Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель А.А. Шальнев

(подпись) (и.о. фамилия)

« » 2016 г.

Комплексный отчет

по лабораторным работам

дисциплины

«Сервис-ориентированные технологии разработки программных систем»

ЛР 09.04.04.04.000 О

Студент группы 8ПИ-61 А.О.Корней

Преподаватель старший преподаватель А.А. Шальнев

и.о., фамилия

БАРНАУЛ 2016

# Лабораторная работа 1

## 1 Задание

**«Модель данных. Создание DataSet»**

**Цель работы:**

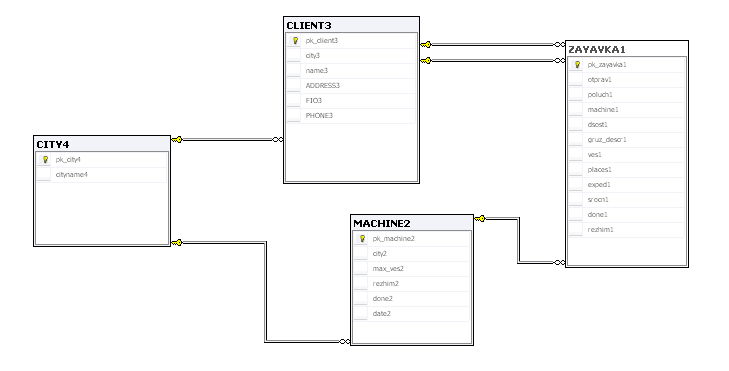
получение навыков по созданию модели данных в виде DataSet.

**Задачи работы:**

1. разработать структуру БД;
2. создать модель данных в виде DataSet;
3. создать саму БД.

## 2 Реализация

Предметная область – грузоперевозки (автотранспорт)

****

# Лабораторная работа 2

## 1 Задание

**«Разработка модуля DataAccessor»**

**Цель работы:**

приобрести навыки разработки слоя данных.

**Задачи:**

разработать и реализовать классы DataAccessor для имеющейся структуры БД.

## 2 Реализация

Все классы реализуют один интерфейс:

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

namespace lab\_2.Accessors

{

interface IDataAccessor

{

int Update(IDbConnection connection, IDbTransaction transaction,

DataSet dataSet, IDbDataAdapter dataAdapter);

int Read(IDbConnection connection, IDbTransaction transaction,

DataSet dataSet, IDbDataAdapter dataAdapter);

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Data;

namespace lab\_2.Accessors

{

public class CityDataAccessor : IDataAccessor

{

/// <summary>

/// Выгружает локальные данные на сервер, обновляет БД

/// </summary>

/// <param name="connection">соединение с базой</param>

/// <param name="transaction">транзакция</param>

/// <param name="dataSet">объект датасета</param>

/// <param name="dataAdapter"></param>

public int Update(IDbConnection connection, IDbTransaction transaction,

DataSet dataSet, IDbDataAdapter dataAdapter)

{

IDbDataParameter parameter;

//

//------------------------------------------------------

dataAdapter.DeleteCommand = connection.CreateCommand();

dataAdapter.DeleteCommand.CommandText =

@"DELETE FROM city4

WHERE pk\_city4=@pk\_city4";

#region DeleteQueryParametres

parameter = dataAdapter.DeleteCommand.CreateParameter();

parameter.ParameterName = "@pk\_city4";

parameter.SourceColumn = "pk\_city4";

dataAdapter.DeleteCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.DeleteCommand.Transaction = transaction;

//

//-------------------------------------------------

dataAdapter.UpdateCommand = connection.CreateCommand();

dataAdapter.UpdateCommand.CommandText =

@"UPDATE city4 SET cityname4=@cityname4

WHERE pk\_city4=@pk\_city4";

#region UpdateQueryParametres

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@pk\_city4";

parameter.SourceColumn = "pk\_city4";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@cityname4";

parameter.SourceColumn = "cityname4";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.UpdateCommand.Transaction = transaction;

//

//--------------------------------------------------

dataAdapter.InsertCommand = connection.CreateCommand();

dataAdapter.InsertCommand.CommandText =

@"INSERT INTO city4 (pk\_city4, cityname4)

VALUES (@pk\_city4, @cityname4)";

#region InsertQueryParametres

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@pk\_city4";

parameter.SourceColumn = "pk\_city4";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@cityname4";

parameter.SourceColumn = "cityname4";

dataAdapter.InsertCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.InsertCommand.Transaction = transaction;

//---------------------------------------------------------

dataAdapter.TableMappings.Add("Table", "CITY4");

return dataAdapter.Update(dataSet);

}

/// <summary>

///

/// </summary>

/// <param name="connection"></param>

/// <param name="transaction"></param>

/// <param name="dataTable"></param>

/// <param name="dataAdapter"></param>

/// <returns></returns>

public int Read(IDbConnection connection, IDbTransaction transaction,

DataSet dataSet, IDbDataAdapter dataAdapter)

{

dataAdapter.SelectCommand = connection.CreateCommand();

dataAdapter.SelectCommand.CommandText =

@"SELECT pk\_city4, cityname4 FROM city4";

dataAdapter.SelectCommand.Transaction = transaction;

dataAdapter.TableMappings.Add("Table", "CITY4");

return dataAdapter.Fill(dataSet);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Data;

namespace lab\_2.Accessors

{

public class ClientDataAccessor : IDataAccessor

{

/// <summary>

/// Выгружает локальные данные на сервер, обновляет БД

/// </summary>

/// <param name="connection">соединение с базой</param>

/// <param name="transaction">транзакция</param>

/// <param name="dataSet">объект датасета</param>

/// <param name="dataAdapter"></param>

public int Update(IDbConnection connection, IDbTransaction transaction,

DataSet dataSet, IDbDataAdapter dataAdapter)

{

IDbDataParameter parameter;

//

//------------------------------------------------------

dataAdapter.DeleteCommand = connection.CreateCommand();

dataAdapter.DeleteCommand.CommandText =

@"DELETE FROM client3

WHERE pk\_client3=@pk\_client3";

#region DeleteQueryParametres

parameter = dataAdapter.DeleteCommand.CreateParameter();

parameter.ParameterName = "@pk\_client3";

parameter.SourceColumn = "pk\_client3";

dataAdapter.DeleteCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.DeleteCommand.Transaction = transaction;

//

//-------------------------------------------------

dataAdapter.UpdateCommand = connection.CreateCommand();

dataAdapter.UpdateCommand.CommandText =

@"UPDATE client3

SET

city3=@city3, name3=@name3, address3=@address3, fio3=@fio3, phone3=@phone3

WHERE

pk\_client3=@pk\_client3";

#region UpdateQueryParametres

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@pk\_client3";

parameter.SourceColumn = "pk\_client3";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@name3";

parameter.SourceColumn = "name3";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@city3";

parameter.SourceColumn = "city3";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@address3";

parameter.SourceColumn = "address3";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@fio3";

parameter.SourceColumn = "fio3";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@phone3";

parameter.SourceColumn = "phone3";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.UpdateCommand.Transaction = transaction;

//

//--------------------------------------------------

dataAdapter.InsertCommand = connection.CreateCommand();

dataAdapter.InsertCommand.CommandText =

@"INSERT INTO client3

(pk\_client3, city3, name3, address3, fio3, phone3)

VALUES

(@pk\_client3, @city3, @name3, @address3, @fio3, @phone3)";

#region InsertQueryParametres

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@pk\_client3";

parameter.SourceColumn = "pk\_client3";

dataAdapter.InsertCommand.Parameters.Add(parameter);

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@city3";

parameter.SourceColumn = "city3";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@name3";

parameter.SourceColumn = "name3";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@address3";

parameter.SourceColumn = "address3";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@fio3";

parameter.SourceColumn = "fio3";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@phone3";

parameter.SourceColumn = "phone3";

dataAdapter.InsertCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.InsertCommand.Transaction = transaction;

//---------------------------------------------------------

dataAdapter.TableMappings.Add("Table", "CLIENT3");

return dataAdapter.Update(dataSet);

}

/// <summary>

///

/// </summary>

/// <param name="connection"></param>

/// <param name="transaction"></param>

/// <param name="dataTable"></param>

/// <param name="dataAdapter"></param>

/// <returns></returns>

public int Read(IDbConnection connection, IDbTransaction transaction,

DataSet dataSet, IDbDataAdapter dataAdapter)

{

dataAdapter.SelectCommand = connection.CreateCommand();

dataAdapter.SelectCommand.CommandText =

@"SELECT

pk\_client3, city3, name3, address3, fio3, phone3

FROM

client3";

dataAdapter.SelectCommand.Transaction = transaction;

dataAdapter.TableMappings.Add("Table", "CLIENT3");

return dataAdapter.Fill(dataSet);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace lab\_2.Accessors

{

class ClientViewDataAccessor : IDataAccessor

{

#region Члены IDataAccessor

public int Update(System.Data.IDbConnection connection, System.Data.IDbTransaction transaction, System.Data.DataSet dataSet, System.Data.IDbDataAdapter dataAdapter)

{

throw new NotImplementedException();

}

public int Read(System.Data.IDbConnection connection, System.Data.IDbTransaction transaction, System.Data.DataSet dataSet, System.Data.IDbDataAdapter dataAdapter)

{

dataAdapter.SelectCommand = connection.CreateCommand();

dataAdapter.SelectCommand.CommandText =

@"SELECT a.pk\_client3, a.city3, b.pk\_city4, b.cityname4, a.name3, a.address3, a.phone3, a.fio3

FROM client3 a, city4 b

WHERE a.city3=b.pk\_city4";

dataAdapter.SelectCommand.Transaction = transaction;

dataAdapter.TableMappings.Add("Table", "ClientView");

return dataAdapter.Fill(dataSet);

}

#endregion

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

namespace lab\_2.Accessors

{

public class MachineDataAccessor : IDataAccessor

{

/// <summary>

/// Выгружает локальные данные на сервер, обновляет БД

/// </summary>

/// <param name="connection">соединение с базой</param>

/// <param name="transaction">транзакция</param>

/// <param name="dataSet">объект датасета</param>

/// <param name="dataAdapter"></param>

public int Update(IDbConnection connection, IDbTransaction transaction,

DataSet dataSet, IDbDataAdapter dataAdapter)

{

IDbDataParameter parameter;

//

//------------------------------------------------------

dataAdapter.DeleteCommand = connection.CreateCommand();

dataAdapter.DeleteCommand.CommandText =

@"DELETE FROM

machine2

WHERE

(pk\_machine2 = @pk\_machine2)";

#region DeleteQueryParametres

parameter = dataAdapter.DeleteCommand.CreateParameter();

parameter.ParameterName = "@pk\_machine2";

parameter.SourceColumn = "pk\_machine2";

dataAdapter.DeleteCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.DeleteCommand.Transaction = transaction;

//

//-------------------------------------------------

dataAdapter.UpdateCommand = connection.CreateCommand();

dataAdapter.UpdateCommand.CommandText =

@"UPDATE machine2

SET city2 = @city2, max\_ves2 = @max\_ves2,

rezhim2 = @rezhim2, done2 = @done2, date2 = @date2

WHERE

pk\_machine2 = @pk\_machine2";

#region UpdateQueryParametres

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@pk\_machine2";

parameter.SourceColumn = "pk\_machine2";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@city2";

parameter.SourceColumn = "city2";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@max\_ves2";

parameter.SourceColumn = "max\_ves2";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@rezhim2";

parameter.SourceColumn = "rezhim2";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@done2";

parameter.SourceColumn = "done2";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@date2";

parameter.SourceColumn = "date2";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.UpdateCommand.Transaction = transaction;

//

//--------------------------------------------------

dataAdapter.InsertCommand = connection.CreateCommand();

dataAdapter.InsertCommand.CommandText =

@"INSERT INTO machine2

(pk\_machine2, city2, max\_ves2, rezhim2, done2, date2)

