new SqlCommand(query, con);

SqlParameter sp = new SqlParameter();

sp.ParameterName = "@id";

sp.Value = ID;

sp.SqlDbType = System.Data.SqlDbType.UniqueIdentifier;

com.Parameters.Add(sp);

sp = new SqlParameter();

sp.ParameterName = "@DebitorId";

sp.Value = debitorId;

sp.SqlDbType = System.Data.SqlDbType.UniqueIdentifier;

com.Parameters.Add(sp);

sp = new SqlParameter();

sp.ParameterName = "@Amount";

sp.Value = amount;

sp.SqlDbType = System.Data.SqlDbType.Money;

com.Parameters.Add(sp);

sp = new SqlParameter();

sp.ParameterName = "@Balance";

sp.Value = balance;

sp.SqlDbType = System.Data.SqlDbType.Money;

com.Parameters.Add(sp);

sp = new SqlParameter();

sp.ParameterName = "@OpenDate";

sp.Value = OpenDate;

sp.SqlDbType = System.Data.SqlDbType.DateTime;

com.Parameters.Add(sp);

try

{

con.Open();

if (com.ExecuteNonQuery() == 1)

flag = true;

}

catch (Exception)

{ }

}

return flag;

}

Хранимые процедуры

internal ArrayList GetAllCreditsForDebitors(string debitorID)

{

ArrayList allCredits = new ArrayList();

using (SqlConnection conn = new SqlConnection(connectionString))

{

SqlCommand com = new SqlCommand("GetAllCreditForCurrentDebitor",conn);

com.CommandType = System.Data.CommandType.StoredProcedure;

SqlParameter sp = new SqlParameter();

sp.ParameterName = "@debitorId";

sp.Value = new Guid(debitorID);

sp.SqlDbType = System.Data.SqlDbType.UniqueIdentifier;

sp.Direction = System.Data.ParameterDirection.Input;

com.Parameters.Add(sp);

conn.Open();

try

{

SqlDataReader dr = com.ExecuteReader();

if (dr.HasRows)

foreach (DbDataRecord result in dr)

{

allCredits.Add(result);

}

}

catch { }

}

return allCredits;

**Локальное сохранение данных**

internal bool SaveDbToLacal()

{

bool result = true;

StreamWriter sw = new StreamWriter(new FileStream("Debitors.csv", FileMode.Create),Encoding.GetEncoding(1251));

string query;

using (SqlConnection con = new SqlConnection(connectionString))

{

try

{

query = String.Format("Select \* From Debitors");

SqlCommand com = new SqlCommand(query,con);

con.Open();

SqlDataReader sdr = com.ExecuteReader();

sw.WriteLine(@"""Id"";""Name"";""PostNumber"";""PhoneNumber""");

if (sdr.HasRows)

{

while (sdr.Read())

{

sw.WriteLine(@"""" + sdr[0].ToString() + @""";""" + sdr[1].ToString() + @""";""" + sdr[2].ToString() + @""";""" + sdr[3].ToString() + @"""", Encoding.ASCII);

}

}

else {

}

sw.Dispose();

}

catch

{

result = false;

}

}

return result;

}

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

{

if (dal.SaveDbToLacal())

{

MessageBox.Show("Данные успешно сохранены", "Банк менеджер", MessageBoxButtons.OK, MessageBoxIcon.Information, MessageBoxDefaultButton.Button1);

}

else { MessageBox.Show("Данные не сохранены", "Банк менеджер", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1); }

}

Поиск на стороне клиента

List<DataGridViewRow> searchedRows;

int currrentRow;

private void btn\_Search\_Click(object sender, EventArgs e)

{

searchedRows = new List<DataGridViewRow>();

string debName = txb\_searchname.Text.Trim();

string debPost = txb\_searchpost.Text.Trim();

string debPhone = txb\_searchphone.Text.Trim();

foreach (DataGridViewRow row in dgv\_debitors.Rows)

