

УО «Полоцкий государственный университет»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ №9

**к выполнению лабораторной работы
по курсу «Базы данных» для специальности
Программное обеспечение информационных технологий 1-40 01 01**

ТЕМА: Создание клиентской части приложения для просмотра, редактирования данных БД. Вызов хранимых процедур из клиентской части на языке программирования C# в среде MS Visual Studio 2010.

ЦЕЛЬ: Научиться создавать клиентское приложение для работы с базой данных с применением встроенных инструментов на Visual C# 2010.

Новополоцк 2011

Методические указания разработала:

Старший преподаватель кафедры технологий программирования Бураченко Ирина
Брониславовна

ТЕМА: Создание клиентской части приложения для просмотра, редактирования данных БД. Вызов хранимых процедур из клиентской части на языке программирования C# в среде MS Visual Studio 2010.

ЦЕЛЬ: научиться создавать клиентское приложение для работы с базой данных с применением встроенных инструментов на Visual C# 2010.

Результат обучения:

После успешного завершения занятия пользователь должен:

- Изучить технологии доступа к данным на языке программирования C# в среде MS Visual Studio 2010.
- Уметь создать клиентскую часть приложения для просмотра, редактирования данных БД и осуществлять вызов хранимых процедур из клиентской части на языке программирования C#.

Используемая программа: Microsoft SQL Server 2008 и Microsoft Visual Studio 2010.

План занятия:

1. Изучить технологии доступа к данным на языке программирования C# в среде MS Visual Studio 2010.
2. Выполнить все задания по ходу лабораторной работы.
3. Выполнение индивидуального задания.

ВВЕДЕНИЕ

Для создания клиентского приложения на Visual C# 2010 используем пример базы данных с названием DB_Books. При выполнении примеров и заданий обращайтесь на соответствие названий БД, таблиц и других объектов проекта.

1.1. Работа с утилитой SQL Server Management Studio

Для написания программного кода в SQL Server Management Studio нужно нажать кнопку «Создать запрос» («New query») на панели инструментов «Стандартная» («Standart»).

1. Создать новую базу данных с названием DB_Books с помощью команды:

```
CREATE DATABASE DB_BOOKS
```

Для выполнения команды нажать F5.

2. Открыть утилиту SQL Server Management Studio. Проверить наличие БД DB_Books, если ее не видите в разделе DataBases, то нажмите F5 для обновления.

3. Создать в ней перечисленные таблицы с помощью следующих команд (для создания новой страницы для кода в SQL Server Management Studio нажать кнопку «Создать запрос»):

```
USE DB_BOOKS
CREATE TABLE Authors(Code_author INT PRIMARY KEY,
name_author CHAR(30), Birthday DATETIME)
```

```
CREATE TABLE Publishing_house(Code_publish INT PRIMARY KEY,
Publish CHAR(30), City CHAR(20))
```

```
CREATE TABLE Books(Code_book INT PRIMARY KEY, Title_book
CHAR(40), Code_author INT FOREIGN KEY REFERENCES Authors(
Code_author), Pages INT, Code_publish INT FOREIGN KEY
REFERENCES Publishing_house(Code_publish))
```

```
CREATE TABLE Deliveries(Code_delivery INT PRIMARY KEY,
Name_delivery CHAR(30), Name_company CHAR(20), Address
VARCHAR(100), Phone BIGINT, INN CHAR(13))
```

```
CREATE TABLE Purchases(Code_purchase INT PRIMARY KEY,
Code_book INT FOREIGN KEY REFERENCES Books(Code_book),
Date_order SMALLDATETIME, Code_delivery INT FOREIGN KEY
REFERENCES Deliveries(Code_delivery), Type_purchase BIT, Cost FLOAT,
Amount INT)
```

4. Запустите команду клавишей F5. В утилите SQL Server Management Studio проверить наличие БД DB_Books и таблиц в ней.

5. В разделе диаграмм создать новую диаграмму см. рисунок 1, в которую добавить из списка пять созданных таблиц, проверить связи между таблицами.