VALUES

(@pk\_machine2, @city2, @max\_ves2, @rezhim2, @done2, @date2)";

#region InsertQueryParametres

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@pk\_machine2";

parameter.SourceColumn = "pk\_machine2";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@city2";

parameter.SourceColumn = "city2";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@max\_ves2";

parameter.SourceColumn = "max\_ves2";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@rezhim2";

parameter.SourceColumn = "rezhim2";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@done2";

parameter.SourceColumn = "done2";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@date2";

parameter.SourceColumn = "date2";

dataAdapter.InsertCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.InsertCommand.Transaction = transaction;

//---------------------------------------------------------

dataAdapter.TableMappings.Add("Table", "MACHINE2");

return dataAdapter.Update(dataSet);

}

/// <summary>

///

/// </summary>

/// <param name="connection"></param>

/// <param name="transaction"></param>

/// <param name="dataTable"></param>

/// <param name="dataAdapter"></param>

/// <returns></returns>

public int Read(IDbConnection connection, IDbTransaction transaction,

DataSet dataSet, IDbDataAdapter dataAdapter)

{

dataAdapter.SelectCommand = connection.CreateCommand();

dataAdapter.SelectCommand.CommandText =

@"SELECT

pk\_machine2, city2, max\_ves2, rezhim2, done2, date2

FROM

machine2";

dataAdapter.SelectCommand.Transaction = transaction;

dataAdapter.TableMappings.Add("Table", "MACHINE2");

return dataAdapter.Fill(dataSet);

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

namespace lab\_2.Accessors

{

public class ZayavkaDataAccessor : IDataAccessor

{

/// <summary>

///

/// </summary>

/// <param name="connection">соединение с базой</param>

/// <param name="transaction">транзакция</param>

/// <param name="dataSet">объект датасета</param>

/// <param name="dataAdapter"></param>

public int Update(IDbConnection connection, IDbTransaction transaction,

DataSet dataSet, IDbDataAdapter dataAdapter)

{

IDbDataParameter parameter;

//

//------------------------------------------------------

dataAdapter.DeleteCommand = connection.CreateCommand();

dataAdapter.DeleteCommand.CommandText =

@"DELETE FROM zayavka1

WHERE pk\_zayavka1=@pk\_zayavka1";

#region DeleteQueryParametres

parameter = dataAdapter.DeleteCommand.CreateParameter();

parameter.ParameterName = "@pk\_zayavka1";

parameter.SourceColumn = "pk\_zayavka1";

dataAdapter.DeleteCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.DeleteCommand.Transaction = transaction;

//

//-------------------------------------------------

dataAdapter.UpdateCommand = connection.CreateCommand();

dataAdapter.UpdateCommand.CommandText =

@"UPDATE zayavka1 SET

otprav1 = @otprav1, poluch1 = @poluch1, machine1=@machine1, dsost1 = @dsost1,

gruz\_descr1 = @gruz\_descr1, ves1 = @ves1, places1 = @places1,

exped1 = @exped1, srocn1 = @srocn1, done1 = @done1, rezhim1 = @rezhim1

WHERE

pk\_zayavka1 = @pk\_zayavka1";

#region UpdateQueryParametres

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@pk\_zayavka1";

parameter.SourceColumn = "pk\_zayavka1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@otprav1";

parameter.SourceColumn = "otprav1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@poluch1";

parameter.SourceColumn = "poluch1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@machine1";

parameter.SourceColumn = "machine1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@dsost1";

parameter.SourceColumn = "dsost1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@gruz\_descr1";

parameter.SourceColumn = "gruz\_descr1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@ves1";

parameter.SourceColumn = "ves1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@places1";

parameter.SourceColumn = "places1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@exped1";

parameter.SourceColumn = "exped1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@srocn1";

parameter.SourceColumn = "srocn1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@done1";

parameter.SourceColumn = "done1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.UpdateCommand.CreateParameter();

parameter.ParameterName = "@rezhim1";

parameter.SourceColumn = "rezhim1";

dataAdapter.UpdateCommand.Parameters.Add(parameter);

//parameter = dataAdapter.UpdateCommand.CreateParameter();

//parameter.ParameterName = "@exr1";

//parameter.SourceColumn = "exr1";

//dataAdapter.UpdateCommand.Parameters.Add(parameter);

//parameter = dataAdapter.UpdateCommand.CreateParameter();

//parameter.ParameterName = "@exd1";

//parameter.SourceColumn = "exd1";

//dataAdapter.UpdateCommand.Parameters.Add(parameter);

#endregion

//;

dataAdapter.UpdateCommand.Transaction = transaction;

//

//--------------------------------------------------

dataAdapter.InsertCommand = connection.CreateCommand();

dataAdapter.InsertCommand.CommandText =

@"INSERT INTO zayavka1

(pk\_zayavka1, otprav1, poluch1, machine1, dsost1, gruz\_descr1,

ves1, places1, exped1, srocn1, done1, rezhim1)

VALUES (@pk\_zayavka1, @otprav1, @poluch1, @machine1, @dsost1, @gruz\_descr1,

@ves1, @places1, @exped1, @srocn1, @done1, @rezhim1)";

#region InsertQueryParametres

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@pk\_zayavka1";

parameter.SourceColumn = "pk\_zayavka1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@otprav1";

parameter.SourceColumn = "otprav1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@poluch1";

parameter.SourceColumn = "poluch1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@machine1";

parameter.SourceColumn = "machine1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@dsost1";

parameter.SourceColumn = "dsost1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@gruz\_descr1";

parameter.SourceColumn = "gruz\_descr1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@ves1";

parameter.SourceColumn = "ves1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@places1";

parameter.SourceColumn = "places1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@exped1";

parameter.SourceColumn = "exped1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@srocn1";

parameter.SourceColumn = "srocn1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@done1";

parameter.SourceColumn = "done1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

//;

parameter = dataAdapter.InsertCommand.CreateParameter();

parameter.ParameterName = "@rezhim1";

parameter.SourceColumn = "rezhim1";

dataAdapter.InsertCommand.Parameters.Add(parameter);

#endregion //;

dataAdapter.InsertCommand.Transaction = transaction;

//---------------------------------------------------------

dataAdapter.TableMappings.Add("Table", "ZAYAVKA1");

return dataAdapter.Update(dataSet);

}

/// <summary>

///

/// </summary>

/// <param name="connection"></param>

/// <param name="transaction"></param>

/// <param name="dataTable"></param>

/// <param name="dataAdapter"></param>

/// <returns></returns>

public int Read(IDbConnection connection, IDbTransaction transaction,

DataSet dataSet, IDbDataAdapter dataAdapter)

{

dataAdapter.SelectCommand = connection.CreateCommand();

dataAdapter.SelectCommand.CommandText =

@"SELECT

pk\_zayavka1, otprav1, poluch1, machine1, dsost1, gruz\_descr1,

ves1, places1, exped1, srocn1, done1, rezhim1

FROM

zayavka1";

dataAdapter.SelectCommand.Transaction = transaction;

dataAdapter.TableMappings.Add("Table", "ZAYAVKA1");

return dataAdapter.Fill(dataSet);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace lab\_2.Accessors

{

class ZayavkaViewDataAccessor : IDataAccessor

{

#region Члены IDataAccessor

public int Update(System.Data.IDbConnection connection, System.Data.IDbTransaction transaction, System.Data.DataSet dataSet, System.Data.IDbDataAdapter dataAdapter)

{

throw new NotImplementedException();

}

public int Read(System.Data.IDbConnection connection, System.Data.IDbTransaction transaction, System.Data.DataSet dataSet, System.Data.IDbDataAdapter dataAdapter)

{

dataAdapter.SelectCommand = connection.CreateCommand();

dataAdapter.SelectCommand.CommandText =

@"SELECT c.pk\_zayavka1, c.otprav1, a.pk\_client3, a.name3, c.poluch1, b.pk\_client3 AS Expr1, b.name3 AS Expr2, c.machine1, c.dsost1, c.gruz\_descr1, c.ves1,

c.places1, c.exped1, c.srocn1, c.done1, c.rezhim1

FROM CLIENT3 a, CLIENT3 b, ZAYAVKA1 c

where a.pk\_client3 = c.otprav1 and b.pk\_client3 = c.poluch1";

dataAdapter.SelectCommand.Transaction = transaction;

dataAdapter.TableMappings.Add("Table", "ZView");

return dataAdapter.Fill(dataSet);

}

#endregion

}

}

# Лабораторная работа 3

## 1 Задание

**«Юнит-тесты для модуля DataAccessor»**

**Цель работы:**

освоить технологию юнит-тестов.

**Задачи:**

для имеющегося слоя данных разработать юнит-тесты. Юнит-тесты должны покрывать основные крайние случаи. В общем, на каждый метод слоя данных должно быть не менее 4-х тестов.

## 2 Реализация

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using lab\_2.Accessors;

using lab\_2.Logging;

using lab\_2;

using lab\_2.BLogic;

using Oracle;

using OracleClient = Oracle.DataAccess.Client;

using NUnit.Framework;

using Spring.Core;

using Spring.Context;

using Spring.Context.Support;

namespace tst

{

[TestFixture]

class CityTester

{

IApplicationContext context;

[SetUp]

public void Init()

{

context = new XmlApplicationContext("context.xml");

}

[TearDown]

public void Exit()

{

context.Dispose();

}

///\*

[Test]//модификатор метода, который делает его видимым для nunit. метод помечается как тестирующий

public void Test\_001\_InsertUpdate()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

dataSet1.CITY4.AddCITY4Row( "Барнаул");

dataSet1.CITY4.AddCITY4Row( "Абакан");

dataSet1.CITY4.AddCITY4Row( "Красноярск");

dataSet1.CITY4.AddCITY4Row( "Улан-Удэ");

//проинициализируем соединение с БД

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

//откроем соединение

connection.Open();

//запустим новую транзакцию

OracleClient.OracleTransaction transaction = connection.BeginTransaction();

lab\_2.Accessors.CityDataAccessor dataAccessor = (CityDataAccessor)context.GetObject("CITY4");

//сольем то, что добавили

dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

//утвердим изменения

transaction.Commit();

//начнем новую транзакцию

transaction = connection.BeginTransaction();

//в новый датасет

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

//вытянем содержимое таблицы

dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

//закончим транзакцию

transaction.Commit();

//закроем соединение

connection.Close();

//проверим идентичность датасетов

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]//модификатор метода, который делает его видимым для nunit. метод помечается как тестирующий

public void Test\_1\_InsertUpdate()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

dataSet1.CITY4.AddCITY4Row( "Барнаул");

dataSet1.CITY4.AddCITY4Row( "Абакан");

dataSet1.CITY4.AddCITY4Row( "Красноярск");

dataSet1.CITY4.AddCITY4Row( "Улан-Удэ");

//проинициализируем соединение с БД

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

//откроем соединение

connection.Open();

//запустим новую транзакцию

OracleClient.OracleTransaction transaction = connection.BeginTransaction();

lab\_2.Accessors.CityDataAccessor dataAccessor = new CityDataAccessor();

//сольем то, что добавили

dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

//утвердим изменения

transaction.Commit();

//начнем новую транзакцию

transaction = connection.BeginTransaction();

//в новый датасет

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

//вытянем содержимое таблицы

dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

//закончим транзакцию

transaction.Commit();

//закроем соединение

connection.Close();

//проверим идентичность датасетов

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_2\_DeleteUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

connection.Open();

OracleClient.OracleTransaction transaction;

transaction = connection.BeginTransaction();

lab\_2.Accessors.CityDataAccessor dataAccessor = new CityDataAccessor();

dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

for (int i = 0; i < dataSet1.CITY4.Rows.Count;i++)

dataSet1.CITY4.Rows[i].Delete();

dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

transaction.Commit();

connection.Close();

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_3\_UpdateUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

connection.Open();