{

if

(

row.Cells["Name"].FormattedValue.ToString().Contains(debName) &&

row.Cells["PostNumber"].FormattedValue.ToString().Contains(debPost) &&

row.Cells["PhoneNumber"].FormattedValue.ToString().Contains(debPhone)

)

searchedRows.Add(row);

}

if (searchedRows.Count == 0)

{

MessageBox.Show("No Data");

btn\_next.Enabled = false;

return;

}

MessageBox.Show("Найдено " + searchedRows.Count +" записей");

btn\_next.Enabled = true;

}

private void btn\_next\_Click(object sender, EventArgs e)

{

dgv\_debitors.CurrentCell = searchedRows[currrentRow].Cells[1];

currrentRow++;

}

**Поиск на стороне сервера**

Create Procedure DebSearch

(

@debName nvarchar(50),

@debPostNumber nvarchar(50),

@debPhoneNumber nvarchar(30)

)

AS

Select \* From Debitors

Where Debitors.Name like '%'+@debName+'%'AND

cast(Debitors.PostNumber as nvarchar(50)) like '%'+@debPostNumber+'%'AND

Debitors.PhoneNumber like '%'+@debPostNumber+'%'

RETURN

Execute DebSearch 'ич','',''

**Disconnected model**

App-файл

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<connectionStrings>

<add name="BankConnectionString"

connectionString ="Data Source=KEINELUST-PC;Initial Catalog=Bank;Integrated Security=True;"

providerName ="System.Data.SqlClient" />

</connectionStrings>

</configuration>

Подключить ссылку System.Configuration

Подключить пространство System.Configuration;

string connectionString = ConfigurationManager.ConnectionStrings["BankConnectionString"].ConnectionString;

DataColumn

DataColumn ID = new DataColumn("Id", typeof(int));

ID.Caption = "Id";

ID.AllowDBNull = false;

ID.Unique = true;

ID.ReadOnly = true;

ID.AutoIncrement = true;

ID.AutoIncrementStep = 1;

ID.AutoIncrementSeed = 0;

**DataTable**

Debitors = new DataTable("Debitors");

Debitors.Columns.AddRange(new DataColumn[] { ID, Name, PostNumber, PhoneNumber });

DataRow DebitorsNewRow = Debitors.NewRow();

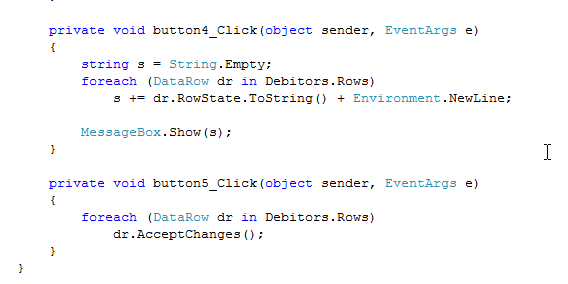
DebitorsNewRow["ID"] = Guid.NewGuid();

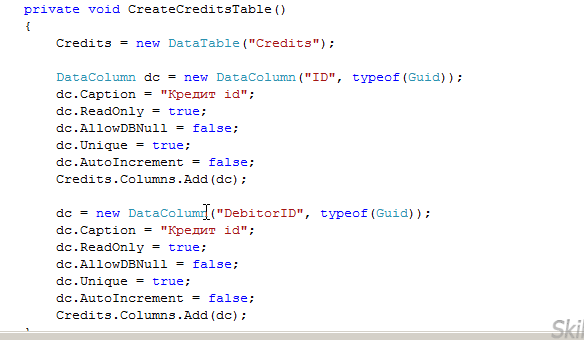
DebitorsNewRow["Name"] = "Гудок Александа";

DebitorsNewRow["PostNumber"] = "3242432";

DebitorsNewRow["PhoneNumber"] = "+38 045 456 45 78";