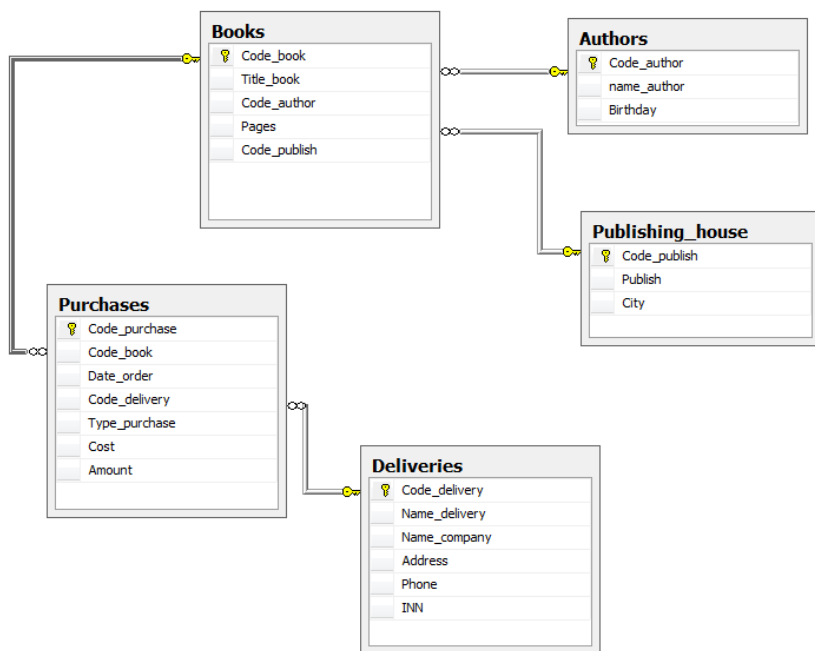


Рисунок 1 – Результат создание диаграммы

1.2. Работа в Visual C# 2010

1. В проекте выбираем меню Tools → Connect to DataBase.
2. В открывшемся окне в поле Data Source ставим Microsoft SQL Server, в поле Server Name – SQLEXPRESS, далее в поле Select or enter DB name выберите имя БД, к которой будем подключаться, and и нажмите OK см. рисунок 2.

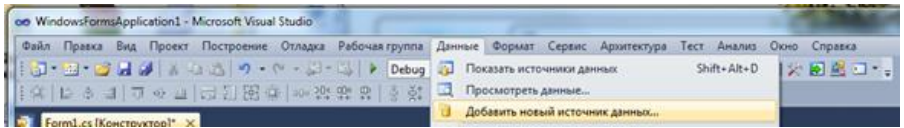


Рисунок 2 – Подключение источника данных

3. Теперь открыв окно Server Explorer можно увидеть подключенную БД.
4. На форму добавить 5 компонентов типа DataGridView (переименовать компоненты на Purchases, Books, Authors, Deliveries, Publish).
5. Во вкладке Data выберем Add New Data Source. В появившемся окне выберем DataBase и нажмем Next. Выбираем нашу БД, жмем Next. В появившемся окне поставим галочку на пункте Table (выбираем все созданные таблицы). Жмем Finish.
6. У каждой таблицы DataGridView изменим свойство DataSource на соответствующие названию этой таблицы см. рисунок 3.

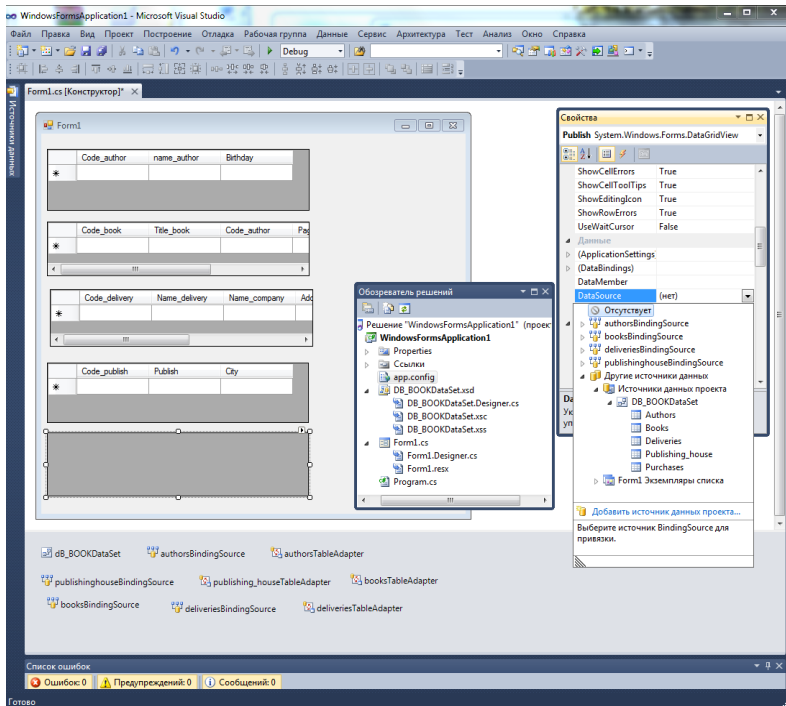


Рисунок 3 – Основная форма

7. На основной форме (Form1) добавить компонент меню. В редакторе меню сделать первый пункт «Работа с таблицами» и в подменю пункты: «Авторы», «Книги», «Издательства», «Поставщики», «Поставки».