OracleClient.OracleTransaction transaction;

transaction = connection.BeginTransaction();

lab\_2.Accessors.CityDataAccessor dataAccessor = new CityDataAccessor();

dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

for (int i = 0; i < dataSet1.CITY4.Rows.Count; i++)

{

if (dataSet1.CITY4.Rows[i].ItemArray[1].ToString().Equals("Барнаул"))

{

dataSet1.CITY4.Rows[i].ItemArray[1] = "Барнаул-1";

break;

}

}

transaction = connection.BeginTransaction();

dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

transaction.Commit();

connection.Close();

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_4\_UpdateEmpty()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

connection.Open();

OracleClient.OracleTransaction transaction = connection.BeginTransaction();

lab\_2.Accessors.CityDataAccessor dataAccessor = new CityDataAccessor();

dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

transaction.Commit();

connection.Close();

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

[ExpectedException(typeof(Oracle.DataAccess.Client.OracleException))]

public void Test\_5\_InsertDuplicate()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

//dataSet1.CITY4.AddCITY4Row(1, "Барнаул");

dataSet1.CITY4.AddCITY4Row( "Барнаул-1");

dataSet1.CITY4.AddCITY4Row( "Абакан");

dataSet1.CITY4.AddCITY4Row( "Красноярск");

dataSet1.CITY4.AddCITY4Row( "Улан-Удэ");

//проинициализируем соединение с БД

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

//откроем соединение

connection.Open();

//запустим новую транзакцию

OracleClient.OracleTransaction transaction = connection.BeginTransaction();

lab\_2.Accessors.CityDataAccessor dataAccessor = new CityDataAccessor();

//сольем то, что добавили

dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

//утвердим изменения

transaction.Commit();

//начнем новую транзакцию

transaction = connection.BeginTransaction();

//в новый датасет

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

//вытянем содержимое таблицы

dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

//закончим транзакцию

transaction.Commit();

//закроем соединение

connection.Close();

//проверим идентичность датасетов

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

\*/

public bool CheckIdent(lab\_2.baykalDataSet dataSet1, lab\_2.baykalDataSet dataSet2)

{

if (dataSet1.CITY4.Count != dataSet2.CITY4.Count) return false;

bool flag = true;

for (int index = 0; index < dataSet1.CITY4.Count; index++)

{

flag = false;

for(int j=0;j<dataSet1.CITY4.Count;j++)

{

if (dataSet1.CITY4.Rows[index].ItemArray[0].Equals(dataSet2.CITY4.Rows[j].ItemArray[0])

&& String.Compare(dataSet1.CITY4.Rows[index].ItemArray[1].ToString(),

dataSet2.CITY4.Rows[j].ItemArray[1].ToString(), false) == 0

)

flag = true;

if (flag) break;

}

if (!flag) return false;

}

return true;

}

}

}

using System;

using System.Data;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using lab\_2.Accessors;

using lab\_2.Logging;

using lab\_2;

using Oracle;

using OracleClient = Oracle.DataAccess.Client;

using NUnit.Framework;

namespace tst

{

class ClientTester

{

///\*

[Test]

[Timeout(2000)]

public void Test\_1\_InsertUpdate()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

dataSet1.CITY4.AddCITY4Row("Барнаул");

dataSet1.CITY4.AddCITY4Row("Абакан");

dataSet1.CITY4.AddCITY4Row("Красноярск");

dataSet1.CITY4.AddCITY4Row("Улан-Удэ");

dataSet1.CLIENT3.AddCLIENT3Row( dataSet1.CITY4.ElementAt(0), "3333333", "fio1", "address1", "name1");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(1), "555-555-555", "fio2", "address2", "name2");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(1), "555-555-555", "fio3", "address3", "name3");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(0), "555-555-555", "fio4", "address4", "name4");

//проинициализируем соединение с БД

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

//откроем соединение

connection.Open();

//запустим новую транзакцию

OracleClient.OracleTransaction transaction = connection.BeginTransaction();

lab\_2.Accessors.CityDataAccessor city\_dataAccessor = new CityDataAccessor();

lab\_2.Accessors.ClientDataAccessor client\_dataAccessor = new ClientDataAccessor();

//сольем то, что добавили

city\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

city\_dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

//утвердим изменения

transaction.Commit();

//начнем новую транзакцию

transaction = connection.BeginTransaction();

//в новый датасет

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

//вытянем содержимое таблицы

city\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

//закончим транзакцию

transaction.Commit();

//закроем соединение

connection.Close();

//проверим идентичность датасетов

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_2\_DeleteUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

connection.Open();

OracleClient.OracleTransaction transaction;

transaction = connection.BeginTransaction();

lab\_2.Accessors.ClientDataAccessor dataAccessor = new ClientDataAccessor();

lab\_2.Accessors.CityDataAccessor cdataAccessor = new CityDataAccessor();

dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

cdataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

for (int i = 0; i < dataSet1.CLIENT3.Rows.Count; i++)

dataSet1.CLIENT3.Rows[i].Delete();

for (int i = 0; i < dataSet1.CITY4.Rows.Count; i++)

dataSet1.CITY4.Rows[i].Delete();

dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

cdataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

cdataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

transaction.Commit();

connection.Close();

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_3\_UpdateUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

connection.Open();

OracleClient.OracleTransaction transaction;

transaction = connection.BeginTransaction();

lab\_2.Accessors.ClientDataAccessor dataAccessor = new ClientDataAccessor();

dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

for (int i = 0; i < dataSet1.CLIENT3.Rows.Count; i++)

{

if (dataSet1.CLIENT3.Rows[i].ItemArray[0].Equals((Object)1))

{

dataSet1.CLIENT3.Rows[i].ItemArray[2] = "Барнаул-1";

break;

}

}

dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

transaction.Commit();

connection.Close();

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

[ExpectedException(typeof(OracleClient.OracleException))]

public void Test\_4\_WhenTryToDeleteParent()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

connection.Open();

OracleClient.OracleTransaction transaction;

transaction = connection.BeginTransaction();

lab\_2.Accessors.ClientDataAccessor client\_dataAccessor = new ClientDataAccessor();

lab\_2.Accessors.CityDataAccessor city\_dataAccessor = new CityDataAccessor();

client\_dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

city\_dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

decimal index = (decimal)dataSet1.CLIENT3.Rows[0].ItemArray[1];

dataSet1.CITY4.Rows[(int)index].Delete();

city\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

client\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

city\_dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

connection.Close();

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

\* \*/

public bool CheckIdent(lab\_2.baykalDataSet dataSet1, lab\_2.baykalDataSet dataSet2)

{

if (dataSet1.CLIENT3.Count != dataSet2.CLIENT3.Count) return false;

bool flag = true;

for (int i = 0; i < dataSet1.CLIENT3.Count; i++)

{

flag = false;

for (int j = 0; j < dataSet1.CLIENT3.Count; j++)

{

flag = true;

for (int k = 0; k < dataSet1.CLIENT3.Columns.Count; k++)

if (!dataSet1.CLIENT3.Rows[i].ItemArray[k].ToString().

Equals(dataSet2.CLIENT3.Rows[j].ItemArray[k].ToString()))

flag = false;

if (flag) break;

}

if (!flag) return false;

}

return true;

}

}

}

using System;

using System.Data;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using lab\_2.Accessors;

using lab\_2.Logging;

using lab\_2.BLogic;

using lab\_2;

using Oracle;

using OracleClient = Oracle.DataAccess.Client;

using NUnit.Framework;

namespace tst

{

class ZayavkaTester

{

///\*

[Test]

[Timeout(2000)]

public void Test\_1\_InsertUpdate()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

dataSet1.CITY4.AddCITY4Row( "Барнаул");

dataSet1.CITY4.AddCITY4Row( "Абакан");

dataSet1.CITY4.AddCITY4Row( "Красноярск");

dataSet1.CITY4.AddCITY4Row( "Улан-Удэ");

dataSet1.CLIENT3.AddCLIENT3Row( dataSet1.CITY4.ElementAt(0), "3333333", "fio1", "address1", "name1");

dataSet1.CLIENT3.AddCLIENT3Row( dataSet1.CITY4.ElementAt(1), "555-555-555", "fio2", "address2", "name2");

dataSet1.CLIENT3.AddCLIENT3Row( dataSet1.CITY4.ElementAt(1), "555-555-555", "fio3", "address3", "name3");

dataSet1.CLIENT3.AddCLIENT3Row( dataSet1.CITY4.ElementAt(0), "555-555-555", "fio4", "address4", "name4");

dataSet1.ZAYAVKA1.AddZAYAVKA1Row(

dataSet1.CLIENT3.ElementAt(0),

dataSet1.CLIENT3.ElementAt(1),

null,

DateTime.Today,

"cheeeez",

(decimal)10,(decimal)3,(decimal)0,(decimal)0,(decimal)0,(decimal)0,(decimal)0,(decimal)0);

dataSet1.ZAYAVKA1.AddZAYAVKA1Row(

dataSet1.CLIENT3.ElementAt(2),

dataSet1.CLIENT3.ElementAt(3),

null,

DateTime.Today,

"cheeeez",

(decimal)10, (decimal)3, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0);

//проинициализируем соединение с БД

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

//откроем соединение

connection.Open();

//запустим новую транзакцию

OracleClient.OracleTransaction transaction = connection.BeginTransaction();

lab\_2.Accessors.CityDataAccessor city\_dataAccessor = new CityDataAccessor();

lab\_2.Accessors.ClientDataAccessor client\_dataAccessor = new ClientDataAccessor();

lab\_2.Accessors.ZayavkaDataAccessor zayavka\_dataAccessor = new ZayavkaDataAccessor();

//сольем то, что добавили

city\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

zayavka\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

//утвердим изменения

transaction.Commit();

//начнем новую транзакцию

transaction = connection.BeginTransaction();

//в новый датасет

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

//вытянем содержимое таблицы

city\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

zayavka\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

//закончим транзакцию

transaction.Commit();

//закроем соединение

connection.Close();

//проверим идентичность датасетов

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_2\_DeleteUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

connection.Open();

OracleClient.OracleTransaction transaction;

transaction = connection.BeginTransaction();

lab\_2.Accessors.MachineDataAccessor mda = new MachineDataAccessor();

lab\_2.Accessors.ZayavkaDataAccessor zayavka\_dataAccessor = new ZayavkaDataAccessor();

lab\_2.Accessors.ClientDataAccessor client\_dataAccessor = new ClientDataAccessor();

lab\_2.Accessors.CityDataAccessor city\_dataAccessor = new CityDataAccessor();

city\_dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

zayavka\_dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

for (int i = 0; i < dataSet1.ZAYAVKA1.Rows.Count; i++)

dataSet1.ZAYAVKA1.Rows[i].Delete();

for (int i = 0; i < dataSet1.CLIENT3.Rows.Count; i++)

dataSet1.CLIENT3.Rows[i].Delete();

for (int i = 0; i < dataSet1.CITY4.Rows.Count; i++)

dataSet1.CITY4.Rows[i].Delete();

zayavka\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

city\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

mda.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

zayavka\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

city\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

transaction.Commit();

connection.Close();

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_3\_UpdateUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

connection.Open();

OracleClient.OracleTransaction transaction;

transaction = connection.BeginTransaction();

lab\_2.Accessors.ZayavkaDataAccessor dataAccessor = new ZayavkaDataAccessor();

dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

for (int i = 0; i < dataSet1.ZAYAVKA1.Rows.Count; i++)

{

if (dataSet1.ZAYAVKA1.Rows[i].ItemArray[0].Equals((Object)1))

{

dataSet1.ZAYAVKA1.Rows[i].ItemArray[6] = (decimal)20;

break;

}

}

dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

transaction.Commit();

connection.Close();

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

[ExpectedException(typeof(OracleClient.OracleException))]

public void Test\_4\_WhenTryToDeleteParent()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

connection.Open();