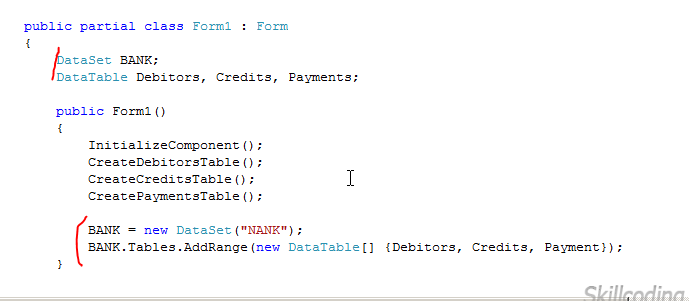
Debitors.Rows.Add(DebitorsNewRow);



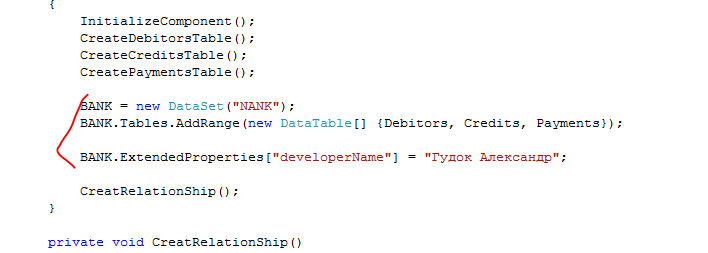


Связать валидные значение для кредитов в поле IdDebitors

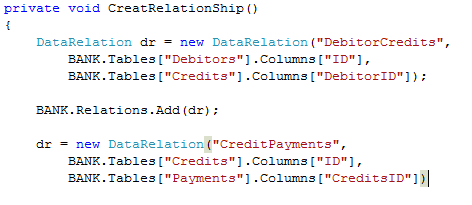
DataSet



Добавляем таблицы в DataSet

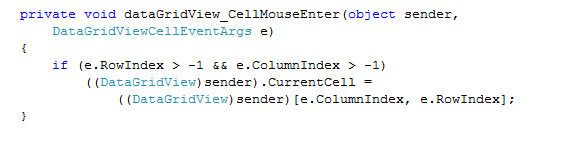


Устанавливаем отношение между полями таблиц

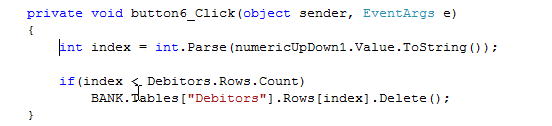


Динамическое выделение ячеек DataGriedView (CellMouseEnter)

Для грида событие –> CellMouseEnter



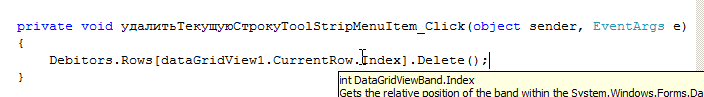
Удаление строк с использованием numericUpDown



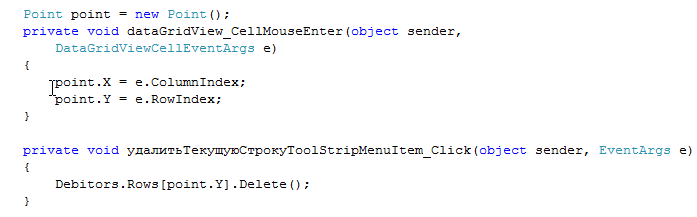
Контекстное меню

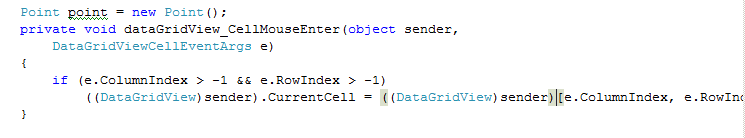
ToolBox-> ContextMenuStrip;