8. Создать пять форм, каждую из которых назвать: FormAuthors, FormPurchases, FormBooks, FormDeliveries, FormPublish.

9. На основной форме в подпунктах меню в соответствующих методах Click вызвать соответствующие формы с помощью кода:

для FormAuthors:

```
FormAuthors myForm2 = new FormAuthors();
myForm2.Show();
```

для FormPurchases:

```
FormPurchases myForm3 = new FormPurchases();
myForm3.Show();
```

для FormBooks:

```
FormBooks myForm4 = new FormBooks();
myForm4.Show();
```

для FormDeliveries:

```
FormDeliveries myForm5 = new FormDeliveries();
myForm5.Show();
```

для FormPublish:

```
FormPublish myForm6 = new FormPublish();
myForm6.Show();
```

10. На формы FormAuthors, FormPurchases, FormBooks, FormDeliveries, FormPublish добавить по паре компонент типа DataGridView и BindingNavigator. Настроить у DataGridView свойство DataSource для связи с соответствующим источником данных. Затем необходимо настроить у BindingNavigator свойство BindingSource для связи с созданной таблицей (значение должно совпадать со значением свойства элемента DataGridView) см. рисунок 4.

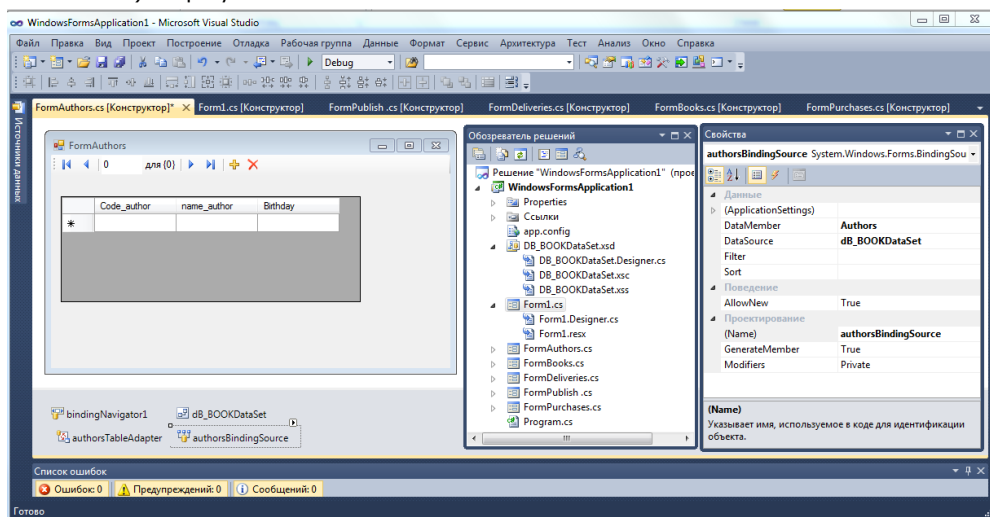


Рисунок 4 – Форма Авторы

11. Проверить работу приложения.

12. На форму FormBooks добавить 3 компонента типа TextBox и 2 компонента ComboBox см рисунок 5.

У 1-го компонента TextBox изменить свойства: (DataBinding)