OracleClient.OracleTransaction transaction;

transaction = connection.BeginTransaction();

lab\_2.Accessors.ClientDataAccessor client\_dataAccessor = new ClientDataAccessor();

lab\_2.Accessors.ZayavkaDataAccessor zayavka\_dataAccessor = new ZayavkaDataAccessor();

client\_dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

zayavka\_dataAccessor.Read(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

decimal index = (decimal)dataSet1.ZAYAVKA1.Rows[0].ItemArray[1];

dataSet1.CLIENT3.Rows[(int)index].Delete();

client\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

zayavka\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

transaction.Commit();

transaction = connection.BeginTransaction();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

zayavka\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

transaction.Commit();

connection.Close();

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

[ExpectedException(typeof(System.Data.NoNullAllowedException))]

public void Test\_5\_AddNotExistingParent()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

dataSet1.CITY4.AddCITY4Row("Барнаул");

dataSet1.CITY4.AddCITY4Row("Абакан");

dataSet1.CITY4.AddCITY4Row("Красноярск");

dataSet1.CITY4.AddCITY4Row("Улан-Удэ");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(0), "3333333", "fio1", "address1", "name1");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(1), "555-555-555", "fio2", "address2", "name2");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(1), "555-555-555", "fio3", "address3", "name3");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(0), "555-555-555", "fio4", "address4", "name4");

dataSet1.ZAYAVKA1.AddZAYAVKA1Row(

dataSet1.CLIENT3.ElementAt(0),

dataSet1.CLIENT3.ElementAt(1),

null,

DateTime.Today,

"cheeeez",

(decimal)10, (decimal)3, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0);

dataSet1.ZAYAVKA1.AddZAYAVKA1Row(

dataSet1.CLIENT3.ElementAt(2),

dataSet1.CLIENT3.ElementAt(3),

null,

DateTime.Today,

"cheeeez",

(decimal)10, (decimal)3, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0);

//проинициализируем соединение с БД

OracleClient.OracleConnection connection =

new Oracle.DataAccess.Client.OracleConnection(@"Data Source=localhost;Persist Security Info=True;User ID=sk\_admin;Password=sksk;");

//откроем соединение

connection.Open();

//запустим новую транзакцию

OracleClient.OracleTransaction transaction = connection.BeginTransaction();

lab\_2.Accessors.CityDataAccessor city\_dataAccessor = new CityDataAccessor();

lab\_2.Accessors.ClientDataAccessor client\_dataAccessor = new ClientDataAccessor();

lab\_2.Accessors.ZayavkaDataAccessor zayavka\_dataAccessor = new ZayavkaDataAccessor();

//сольем то, что добавили

city\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

zayavka\_dataAccessor.Update(connection, transaction, dataSet1, new OracleClient.OracleDataAdapter());

//утвердим изменения

transaction.Commit();

//начнем новую транзакцию

transaction = connection.BeginTransaction();

//в новый датасет

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

//вытянем содержимое таблицы

city\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

client\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

zayavka\_dataAccessor.Read(connection, transaction, dataSet2, new OracleClient.OracleDataAdapter());

//закончим транзакцию

transaction.Commit();

//закроем соединение

connection.Close();

//проверим идентичность датасетов

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

public bool CheckIdent(lab\_2.baykalDataSet dataSet1, lab\_2.baykalDataSet dataSet2)

{

if (dataSet1.ZAYAVKA1.Count != dataSet2.ZAYAVKA1.Count) return false;

bool flag = true;

for (int i = 0; i < dataSet1.ZAYAVKA1.Count; i++)

{

flag = false;

for (int j = 0; j < dataSet1.ZAYAVKA1.Count; j++)

{

flag = true;

for (int k = 0; k < dataSet1.ZAYAVKA1.Columns.Count; k++)

if (!dataSet1.ZAYAVKA1.Rows[i].ItemArray[k].ToString().

Equals(dataSet2.ZAYAVKA1.Rows[j].ItemArray[k].ToString()))

flag = false;

if (flag) break;

}

if (!flag) return false;

}

return true;

}

\* \*/

}

}

# Лабораторная работа 4

## 1 Задание

**«Разработка модуля BusinessLogic»**

**Цель работы:**

приобрести навыки разработки слоя бизнес-логики.

**Задачи:**

разработать и реализовать слой бизнес-логики (в виде модели реальной программной системы, т.к. в рамках лабораторных работ сложно создать ПО с реальной бизнес-логикой) в соответствии со спецификой использования DataSet в качестве модели данных.

## 2 Реализация

using System;

namespace lab\_2.BLogic

{

public interface IBusinessLogic

{

lab\_2.baykalDataSet AssociatedDataset { get; set; }

lab\_2.baykalDataSet FillAll(lab\_2.baykalDataSet dataSet);

int FillCityTable(lab\_2.baykalDataSet dataSet);

int FillClientTable(lab\_2.baykalDataSet dataSet);

int FillClientView(lab\_2.baykalDataSet dataSet);

int FillMachineTable(lab\_2.baykalDataSet dataSet);

int FillZayavkaTable(lab\_2.baykalDataSet dataSet);

int FillZayavkaView(baykalDataSet dataSet);

lab\_2.baykalDataSet ReadAll();

lab\_2.baykalDataSet ReadCityTable();

lab\_2.baykalDataSet ReadClientTable();

lab\_2.baykalDataSet ReadMachineTable();

lab\_2.baykalDataSet ReadZayavkaTable();

void UpdateAll(lab\_2.baykalDataSet dataSet);

int UpdateCityTable(lab\_2.baykalDataSet dataSet);

int UpdateClientTable(lab\_2.baykalDataSet dataSet);

int UpdateClientView(lab\_2.baykalDataSet dataSet);

int UpdateMachineTable(lab\_2.baykalDataSet dataSet);

int UpdateZayavkaTable(lab\_2.baykalDataSet dataSet);

int UpdateZayavkaView(baykalDataSet dataSet);

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

//using Oracle;

//using OracleClient = Oracle.DataAccess.Client;

using lab\_2.Accessors;

using lab\_2.Logging;

using System.Data.SqlClient;

namespace lab\_2.BLogic

{

public class BusinessLogic : lab\_2.BLogic.IBusinessLogic

{

//static lab\_2.Properties.Settings s = new lab\_2.Properties.Settings();

static Logger logger = new Logger("d:\\sss.txt", (ushort)0);

static String ConnectionString = @"Data Source=book\sqlexpress8;Initial Catalog=dbb;Integrated Security=True;Pooling=False";

public baykalDataSet AssociatedDataset

{

get;

set;

}

#region Члены IBusinessLogic

public baykalDataSet FillAll(baykalDataSet dataSet)

{

AssociatedDataset = dataSet;

return ReadAll();

}

public int FillCityTable(baykalDataSet dataSet)

{

int readString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

readString =

(new CityDataAccessor()).Read(connection,transaction,AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return readString;

//return readString;

}

public int FillClientTable(baykalDataSet dataSet)

{

int readString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

readString =

(new ClientDataAccessor()).Read(connection,transaction,AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return readString;

}

public int FillClientView(baykalDataSet dataSet)

{

int readString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

readString =

(new ClientViewDataAccessor()).Read(connection,transaction,AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return readString;

}

public int FillMachineTable(baykalDataSet dataSet)

{

int readString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

readString =

(new MachineDataAccessor()).Read(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return readString;

}

public int FillZayavkaTable(baykalDataSet dataSet)

{

int readString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

readString =

(new ZayavkaDataAccessor()).Read(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return readString;

}

public int FillZayavkaView(baykalDataSet dataSet)

{

int readString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

readString =

(new ZayavkaDataAccessor()).Read(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return readString;

//return readString;

}

public baykalDataSet ReadAll()

{

//baykalDataSet ds = new baykalDataSet();

if (AssociatedDataset == null)

AssociatedDataset = new baykalDataSet();

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

(new CityDataAccessor()).Read(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

(new ClientDataAccessor()).Read(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

(new MachineDataAccessor()).Read(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

(new ZayavkaDataAccessor()).Read(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

(new ZayavkaViewDataAccessor()).Read(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

(new ClientViewDataAccessor()).Read(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return AssociatedDataset;

}

public baykalDataSet ReadCityTable()

{

if (AssociatedDataset == null)

AssociatedDataset = new baykalDataSet();

FillCityTable(AssociatedDataset);

return AssociatedDataset;

}

public baykalDataSet ReadClientTable()

{

if (AssociatedDataset == null)

AssociatedDataset = new baykalDataSet();

FillClientTable(AssociatedDataset);

return AssociatedDataset;

}

public baykalDataSet ReadMachineTable()

{

if (AssociatedDataset == null)

AssociatedDataset = new baykalDataSet();

FillMachineTable(AssociatedDataset);

return AssociatedDataset;

}

public baykalDataSet ReadZayavkaTable()

{

if (AssociatedDataset == null)

AssociatedDataset = new baykalDataSet();

FillZayavkaTable(AssociatedDataset);

return AssociatedDataset;

}

public void UpdateAll(baykalDataSet dataSet)

{

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

(new CityDataAccessor()).Update(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

(new ClientDataAccessor()).Update(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

(new MachineDataAccessor()).Update(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

(new ZayavkaDataAccessor()).Update(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

}

public int UpdateCityTable(baykalDataSet dataSet)

{

int UpdateString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

UpdateString =

(new CityDataAccessor()).Update(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return UpdateString;

//return UpdateString;

}

public int UpdateClientTable(baykalDataSet dataSet)

{

int UpdateString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

UpdateString =

(new ClientDataAccessor()).Update(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return UpdateString;

}

public int UpdateClientView(baykalDataSet dataSet)

{

int UpdateString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

UpdateString =

(new ClientViewDataAccessor()).Update(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return UpdateString;

}

public int UpdateMachineTable(baykalDataSet dataSet)

{

int UpdateString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

UpdateString =

(new MachineDataAccessor()).Update(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return UpdateString;

}

public int UpdateZayavkaTable(baykalDataSet dataSet)

{

int UpdateString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

UpdateString =

(new ZayavkaDataAccessor()).Update(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return UpdateString;

}

public int UpdateZayavkaView(baykalDataSet dataSet)

{

int UpdateString = 0;

AssociatedDataset = (baykalDataSet)dataSet;

using (SqlConnection connection = new SqlConnection(ConnectionString))

{

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

try

{

UpdateString =

(new ZayavkaViewDataAccessor()).Update(connection, transaction, AssociatedDataset,

new SqlDataAdapter());

transaction.Commit();

}

catch (Exception e)

{

logger.Log(this.ToString(), e.Message, 0);

transaction.Rollback();

throw;

}

connection.Close();

}

return UpdateString;

}

#endregion

}

}

# Лабораторная работа 5

## 1 Задание

**«Юнит-тесты для модуля Business Logic»**

**Цель работы:**

закрепить навыки юнит- тестирования.

**Задачи:**

для имеющегося слоя бизнес-логики разработать юнит-тесты. Юнит-тесты должны покрывать основные крайние случаи. В общем, на каждый метод слоя бизнес-логики должно быть не менее 4-х тестов.