В свойствах Грида привязываем к нему ContextMenuStrip

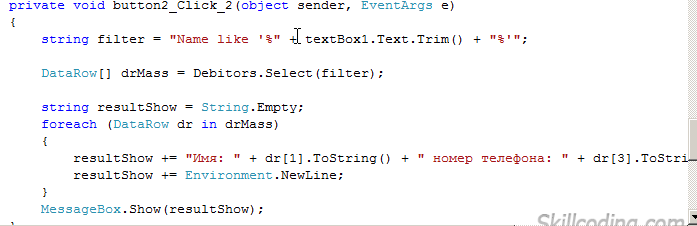


Другой способ

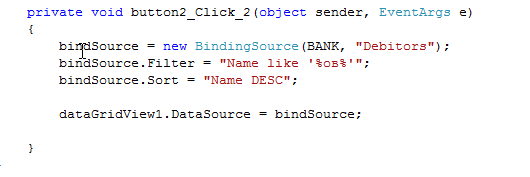




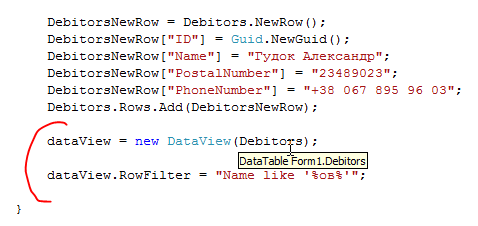
Поиск



BindingSource

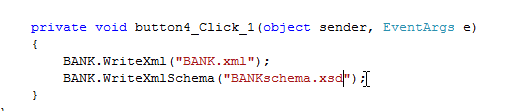


DataView

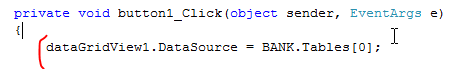


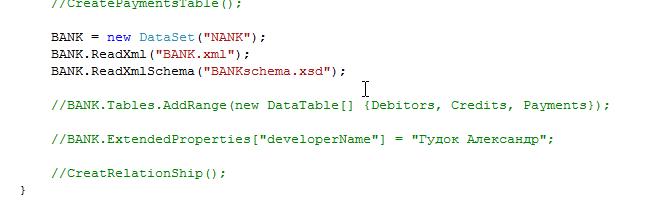


Сохранение данных в xml

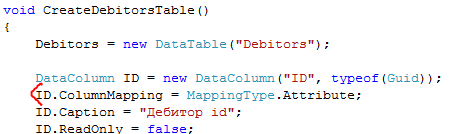


Чтение из xml-файла

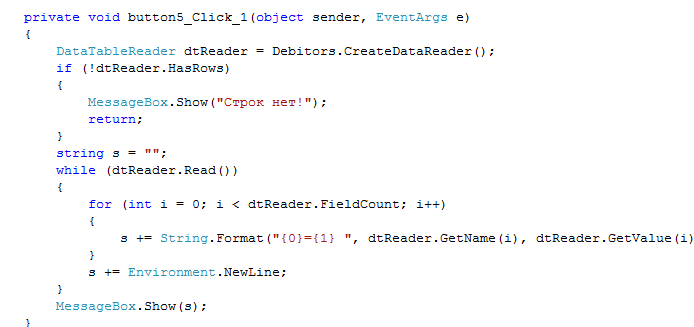