Text booksBindingSource - Code book

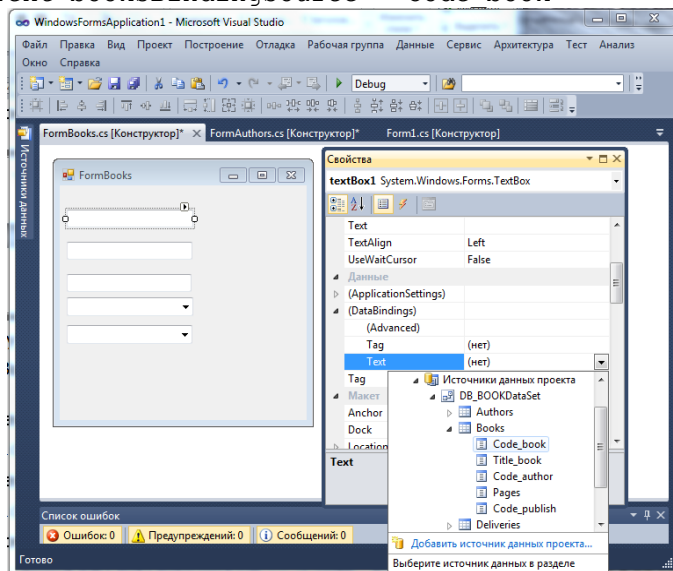


Рисунок 5 – Изменение свойства DataBinding на объекте TextBox

У 2-го компонента TextBox изменить свойства: (DataBinding)

Text booksBindingSource - Title_book

У 1-го компонента ComboBox изменить свойства: (DataBinding)

SelectedValue booksBindingSource - Code_author

DataSource authorsBindingSource

DisplayMember name_author

ValueMember Code_author

У 3-го компонента TextBox изменить свойства: (DataBinding)

Text booksBindingSource - Pages

У 2-го компонента ComboBox изменить свойства: (DataBinding)

SelectedValue booksBindingSource - Code_publish

DataSource publishinghouseBindingSource

DisplayMember Publish

ValueMember Code_publish

13. У компонента DataGridView убрать все галочки со свойств редактирования и добавления.

14. На форму FormBooks добавить компонент типа Button (кнопка обновления данных), свойство Text изменить на «Обновить» и прописать событие Click:

```
this.Validate();  
this.booksBindingSource.EndEdit();  
this.booksTableAdapter.Update(this.dB_BOOKSDataSet  
.Books);
```

15. Аналогично для остальных форм добавить элементы типа TextBox

16. Проверить работу приложения.

17. На форму FormBooks добавить 5 компонентов типа Button см рисунок 6.

У 1-го компонента Button изменить свойства и метод:

Text Фильтр по текущему издательству;

В методе Click кнопки написать код:

```
int bb = dataGridView1.CurrentRow.Index;  
booksBindingSource.Filter = "Code_Publish = " +  
dataGridView1[4,bb].Value;.
```

У 2-го компонента Button изменить свойства и метод:

Text Фильтр по текущему названию книги.

В методе Click кнопки написать код:

```
int bb = dataGridView1.CurrentRow.Index;  
booksBindingSource.Filter = "Title_book = " +  
dataGridView1[1, bb].Value;.
```

У 3-го компонента Button изменить свойства и метод:

Text Фильтр по текущему автору.

В методе Click кнопки написать код:

```
int bb = dataGridView1.CurrentRow.Index;  
booksBindingSource.Filter = "Code_Author = " +  
dataGridView1[0, bb].Value;.
```

У 4-го компонента Button изменить свойства и метод:

Text Фильтр по количеству книг.

В методе Click кнопки написать код:

```
int bb = dataGridView1.CurrentRow.Index;  
booksBindingSource.Filter = "Pages = " +  
dataGridView1[3, bb].Value;.
```

У 5-го компонента Button изменить свойства и метод:

Text Снять фильтр.

В методе Click кнопки написать код:

```
booksBindingSource.Filter = "";
```

18. Аналогично для остальных форм добавить элементы типа Button, которые будут запускать фильтры по соответствующим значениям полей текущей записи в объекте Grid.

19. Проверить работу приложения.

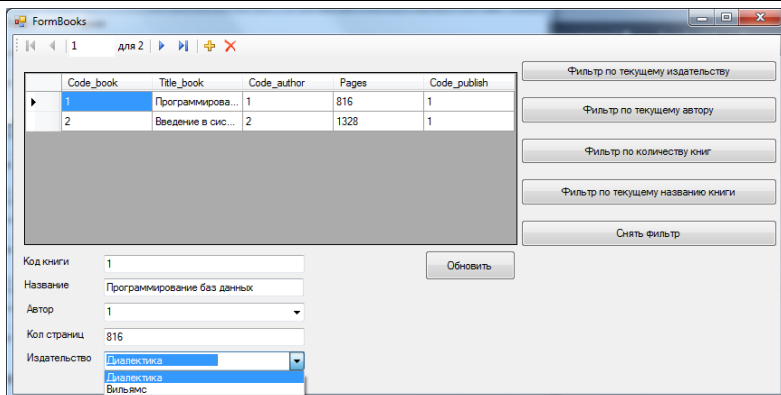


Рисунок 6 – Пример расположения компонентов на форме FormBooks

20. Создать форму, назвать FormProcedure.