## 2 Реализация

Конфигурация Spring

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using lab\_2.Accessors;

using lab\_2.Logging;

using lab\_2;

using lab\_2.BLogic;

using Oracle;

using OracleClient = Oracle.DataAccess.Client;

using NUnit.Framework;

using Spring.Core;

using Spring.Context;

using Spring.Context.Support;

namespace tst

{

[TestFixture]

class CityTester

{

IApplicationContext context;

[SetUp]

public void Init()

{

context = new XmlApplicationContext("context.xml");

}

[TearDown]

public void Exit()

{

context.Dispose();

}

///\*

[Test]

public void Test\_BL01\_UpdateAndReadCity()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

BLogic.FillCityTable(dataSet1);

dataSet1.CITY4.AddCITY4Row( "Барнаул");

dataSet1.CITY4.AddCITY4Row( "Абакан");

dataSet1.CITY4.AddCITY4Row( "Красноярск");

dataSet1.CITY4.AddCITY4Row( "Улан-Удэ");

BLogic.UpdateCityTable(dataSet1);

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

BLogic.FillCityTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_BL02\_DeleteUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

BLogic.FillCityTable(dataSet1);

for (int i = 0; i < dataSet1.CITY4.Rows.Count; i++)

dataSet1.CITY4.Rows[i].Delete();

BLogic.UpdateCityTable(dataSet1);

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

BLogic.FillCityTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_BL03\_UpdateUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

BLogic.FillCityTable(dataSet1);

for (int i = 0; i < dataSet1.CITY4.Rows.Count; i++)

{

if (dataSet1.CITY4.Rows[i].ItemArray[1].ToString().Equals("Барнаул"))

{

dataSet1.CITY4.Rows[i].ItemArray[1] = "Барнаул-1";

break;

}

}

BLogic.UpdateCityTable(dataSet1);

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

BLogic.FillCityTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[TestCase(Result=0)]

public int Test\_BL04\_InsertDuplicate()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

BLogic.FillCityTable(dataSet1);

//dataSet1.CITY4.AddCITY4Row(1, "Барнаул");

dataSet1.CITY4.AddCITY4Row("Барнаул-1");

dataSet1.CITY4.AddCITY4Row( "Абакан");

dataSet1.CITY4.AddCITY4Row( "Красноярск");

dataSet1.CITY4.AddCITY4Row( "Улан-Удэ");

return 0;

//return BLogic.UpdateCityTable(dataSet1);

}

public bool CheckIdent(lab\_2.baykalDataSet dataSet1, lab\_2.baykalDataSet dataSet2)

{

if (dataSet1.CITY4.Count != dataSet2.CITY4.Count) return false;

bool flag = true;

for (int index = 0; index < dataSet1.CITY4.Count; index++)

{

flag = false;

for(int j=0;j<dataSet1.CITY4.Count;j++)

{

if (dataSet1.CITY4.Rows[index].ItemArray[0].Equals(dataSet2.CITY4.Rows[j].ItemArray[0])

&& String.Compare(dataSet1.CITY4.Rows[index].ItemArray[1].ToString(),

dataSet2.CITY4.Rows[j].ItemArray[1].ToString(), false) == 0

)

flag = true;

if (flag) break;

}

if (!flag) return false;

}

return true;

}

}

}

using System;

using System.Data;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using lab\_2.Accessors;

using lab\_2.Logging;

using lab\_2;

using Oracle;

using OracleClient = Oracle.DataAccess.Client;

using NUnit.Framework;

namespace tst

{

class ClientTester

{

[Test]

[Timeout(2000)]

public void Test\_BL01\_InsertUpdate()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new lab\_2.BLogic.BusinessLogic();

BLogic.FillClientTable(dataSet1);

dataSet1.CITY4.AddCITY4Row("Барнаул");

dataSet1.CITY4.AddCITY4Row("Абакан");

dataSet1.CITY4.AddCITY4Row( "Красноярск");

dataSet1.CITY4.AddCITY4Row("Улан-Удэ");

dataSet1.CLIENT3.AddCLIENT3Row( dataSet1.CITY4.ElementAt(0), "3333333", "fio1", "address1", "name1");

dataSet1.CLIENT3.AddCLIENT3Row( dataSet1.CITY4.ElementAt(1), "555-555-555", "fio2", "address2", "name2");

dataSet1.CLIENT3.AddCLIENT3Row( dataSet1.CITY4.ElementAt(1), "555-555-555", "fio3", "address3", "name3");

dataSet1.CLIENT3.AddCLIENT3Row( dataSet1.CITY4.ElementAt(0), "555-555-555", "fio4", "address4", "name4");

BLogic.UpdateCityTable(dataSet1);

BLogic.UpdateClientTable(dataSet1);

BLogic.FillCityTable(dataSet2);

BLogic.FillClientTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_BL02\_DeleteUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new lab\_2.BLogic.BusinessLogic();

BLogic.FillCityTable(dataSet1);

BLogic.FillClientTable(dataSet1);

for (int i = 0; i < dataSet1.CLIENT3.Rows.Count; i++)

dataSet1.CLIENT3.Rows[i].Delete();

for (int i = 0; i < dataSet1.CITY4.Rows.Count; i++)

dataSet1.CITY4.Rows[i].Delete();

BLogic.UpdateCityTable(dataSet1);

BLogic.UpdateClientTable(dataSet1);

BLogic.FillCityTable(dataSet2);

BLogic.FillClientTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_BL03\_UpdateUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new lab\_2.BLogic.BusinessLogic();

BLogic.FillCityTable(dataSet1);

BLogic.FillClientTable(dataSet1);

for (int i = 0; i < dataSet1.CLIENT3.Rows.Count; i++)

{

if (dataSet1.CLIENT3.Rows[i].ItemArray[0].Equals((Object)1))

{

dataSet1.CLIENT3.Rows[i].ItemArray[2] = "Барнаул-1";

break;

}

}

BLogic.UpdateCityTable(dataSet1);

BLogic.UpdateClientTable(dataSet1);

BLogic.FillCityTable(dataSet2);

BLogic.FillClientTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

public bool CheckIdent(lab\_2.baykalDataSet dataSet1, lab\_2.baykalDataSet dataSet2)

{

if (dataSet1.CLIENT3.Count != dataSet2.CLIENT3.Count) return false;

bool flag = true;

for (int i = 0; i < dataSet1.CLIENT3.Count; i++)

{

flag = false;

for (int j = 0; j < dataSet1.CLIENT3.Count; j++)

{

flag = true;

for (int k = 0; k < dataSet1.CLIENT3.Columns.Count; k++)

if (!dataSet1.CLIENT3.Rows[i].ItemArray[k].ToString().

Equals(dataSet2.CLIENT3.Rows[j].ItemArray[k].ToString()))

flag = false;

if (flag) break;

}

if (!flag) return false;

}

return true;

}

}

}

using System;

using System.Data;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using lab\_2.Accessors;

using lab\_2.Logging;

using lab\_2;

using lab\_2.BLogic;

using NUnit.Framework;

namespace tst

{

class MachineTester

{

[Test]

[Timeout(2000)]

public void Test\_BL01\_InsertUpdate()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

BLogic.FillMachineTable(dataSet1);

dataSet1.CITY4.AddCITY4Row("Барнаул");

dataSet1.CITY4.AddCITY4Row("Абакан");

dataSet1.CITY4.AddCITY4Row("Красноярск");

dataSet1.CITY4.AddCITY4Row("Улан-Удэ");

dataSet1.MACHINE2.AddMACHINE2Row(dataSet1.CITY4.ElementAt(1), 4, 1, 0, DateTime.Today);

dataSet1.MACHINE2.AddMACHINE2Row(dataSet1.CITY4.ElementAt(2), 41, 1, 0, DateTime.Today);

dataSet1.MACHINE2.AddMACHINE2Row(dataSet1.CITY4.ElementAt(3), 4, 1, 0, DateTime.Today);

dataSet1.MACHINE2.AddMACHINE2Row(dataSet1.CITY4.ElementAt(1), 4, 1, 0, DateTime.Today);

dataSet1.MACHINE2.AddMACHINE2Row(dataSet1.CITY4.ElementAt(2), 4, 1, 0, DateTime.Today);

dataSet1.MACHINE2.AddMACHINE2Row(dataSet1.CITY4.ElementAt(0), 4, 1, 0, DateTime.Today);

dataSet1.MACHINE2.AddMACHINE2Row(dataSet1.CITY4.ElementAt(1), 21, 1, 0, DateTime.Today);

BLogic.UpdateCityTable(dataSet1);

BLogic.UpdateMachineTable(dataSet1);

BLogic.FillCityTable(dataSet2);

BLogic.FillMachineTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_BL02\_DeleteUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

BLogic.FillCityTable(dataSet1);

BLogic.FillMachineTable(dataSet1);

for (int i = 0; i < dataSet1.MACHINE2.Rows.Count; i++)

dataSet1.MACHINE2.Rows[i].Delete();

for (int i = 0; i < dataSet1.CITY4.Rows.Count; i++)

dataSet1.CITY4.Rows[i].Delete();

BLogic.UpdateCityTable(dataSet1);

BLogic.UpdateMachineTable(dataSet1);

BLogic.FillCityTable(dataSet2);

BLogic.FillMachineTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_BL03\_UpdateUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

BLogic.FillCityTable(dataSet1);

BLogic.FillMachineTable(dataSet1);

for (int i = 0; i < dataSet1.CLIENT3.Rows.Count; i++)

{

if (dataSet1.CLIENT3.Rows[i].ItemArray[0].Equals((Object)1))

{

dataSet1.CLIENT3.Rows[i].ItemArray[2] = (decimal)4.2;

break;

}

}

BLogic.UpdateCityTable(dataSet1);

BLogic.UpdateMachineTable(dataSet1);

BLogic.FillCityTable(dataSet2);

BLogic.FillMachineTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_BL04\_WhenTryToDeleteParent()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

BLogic.FillCityTable(dataSet1);

BLogic.FillMachineTable(dataSet1);

decimal index = (decimal)dataSet1.MACHINE2.Rows[0].ItemArray[1];

dataSet1.CITY4.Rows[(int)index].Delete();

//if (BLogic.UpdateCityTable(dataSet1) == 0)

{

dataSet1.CITY4.Rows[(int)index].RejectChanges();

}

BLogic.UpdateMachineTable(dataSet1);

BLogic.FillCityTable(dataSet2);

BLogic.FillMachineTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

public bool CheckIdent(lab\_2.baykalDataSet dataSet1, lab\_2.baykalDataSet dataSet2)

{

if (dataSet1.MACHINE2.Count != dataSet2.MACHINE2.Count) return false;

bool flag = true;

for (int i = 0; i < dataSet1.MACHINE2.Count; i++)

{

flag = false;

for (int j = 0; j < dataSet1.MACHINE2.Count; j++)

{

if (dataSet1.MACHINE2.Rows[i].ItemArray[0].Equals(dataSet2.MACHINE2.Rows[j].ItemArray[0])

&& Int32.Equals( dataSet1.MACHINE2.Rows[i].ItemArray[1], dataSet2.MACHINE2.Rows[j].ItemArray[1])

&& Int32.Equals(dataSet1.MACHINE2.Rows[i].ItemArray[2], dataSet2.MACHINE2.Rows[j].ItemArray[2])

&& Int32.Equals(dataSet1.MACHINE2.Rows[i].ItemArray[3], dataSet2.MACHINE2.Rows[j].ItemArray[3])

&& Int32.Equals(dataSet1.MACHINE2.Rows[i].ItemArray[4], dataSet2.MACHINE2.Rows[j].ItemArray[4])

&& DateTime.Compare((DateTime)dataSet1.MACHINE2.Rows[i].ItemArray[5], (DateTime)dataSet2.MACHINE2.Rows[j].ItemArray[5])==0

)

flag = true;

if (flag) break;

}

if (!flag) return false;

}

return true;

}

//\* \*/

}

}

using System;

using System.Data;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using lab\_2.Accessors;

using lab\_2.Logging;

using lab\_2.BLogic;

using lab\_2;

using Oracle;

using OracleClient = Oracle.DataAccess.Client;

using NUnit.Framework;

namespace tst

{

class ZayavkaTester

{

[Test]