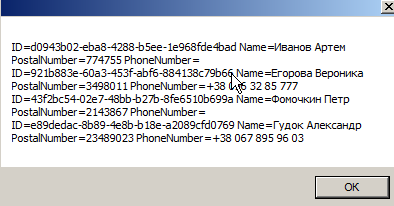


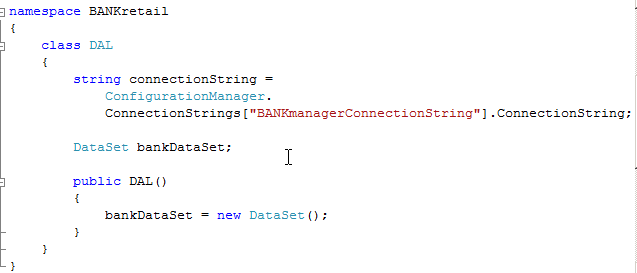
Изменить стиль сохранения в xml



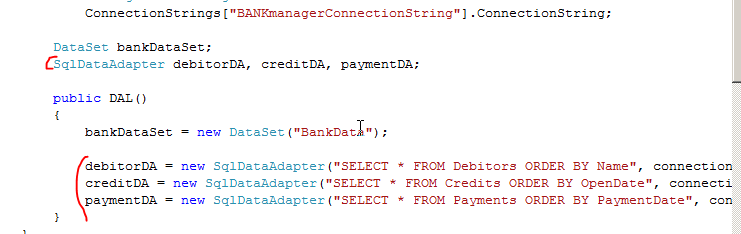
DataTableReader

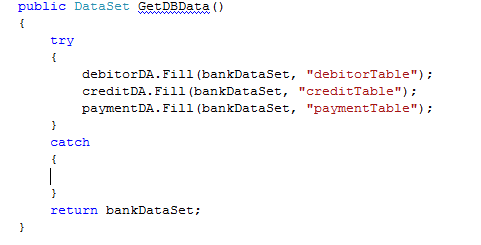




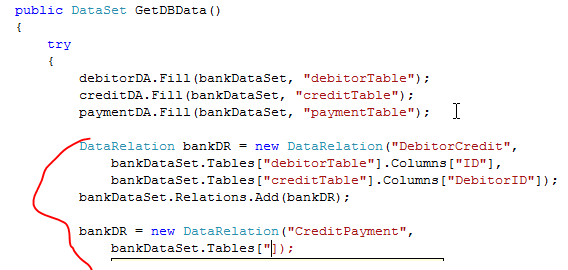


Загрузка данных c базы DataAdapter

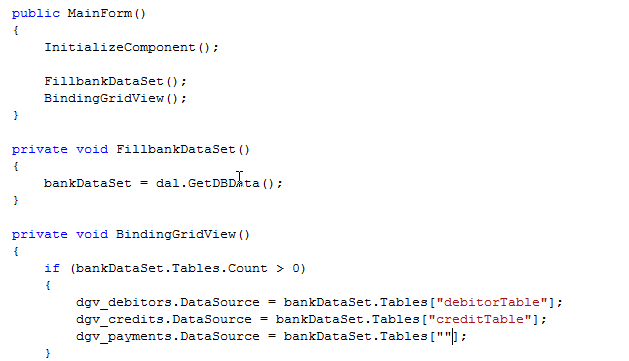




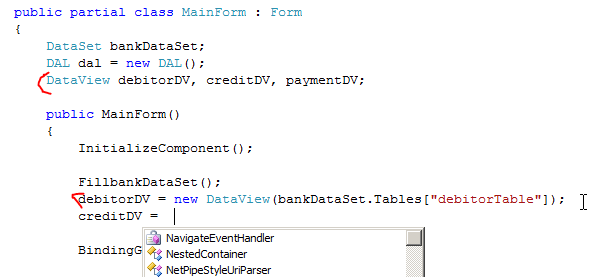
Устанавливаем отношение между таблицами

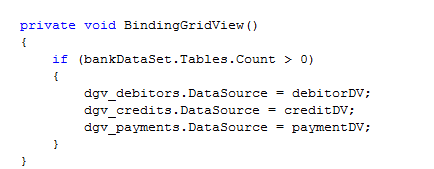
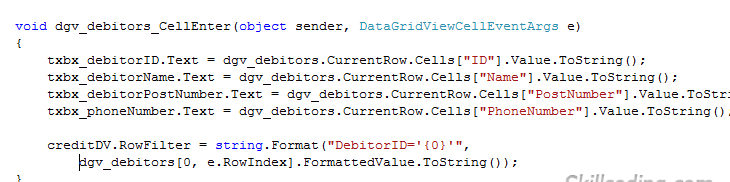