21. Добавить на главной форме в меню пункт с названием «Работа с процедурами». В методе Click пункта меню написать код для запуска формы FormProcedure.

22. Зайти Tool → Choose Toolbox Items. Поставить галочки на элементах SqlCommand и SqlConnection, применить изменения см. рисунок 7.

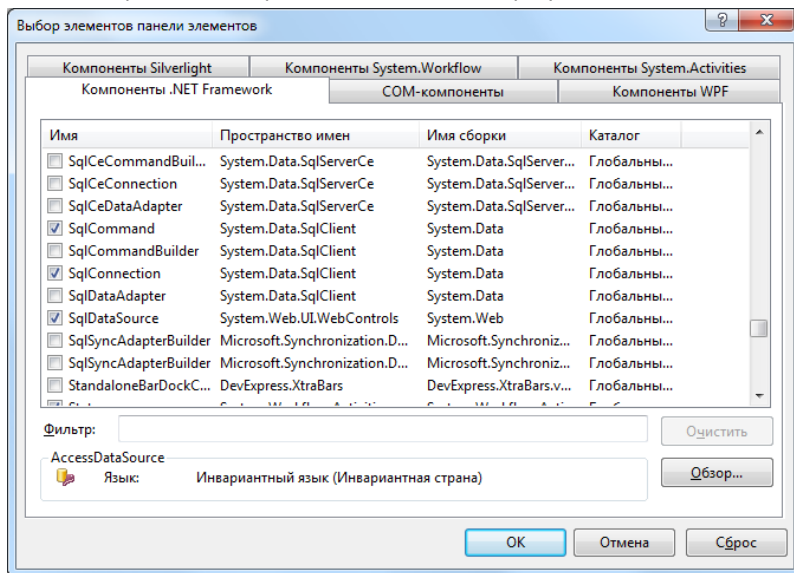


Рисунок 7 – Выбор элементов панели инструментов

23. Добавить на форму компонент SqlConnection и в свойстве ConnectionString выбрать DB_BOOK.mdf

24. Теперь можно подключить хранимую процедуру Count_purchases:

```
CREATE PROC [dbo].[Count_purchases]
@d1 SMALLDATETIME, @d2 SMALLDATETIME,
@c INT OUTPUT
AS
SELECT @c=count(Code_purchase) from Purchases WHERE Date_order
BETWEEN @d1 AND @d2
SET @c = ISNULL(@c,0)
GO
```

На форму FormProcedure добавить компонент SqlCommand. Изменить следующие его свойства:

Connection на SqlConnection1;
CommandType на StoredProcedure;
CommantText на Count_purchases.

25. У компонента SqlCommand1 выбрать свойство Parameters и в свойствах каждого входного параметра исправить свойство SqlDbType – на DateTime, а для выходного параметра свойство Value – Int. Также, если параметр созначением ReturnValue (параметр Direction) не создан, то необходимо создать его (он должен быть на самом верху) и задать ему имя @ReturnValue со свойством SqlDbType – Int.

26. На форму FormProcedure добавить 3 компонента типа TextBox (имена соответственно TextBox1, TextBox2, TextBox3) и 1 компонент типа Button. Рядом с каждым компонентом TextBox поставить Label и исправить их свойства Text соответственно на «Количество покупок за указанный период», «Введите дату начала периода», «Введите дату конца периода».

27. На кнопке поменять название на «Выполнить запрос». В методе Click кнопки написать следующий код:

```
int count_save;
sqlCommand1.Parameters["@d1"].Value =
Convert.ToDateTime(textBox1.Text);
sqlCommand1.Parameters["@d2"].Value =
Convert.ToDateTime(textBox2.Text);
sqlConnection1.Open();
sqlCommand1.ExecuteNonQuery();
sqlConnection1.Close();
count_save =
(int)sqlCommand1.Parameters["@ReturnValue"].Value;
textBox3.Text = Convert.ToString(count_save);
```

28. Проверить работу приложения.

ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ

На Visual C# 2010 создать новый проект, далее для индивидуальной БД, создать интерфейс, включающий все функции и процедуры, которые описаны в задании лабораторной работы №1.

Продемонстрируйте Вашу работу преподавателю!

КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Как осуществить подключение к базе данных в среде MS Visual Studio 2010?
2. Опишите процесс создания на форме запроса с параметрами?