[Timeout(2000)]

public void Test\_BL01\_InsertUpdate()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

dataSet1.CITY4.AddCITY4Row("Барнаул");

dataSet1.CITY4.AddCITY4Row("Абакан");

dataSet1.CITY4.AddCITY4Row("Красноярск");

dataSet1.CITY4.AddCITY4Row("Улан-Удэ");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(0), "3333333", "fio1", "address1", "name1");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(1), "555-555-555", "fio2", "address2", "name2");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(1), "555-555-555", "fio3", "address3", "name3");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(0), "555-555-555", "fio4", "address4", "name4");

dataSet1.ZAYAVKA1.AddZAYAVKA1Row(

dataSet1.CLIENT3.ElementAt(0),

dataSet1.CLIENT3.ElementAt(1),

null,

DateTime.Today,

"cheeeez",

(decimal)10, (decimal)3, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0);

dataSet1.ZAYAVKA1.AddZAYAVKA1Row(

dataSet1.CLIENT3.ElementAt(2),

dataSet1.CLIENT3.ElementAt(3),

null,

DateTime.Today,

"cheeeez",

(decimal)10, (decimal)3, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0);

BLogic.UpdateCityTable(dataSet1);

BLogic.UpdateClientTable(dataSet1);

BLogic.UpdateZayavkaTable(dataSet1);

BLogic.FillCityTable(dataSet2);

BLogic.FillClientTable(dataSet2);

BLogic.FillZayavkaTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2) && (new CityTester().CheckIdent(dataSet1,dataSet2)));

}

[Test]

public void Test\_BL02\_DeleteUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

BLogic.FillCityTable(dataSet1);

BLogic.FillClientTable(dataSet1);

BLogic.FillZayavkaTable(dataSet1);

for (int i = 0; i < dataSet1.ZAYAVKA1.Rows.Count; i++)

dataSet1.ZAYAVKA1.Rows[i].Delete();

for (int i = 0; i < dataSet1.CLIENT3.Rows.Count; i++)

dataSet1.CLIENT3.Rows[i].Delete();

for (int i = 0; i < dataSet1.CITY4.Rows.Count; i++)

dataSet1.CITY4.Rows[i].Delete();

BLogic.UpdateCityTable(dataSet1);

BLogic.UpdateClientTable(dataSet1);

BLogic.UpdateZayavkaTable(dataSet1);

BLogic.FillCityTable(dataSet2);

BLogic.FillClientTable(dataSet2);

BLogic.FillZayavkaTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

public void Test\_BL03\_UpdateUpdate()

{

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

BLogic.FillCityTable(dataSet1);

BLogic.FillClientTable(dataSet1);

BLogic.FillZayavkaTable(dataSet1);

for (int i = 0; i < dataSet1.ZAYAVKA1.Rows.Count; i++)

{

if (dataSet1.ZAYAVKA1.Rows[i].ItemArray[0].Equals((Object)1))

{

dataSet1.ZAYAVKA1.Rows[i].ItemArray[6] = (decimal)20;

break;

}

}

BLogic.UpdateCityTable(dataSet1);

BLogic.UpdateClientTable(dataSet1);

BLogic.UpdateZayavkaTable(dataSet1);

BLogic.FillCityTable(dataSet2);

BLogic.FillClientTable(dataSet2);

BLogic.FillZayavkaTable(dataSet2);

Assert.IsTrue(CheckIdent(dataSet1, dataSet2));

}

[Test]

[ExpectedException(typeof(System.Data.NoNullAllowedException))]

public void Test\_BL05\_AddNotExistingParent()

{

//создадим датасет, заполним его значениями

lab\_2.baykalDataSet dataSet1 = new baykalDataSet();

lab\_2.baykalDataSet dataSet2 = new baykalDataSet();

lab\_2.BLogic.BusinessLogic BLogic = new BusinessLogic();

dataSet1.CITY4.AddCITY4Row("Барнаул");

dataSet1.CITY4.AddCITY4Row("Абакан");

dataSet1.CITY4.AddCITY4Row("Красноярск");

dataSet1.CITY4.AddCITY4Row("Улан-Удэ");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(0), "3333333", "fio1", "address1", "name1");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(1), "555-555-555", "fio2", "address2", "name2");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(1), "555-555-555", "fio3", "address3", "name3");

dataSet1.CLIENT3.AddCLIENT3Row(dataSet1.CITY4.ElementAt(0), "555-555-555", "fio4", "address4", "name4");

dataSet1.ZAYAVKA1.AddZAYAVKA1Row(

dataSet1.CLIENT3.ElementAt(0),

dataSet1.CLIENT3.ElementAt(1),

null,

DateTime.Today,

"cheeeez",

(decimal)10, (decimal)3, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0);

dataSet1.ZAYAVKA1.AddZAYAVKA1Row(

dataSet1.CLIENT3.ElementAt(2),

dataSet1.CLIENT3.ElementAt(3),

null,

DateTime.Today,

"cheeeez",

(decimal)10, (decimal)3, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0, (decimal)0);

BLogic.UpdateCityTable(dataSet1);

BLogic.UpdateClientTable(dataSet1);

Assert.AreEqual(0, BLogic.UpdateZayavkaTable(dataSet1));

}

public bool CheckIdent(lab\_2.baykalDataSet dataSet1, lab\_2.baykalDataSet dataSet2)

{

if (dataSet1.ZAYAVKA1.Count != dataSet2.ZAYAVKA1.Count) return false;

bool flag = true;

for (int i = 0; i < dataSet1.ZAYAVKA1.Count; i++)

{

flag = false;

for (int j = 0; j < dataSet1.ZAYAVKA1.Count; j++)

{

flag = true;

for (int k = 0; k < dataSet1.ZAYAVKA1.Columns.Count; k++)

if (!dataSet1.ZAYAVKA1.Rows[i].ItemArray[k].ToString().

Equals(dataSet2.ZAYAVKA1.Rows[j].ItemArray[k].ToString()))

flag = false;

if (flag) break;

}

if (!flag) return false;

}

return true;

}

\* \*/

}

}

# Лабораторная работа 6

## 1 Задание

**«Презентационный слой на стороне клиента»**

**Цель работы:**

получить практические навыки в использовании spring framework. Научится создавать web-service на платформе .net.

**Задачи:**

создать web-service для удаленного доступа к бизнес-логике с использованием spring framework.

## 2 Реализация

Конфигурация

<?xml version="1.0"?>

<configuration>

<configSections>

<!-- Spring -->

<sectionGroup name="spring">

<section name="objects" type="Spring.Context.Support.DefaultSectionHandler, Spring.Core"/>

<section name="context" type="Spring.Context.Support.ContextHandler, Spring.Core"/>

</sectionGroup>

<!-- Spring -->

</configSections>

<spring>

<context type="Spring.Context.Support.XmlApplicationContext, Spring.Core">

<resource uri="config://spring/objects"/>

</context>

<objects xmlns="http://www.springframework.net">

<object id="City" type="lab\_2.Accessors.CityDataAccessor, lab\_2" />

<object id="Client" type="lab\_2.Accessors.ClientDataAccessor, lab\_2" />

<object id="Machine" type="lab\_2.Accessors.MachineDataAccessor, lab\_2" />

<object id="Zayavka" type="lab\_2.Accessors.ZayavkaDataAccessor, lab\_2" />

<object id="ClientView" type="lab\_2.Accessors.ClientViewDataAccessor, lab\_2" />

<object id="ZayavkaView" type="lab\_2.Accessors.ZayavkaViewDataAccessor, lab\_2" />

<object id="Connection" type="System.Data.SqlClient.SqlConnection, System.Data" >

<property name="ConnectionString" value='Data Source=book\sqlexpress8;Initial Catalog=dbb;Integrated Security=True;Pooling=False' />

</object>

<object id="BuisnessLogicObject" type="lab\_2.BLogic.BusinessLogic, lab\_2" autowire="byName"/>

<object id="BLExporter" type="Spring.Web.Services.WebServiceExporter, Spring.Web">

<property name="TargetName" value="BuisnessLogicObject"/>

<property name="Namespace" value="http://Special/services"/>

<property name="Description" value="My exported function"/>

<property name="MemberAttributes">

<dictionary>

<entry key="UpdateAll">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Функция выгзагрузки данных на сервер."/>

</object>

</entry>

<entry key="UpdateCityTable">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Заполнение DataSet."/>

</object>

</entry>

<entry key="UpdateClientTable">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Заполнение DataSet."/>

</object>

</entry>

<entry key="UpdateMachineTable">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Заполнение DataSet."/>

</object>

</entry>

<entry key="UpdateZayavkaTable">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Заполнение DataSet."/>

</object>

</entry>

<entry key="ReadAll">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Функция загрузки данных с сервера."/>

</object>

</entry>

<entry key="ReadCityTable">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Заполнение DataSet."/>

</object>

</entry>

<entry key="ReadClientTable">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Заполнение DataSet."/>

</object>

</entry>

<entry key="ReadClientView">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Заполнение DataSet."/>

</object>

</entry>

<entry key="ReadMachineTable">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Заполнение DataSet."/>

</object>

</entry>

<entry key="ReadZayavkaTable">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Заполнение DataSet."/>

</object>

</entry>

<entry key="ReadZayavkaView">

<object type="System.Web.Services.WebMethodAttribute, System.Web.Services">

<property name="Description" value="Заполнение DataSet."/>

</object>

</entry>

</dictionary>

</property>

</object>

</objects>

</spring>

<system.web>

<authentication mode="Windows"/>

<pages>

<controls>

<add tagPrefix="asp" namespace="System.Web.UI" assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add tagPrefix="asp" namespace="System.Web.UI.WebControls" assembly="System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

</controls>

</pages>

<httpHandlers>

<remove verb="\*" path="\*.asmx"/>

<!-- This is my line-->

<add verb="\*" path="\*.asmx" type="Spring.Web.Services.WebServiceHandlerFactory, Spring.Web"/>

<!--<add verb="\*" path="\*.asmx" validate="false" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>-->

<add verb="\*" path="\*\_AppService.axd" validate="false" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add verb="GET,HEAD" path="ScriptResource.axd" type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" validate="false"/>

</httpHandlers>

<httpModules>

<add name="ScriptModule" type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

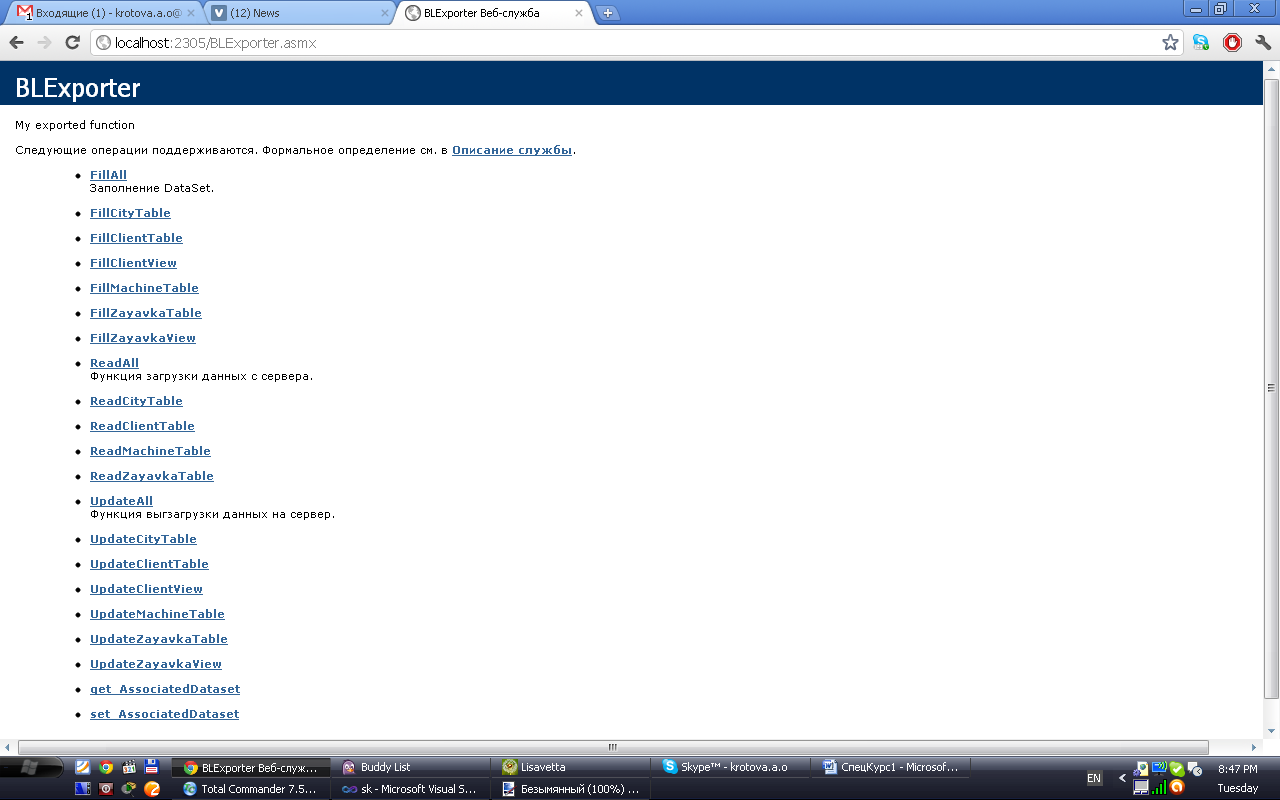
</httpModules>

</system.web>

</configuration>

## 3 Результаты работы

[http://localhost:2305/BLExporter.asmx](http://localhost:49990/BLExporter.asmx)

****

# Лабораторная работа 7

## 1 Задание

**«Клиентская часть»**

**Цель работы:**

освоить технологию DataBinding. Получить навыки в разработке клиентской части, которая использует в качестве бизнес-логики Web-service.