Пкрадача данных с DataSet в GridView

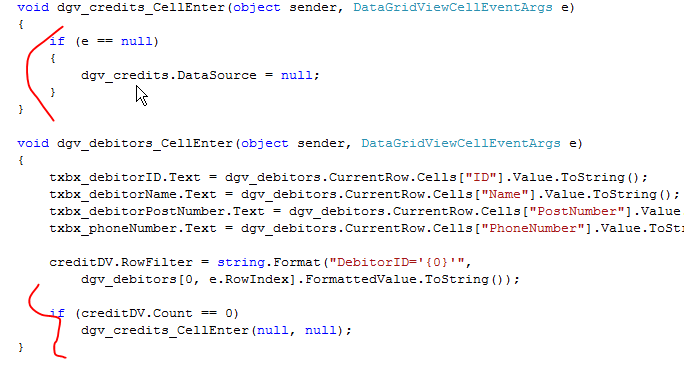


Отображения данных для выбраного дебитора DataView

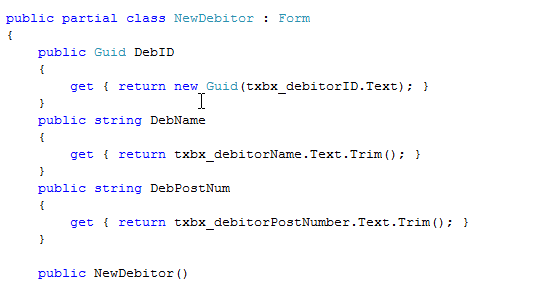


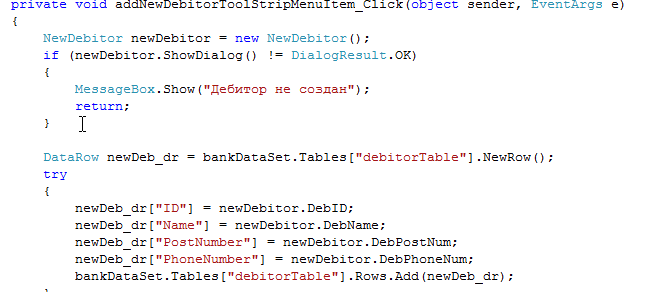
  


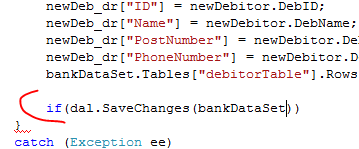
Сброс данных в DataGridView



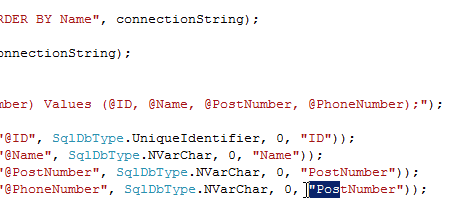
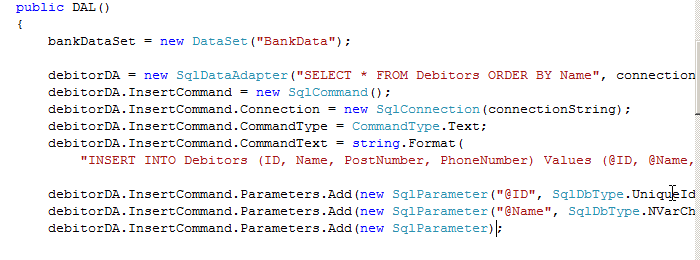
Добавление дебитора



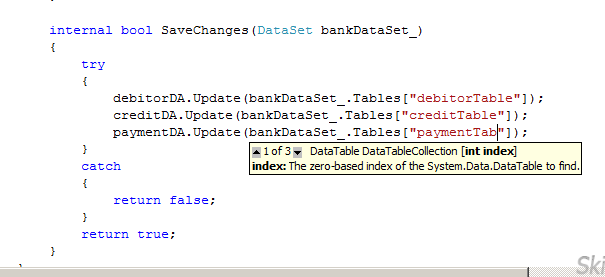




Определяем команды для DataAdaptet



Апдейт данных DataAdapter



Автонастройка DataAdapter

Для CommandBuilder Обязательно нужен запрос на выборку для адаптеров

Использовать только для простой выборки из одной таблици