**Задачи:**

реализовать клиентскую часть с использованием web-service, реализованного в предыдущей лабораторной работе. Использовать технологию DataBinding для привязки пользовательского интерфейса к модели данных. Пользовательский интерфейс должен иметь списки данных и индивидуальные окна для редактирования отдельных записей. Также должна быть возможность редактировать справочники из любого окна.

## 2 Реализация

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using ClientApplication.localhost;

namespace ClientApplication

{

public partial class MainForm : Form

{

bool loaded = false;

public MainForm()

{

InitializeComponent();

assist.exporter = new BLExporter();

assist.dataSet = new baykalDataSet();

}

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

{

assist.dataSet = assist.exporter.ReadAll();

zViewBindingSource.DataSource = assist.dataSet.ZView;

cITY4BindingSource.DataSource = assist.dataSet.CITY4;

mACHINE2BindingSource.DataSource = assist.dataSet.MACHINE2;

newOrderButton.Enabled = removeOrderButton.Enabled = false;

loaded = true;

}

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

{

}

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

{

ClientView clientView = new ClientView();

clientView.ShowDialog();

loaded = false;

zViewBindingSource.DataSource = assist.dataSet.ZView;

cITY4BindingSource.DataSource = assist.dataSet.CITY4;

mACHINE2BindingSource.DataSource = assist.dataSet.MACHINE2;

newOrderButton.Enabled = removeOrderButton.Enabled = false;

loaded = true;

}

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

{

AddOrder addOrder = new AddOrder();

addOrder.Text = "Новая заявка";

int pkMachine =

((int)dataGridView2.SelectedRows[0].Cells[0].Value);

bool good = true;

bool willAdd = false;

int avRec = (assist.dataSet.CITY4.FindBypk\_city4

(assist.dataSet.MACHINE2.FindBypk\_machine2(pkMachine).city2)).pk\_city4;

var clc =

(from baykalDataSet.CLIENT3Row cr in assist.dataSet.CLIENT3

where cr.city3 == avRec

select cr).Distinct().Count();

if ((int)clc == 0)

{

good = false;

if (MessageBox.Show(this, "По данному направлению нет клиентов, добавить?", "Предупреждение",

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

{

willAdd = true;

}

}

if (willAdd)

{

AddClient addClient = new AddClient();

addClient.Filter(avRec.ToString());

addClient.Tag = null;

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

{

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

good = true;

}

else

{

assist.dataSet.CLIENT3.RejectChanges();

}

}

if (good)

{

addOrder.Tag = new zExchange() { row = null, machine1 = pkMachine };

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

{

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

zViewBindingSource.DataSource = assist.dataSet.ZView;

//zViewBindingSource.Filter = "";

dataGridView2.Update();

dataGridView1.Update();

assist.dataSet.AcceptChanges();

}

else

{

assist.dataSet.CLIENT3.RejectChanges();

}

}

}

private void dataGridView3\_SelectionChanged(object sender, EventArgs e)

{

//if (loaded)

{

String pkCity = "0";

if (dataGridView3.SelectedRows.Count > 0)

{

pkCity =

((int)dataGridView3.SelectedRows[0].Cells[0].Value).ToString();

dataGridView2.ClearSelection();

}

mACHINE2BindingSource.Filter = "city2 = " + pkCity;

}

}

private void dataGridView2\_SelectionChanged(object sender, EventArgs e)

{

//if (loaded)

{

String filterForZ = @"0";

newOrderButton.Enabled = removeOrderButton.Enabled = false;

if (dataGridView2.SelectedRows.Count > 0)

{

String pkMachine =

((int)dataGridView2.SelectedRows[0].Cells[0].Value).ToString();

filterForZ = pkMachine;

newOrderButton.Enabled = removeOrderButton.Enabled = true;

}

zViewBindingSource.Filter = "machine1 = " + filterForZ;

}

}

private void button5\_Click(object sender, EventArgs e)

{

AddCity addCity = new AddCity();

addCity.Text = "Новое направление";

addCity.Tag = null;

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

{

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

loaded = false;

cITY4BindingSource.DataSource = assist.dataSet.CITY4;

dataGridView3.Update();

loaded = true;

}

else

{

assist.dataSet.CITY4.RejectChanges();

}

}

private void dataGridView3\_CellContentDoubleClick(object sender, DataGridViewCellEventArgs e)

{

AddCity addCity = new AddCity();

addCity.Text = "Редактировать направление";

int i;

i = (int)dataGridView3.CurrentRow.Cells[0].Value;

addCity.Tag = assist.dataSet.CITY4.FindBypk\_city4(i);

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

{

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

loaded = false;

cITY4BindingSource.DataSource = assist.dataSet.CITY4;

dataGridView3.Refresh();

loaded = true;

}

else

{

assist.dataSet.CITY4.RejectChanges();

}

}

private void button6\_Click(object sender, EventArgs e)

{

if (MessageBox.Show(this, "Вы уверены?", "Подтвердить удаление",

MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)

{

int index = (int)dataGridView3.SelectedRows[0].Cells[0].Value;

baykalDataSet.CITY4Row rowDelete

= assist.dataSet.CITY4.FindBypk\_city4(index);

var clients =

(from baykalDataSet.CLIENT3Row rowClient in assist.dataSet.CLIENT3

where rowClient.city3 == index

select rowClient).Distinct().Count();

var machines =

(from baykalDataSet.MACHINE2Row rowMachine in assist.dataSet.MACHINE2

where rowMachine.city2 == index

select rowMachine).Distinct().Count();

if ((int)clients > 0 && (int)machines > 0)

{

MessageBox.Show(this, "На запись есть ссылки, удаление запрещено", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Stop);

return;

}

else

{

//rowDelete.Delete();

cITY4BindingSource.RemoveCurrent();

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

loaded = false;

cITY4BindingSource.DataSource = assist.dataSet.CITY4;

dataGridView3.Refresh();

loaded = true;

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

AddMachine addMachine = new AddMachine();

addMachine.Text = "Новая машина";

int pkCity =

((int)dataGridView3.SelectedRows[0].Cells[0].Value);

addMachine.Tag = new mExchange() { row = null, city2 = pkCity };

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

{

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

loaded = false;

mACHINE2BindingSource.DataSource = assist.dataSet.MACHINE2;

dataGridView2.Refresh();

loaded = true;

}

else

{

assist.dataSet.MACHINE2.RejectChanges();

}

}

private void dataGridView2\_CellContentDoubleClick(object sender, DataGridViewCellEventArgs e)

{

AddMachine addMachine = new AddMachine();

addMachine.Text = "Редактировать машину";

baykalDataSet.MACHINE2Row rowEdit;

int pkCity =

((int)dataGridView3.SelectedRows[0].Cells[0].Value);

int i;

i = (int)dataGridView2.CurrentRow.Cells[0].Value;

rowEdit = assist.dataSet.MACHINE2.FindBypk\_machine2(i);

addMachine.Tag = new mExchange() { row = rowEdit, city2 = pkCity };

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

{

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

loaded = false;

mACHINE2BindingSource.DataSource = assist.dataSet.MACHINE2;

dataGridView2.Refresh();

loaded = true;

}

else

{

assist.dataSet.MACHINE2.RejectChanges();

}

}

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

{

if (MessageBox.Show(this, "Вы уверены?", "Подтвердить удаление",

MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)

{

int index = (int)dataGridView2.SelectedRows[0].Cells[0].Value;

baykalDataSet.MACHINE2Row rowDelete

= assist.dataSet.MACHINE2.FindBypk\_machine2(index);

var zay =

(from baykalDataSet.ZAYAVKA1Row rowZ in assist.dataSet.ZAYAVKA1

where rowZ.machine1 == index

select rowZ).Distinct().Count();

if ((int)zay > 0)

{

MessageBox.Show(this, "На запись есть ссылки, удаление запрещено", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Stop);

return;

}

else

{

//rowDelete.Delete();

mACHINE2BindingSource.RemoveCurrent();

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

loaded = false;

mACHINE2BindingSource.DataSource = assist.dataSet.MACHINE2;

dataGridView2.Refresh();

loaded = true;

}

}

}

private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

AddOrder addOrder = new AddOrder();

addOrder.Text = "Редактировать заявку";

baykalDataSet.ZAYAVKA1Row rowEdit = null;

int pkMachine =

((int)dataGridView2.SelectedRows[0].Cells[0].Value);

int i;

i = (int)dataGridView2.CurrentRow.Cells[0].Value;

rowEdit = assist.dataSet.ZAYAVKA1.FindBypk\_zayavka1(i);

addOrder.Tag = new zExchange() { row = rowEdit, machine1 = pkMachine };

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

{

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

zViewBindingSource.DataSource = assist.dataSet.ZView;

dataGridView2.Refresh();

dataGridView1.Refresh();

assist.dataSet.AcceptChanges();

}

else

{

assist.dataSet.CLIENT3.RejectChanges();

}

}

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

{

if (MessageBox.Show(this, "Вы уверены?", "Подтвердить удаление",

MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)

{

int index = (int)dataGridView1.SelectedRows[0].Cells[0].Value;

baykalDataSet.ZAYAVKA1Row rowDelete

= assist.dataSet.ZAYAVKA1.FindBypk\_zayavka1(index);

var zv = from baykalDataSet.ZViewRow z in assist.dataSet.ZView

where z.pk\_zayavka1 == index

select z;

{

rowDelete.Delete();

baykalDataSet.ZViewRow row = (zv.ElementAt(0));

row.Delete();

assist.dataSet.ZView.AcceptChanges();

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

loaded = false;

zViewBindingSource.DataSource = assist.dataSet.ZView;

dataGridView1.Refresh();

loaded = true;

}

}

}

}

class mExchange

{

public baykalDataSet.MACHINE2Row row { get; set; }

public int city2 { get; set; }

}

class zExchange

{

public baykalDataSet.ZAYAVKA1Row row { get; set; }

public int machine1 { get; set; }

}

class cbItem

{

public int value { get; set; }

public String str { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using ClientApplication.localhost;

namespace ClientApplication

{

public partial class ClientView : Form

{

bool loaded =false;

bool cityf = false;

private void ApplyFilter(bool filterCity)

{

String Filter = @"";

clientViewBindingSource.Filter = null;

bool flag = false;

if (textBox1.Text.Trim() != "")

{

Filter += "name3 = '" + textBox1.Text + "'";

flag = true;

}

if (comboBox1.SelectedValue != null && filterCity)

{

Filter += flag ? " and " : " ";

Filter += " city3 = " + ((int)comboBox1.SelectedValue).ToString();

}

if (Filter != "")

clientViewBindingSource.Filter = Filter;

dataGridView1.ClearSelection();

}

public ClientView()

{

InitializeComponent();

}

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

{

clientViewBindingSource.DataSource = assist.dataSet.ClientView;

cITY4BindingSource.DataSource = assist.dataSet.CITY4;

loaded = true;

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if(loaded)

ApplyFilter(cityf);

}

private void comboBox1\_SelectedValueChanged(object sender, EventArgs e)

{

if (loaded)

{

cityf = true;

ApplyFilter(cityf);

}

}

private void button3\_Click(object sender, EventArgs e)

{

if (loaded)

{

if (comboBox1.Items.Count > 0)

comboBox1.SelectedIndex = 0;

cityf = false;

ApplyFilter(cityf);

}

}

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

{

AddClient addClient = new AddClient();

addClient.Tag = null;

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

{

//assist.exporter.UpdateAll(assist.dataSet);

// assist.dataSet = assist.exporter.ReadAll();

assist.exporter.UpdateClientTable(assist.dataSet);

assist.dataSet = assist.exporter.ReadClientView();

clientViewBindingSource.DataSource = assist.dataSet.ClientView;

dataGridView1.Update();

}

else

{

assist.dataSet.CLIENT3.RejectChanges();

}

}

private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

AddClient addClient = new AddClient();

int i;

i = (int)dataGridView1.CurrentRow.Cells[0].Value;

addClient.Tag = assist.dataSet.CLIENT3.FindBypk\_client3(i);

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

{

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

clientViewBindingSource.DataSource = assist.dataSet.ClientView;

dataGridView1.Update();

assist.dataSet.CLIENT3.AcceptChanges();

}

else

{

assist.dataSet.CLIENT3.RejectChanges();

}

}

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

{

if (MessageBox.Show(this, "Вы уверены?", "Подтвердить удаление",

MessageBoxButtons.OKCancel,MessageBoxIcon.Question) == DialogResult.OK)

{

int index = (int)dataGridView1.SelectedRows[0].Cells[0].Value;

baykalDataSet.CLIENT3Row rowDelete =

assist.dataSet.CLIENT3.FindBypk\_client3(index);

var rd = from baykalDataSet.ClientViewRow r in assist.dataSet.ClientView

where r.pk\_client3 == index

select r;

var con = (from baykalDataSet.ZAYAVKA1Row row in assist.dataSet.ZAYAVKA1

where row.otprav1 == index || row.poluch1 == index

select row).Distinct().Count();

if (con > 0)

{

MessageBox.Show(this, "На запись есть ссылки, удаление запрещено", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Stop);

return;

}

else

{

rowDelete.Delete();

((baykalDataSet.ClientViewRow)rd.ElementAt(0)).Delete();

baykalDataSet1.ClientView.AcceptChanges();

assist.exporter.UpdateAll(assist.dataSet);

assist.dataSet = assist.exporter.ReadAll();

//assist.dataSet = baykalDataSet1;

clientViewBindingSource.DataSource = assist.dataSet.ClientView;

dataGridView1.Refresh();

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using ClientApplication.localhost;

namespace ClientApplication

{

public partial class AddClient : Form

{

baykalDataSet ds;

baykalDataSet.CLIENT3Row row;

bool toClose = true;

string flt = "";

public AddClient()

{

InitializeComponent();

}

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

{

try

{

if (textBox1.Text.Trim() == "")

throw new Exception("Введите название");

if (textBox1.Text.Trim() == "")

throw new Exception("Введите адрес");

if (textBox1.Text.Trim() == "")

throw new Exception("Введите ФИО");

if (textBox1.Text.Trim() == "")

throw new Exception("Введите телефон");

if (Tag == null)

this.Tag = assist.dataSet.CLIENT3.AddCLIENT3Row(

assist.dataSet.CITY4.FindBypk\_city4((int)comboBox1.SelectedValue),

textBox1.Text,

textBox2.Text,

textBox3.Text,

textBox4.Text);

toClose = true;

}

catch (Exception ex)

{

MessageBox.Show(this, ex.Message, "Ошибка", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

toClose = false;

}

}

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

{

//ds = assist.dataSet;

cITY4BindingSource.DataSource = assist.dataSet.CITY4;

if (flt != "")

{

cITY4BindingSource.DataSource = assist.exporter.ReadAll().CITY4;

cITY4BindingSource.Filter = "pk\_city4 = " + "'" + flt + "'";

}

row = (baykalDataSet.CLIENT3Row)this.Tag;

if (row != null)

{

comboBox1.DataBindings.Add("SelectedValue", row, "city3");

textBox1.DataBindings.Add("Text", row, "name3");

textBox2.DataBindings.Add("Text", row, "address3");

textBox3.DataBindings.Add("Text", row, "fio3");

textBox4.DataBindings.Add("Text", row, "phone3");

}

}

public void Filter(string s)

{

flt = s;

}

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

{

}

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

{

if (!toClose && DialogResult == DialogResult.OK)

e.Cancel = true;

toClose = true;

}

}

}

using System;

using System.Collections.Generic;

using System.Collections;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace ClientApplication

{

public partial class AddMachine : Form

{

mExchange tg;

bool toClose = true;

public AddMachine()

{

InitializeComponent();

}

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

{

tg = (mExchange)this.Tag;

cITY4BindingSource.DataSource = assist.dataSet.CITY4;

comboBox3.DataBindings.Add("SelectedValue", tg, "city2");

comboBox3.Enabled = false;

ArrayList arrayList1 = new ArrayList();

arrayList1.Add(new cbItem { value = 1, str = "Да" });

arrayList1.Add(new cbItem { value = 0, str = "Нет" });

comboBox1.DataSource = arrayList1;

comboBox1.ValueMember = "value";

comboBox1.DisplayMember = "str";

comboBox1.DropDownStyle = ComboBoxStyle.DropDownList;

ArrayList arrayList2 = new ArrayList();

arrayList2.Add(new cbItem { value = 1, str = "Да" });

arrayList2.Add(new cbItem { value = 0, str = "Нет" });

comboBox2.DataSource = arrayList2;

comboBox2.ValueMember = "value";

comboBox2.DisplayMember = "str";

comboBox2.DropDownStyle = ComboBoxStyle.DropDownList;

if (tg.row != null)

{

comboBox3.DataBindings.RemoveAt(0);

comboBox3.DataBindings.Add("SelectedValue", tg.row, "city2");

dateTimePicker1.DataBindings.Add("Value", tg.row, "date2");

numericUpDown1.DataBindings.Add("Value", tg.row, "max\_ves2");

comboBox1.DataBindings.Add("SelectedValue", tg.row, "rezhim2");

comboBox2.DataBindings.Add("SelectedValue", tg.row, "done2");

}

}

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

{

try

{

localhost.baykalDataSet.CITY4Row r = assist.dataSet.CITY4.FindBypk\_city4((int)comboBox3.SelectedValue);

if (tg.row == null)

tg.row = assist.dataSet.MACHINE2.AddMACHINE2Row(

assist.dataSet.CITY4.FindBypk\_city4((int)comboBox3.SelectedValue),

(int)numericUpDown1.Value,

(int)comboBox1.SelectedValue,

(int)comboBox2.SelectedValue,

dateTimePicker1.Value);

toClose = true;

}

catch(Exception ex)

{

MessageBox.Show(this, ex.Message, "Ошибка", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

toClose = false;

}

}

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

{

if (!toClose && DialogResult == DialogResult.OK)

e.Cancel = true;

toClose = true;

}

}

}

using System;

using System.Collections.Generic;

using System.Collections;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using ClientApplication.localhost;

namespace ClientApplication

{

public partial class AddOrder : Form

{

zExchange tg;

localhost.baykalDataSet.ZAYAVKA1Row row;

bool toClose = true;

public AddOrder()

{

InitializeComponent();

}

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

{

tg = (zExchange)this.Tag;

row = tg.row;

if (row != null)

{

comboBox1.DataBindings.Add("SelectedValue", row, "otprav1");

comboBox2.DataBindings.Add("SelectedValue", row, "poluch1");

dateTimePicker1.DataBindings.Add("Value", row, "dsost1");

textBox1.DataBindings.Add("Text", row, "gruz\_descr1");

numericUpDown1.DataBindings.Add("Value", row, "ves1");

numericUpDown2.DataBindings.Add("Value", row, "places1");

comboBox3.DataBindings.Add("SelectedValue", row, "exped1");

comboBox4.DataBindings.Add("SelectedValue", row, "srocn1");

comboBox5.DataBindings.Add("SelectedValue", row, "done1");

comboBox6.DataBindings.Add("SelectedValue", row, "rezhim1");

}

senderBS.DataSource = assist.dataSet.CLIENT3;

recvBS.DataSource = assist.exporter.ReadAll().CLIENT3;

int avRec = (assist.dataSet.CITY4.FindBypk\_city4

(assist.dataSet.MACHINE2.FindBypk\_machine2(tg.machine1).city2)).pk\_city4;

recvBS.Filter = "city3 = " + avRec.ToString();

ArrayList arrayList3 = new ArrayList();

arrayList3.Add(new cbItem { value = 1, str = "Да" });

arrayList3.Add(new cbItem { value = 0, str = "Нет" });

comboBox3.DataSource = arrayList3;

comboBox3.ValueMember = "value";

comboBox3.DisplayMember = "str";

comboBox3.DropDownStyle = ComboBoxStyle.DropDownList;

ArrayList arrayList4 = new ArrayList();

arrayList4.Add(new cbItem { value = 1, str = "Да" });

arrayList4.Add(new cbItem { value = 0, str = "Нет" });

comboBox4.DataSource = arrayList4;

comboBox4.ValueMember = "value";

comboBox4.DisplayMember = "str";

comboBox4.DropDownStyle = ComboBoxStyle.DropDownList;

ArrayList arrayList5 = new ArrayList();

arrayList5.Add(new cbItem { value = 1, str = "Да" });

arrayList5.Add(new cbItem { value = 0, str = "Нет" });

comboBox5.DataSource = arrayList5;

comboBox5.ValueMember = "value";

comboBox5.DisplayMember = "str";

comboBox5.DropDownStyle = ComboBoxStyle.DropDownList;

ArrayList arrayList6 = new ArrayList();

arrayList6.Add(new cbItem { value = 1, str = "Да" });

arrayList6.Add(new cbItem { value = 0, str = "Нет" });

comboBox6.DataSource = arrayList6;

comboBox6.ValueMember = "value";

comboBox6.DisplayMember = "str";

comboBox6.DropDownStyle = ComboBoxStyle.DropDownList;

}

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

{

try

{

//int buf;

if (textBox1.Text.Trim() == "")

throw new Exception("Введите описание");

if (row == null)

{

assist.dataSet.ZAYAVKA1.AddZAYAVKA1Row

(assist.dataSet.CLIENT3.FindBypk\_client3((int)comboBox1.SelectedValue),

assist.dataSet.CLIENT3.FindBypk\_client3((int)comboBox2.SelectedValue),

assist.dataSet.MACHINE2.FindBypk\_machine2(tg.machine1),

dateTimePicker1.Value,

textBox1.Text.Trim(),

(int)numericUpDown1.Value,

(int)numericUpDown2.Value,

(int)comboBox3.SelectedValue,

(int)comboBox3.SelectedValue,

(int)comboBox3.SelectedValue,

(int)comboBox3.SelectedValue);

}

toClose = true;

}

catch (Exception ex)

{

MessageBox.Show(this, ex.Message, "Ошибка", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);

toClose = false;

}

}

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

{

if (!toClose && DialogResult == DialogResult.OK)

e.Cancel = true;

toClose = true;

}

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

{

}

}

}

