

ЛАБОРАТОРНАЯ РАБОТА № 12.

Разработка приложения базы данных.

Цель: научиться создавать объекты баз данных, научиться создавать приложение для просмотра данных из базы данных.

Программное обеспечение: SQL Server Management Studio, Visual Studio

Задание: Разработать приложение для чтения данных из таблицы Абитуриенты.

Порядок выполнения:

1. Запустите среду разработки Visual Studio, параметры – C#.
2. Создайте проект **Windows Forms Application** с именем Kollege, в качестве места расположения укажите папку Programs.
3. На форму добавьте элементы, представленные в таблице:

Элемент управления	Свойство	Значение	Пояснение
StatusStrip StatusLabel	Name	lbRecordCount	Для вывода количества записей
	Text	-	-
DataGridView	Dock	Dock.Fill	Для отображения строк таблицы

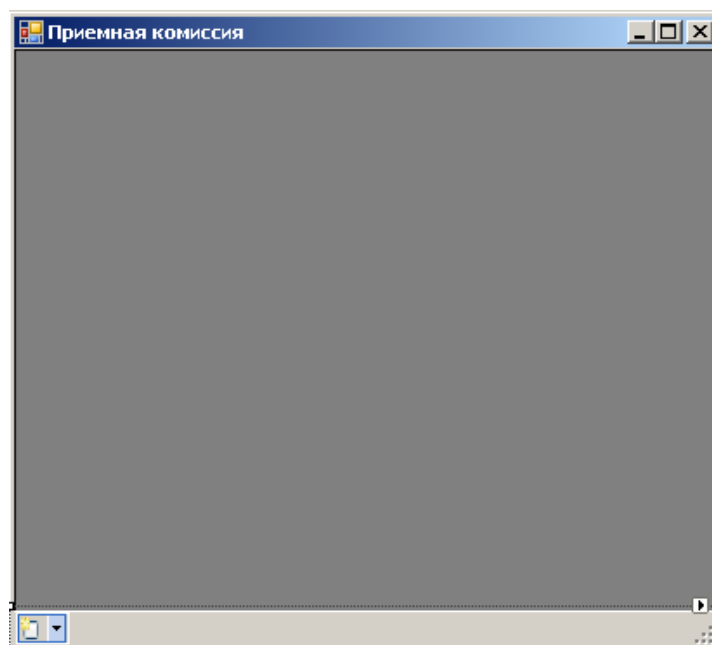


Рисунок 12.1 — Дизайн формы приложения

Для подключения к базе данных необходимо создать объект **Connection**. Строку соединения с базой данных будем хранить не в коде приложения, а во внешнем файле настроек проекта. Для этого откройте окно свойств проекта с помощью меню **Проект** —>

Свойства Kollege и выберите вкладку **Параметры**.

- Создайте новый параметр, который будет хранить строку соединения с базой данных (см. Рисунок 12.2):

Имя – ConString;

Тип – Строка подключения;

Область — Приложение;

Значение — с помощью диалогового окна настройте подключение к своей базе данных.

	Имя ▲	Тип	Область	Значение
►	ConString	(Строка подключения) ▼	Приложение	Data Source=EDU-MSSQL2008R2;Initial Catalog=_v110_Vladimirov;Integrated Security=True

Рисунок 12.2 — Задание свойств параметра

Закройте окно свойств проекта и перейдите в **Обозреватель решений**. В файле конфигурации приложения (app.config) появился параметр соединения с базой данных, как показано на рисунке 12.3.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="Приемная_комиссия.Properties.Settings.ConString" connectionString="Data
      providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

Рисунок 12.3 — Код файла конфигурации приложения

- Для формы создайте обработчик события Load, в тело которого необходимо ввести следующий код:

```
SqlConnection conn = new SqlConnection();
conn.ConnectionString = Properties.Settings.Default.ConString;
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = "Select * from Абитуриенты";
conn.Open();
SqlDataReader rdr = cmd.ExecuteReader();
DataTable dt = new DataTable();
for (int i = 0; i < rdr.FieldCount; i++)
{
    dt.Columns.Add(new DataColumn(rdr.GetName(i), rdr.GetFieldType(i)));
}
while (rdr.Read())
{
    DataRow row = dt.NewRow();
    for (int i = 0; i < rdr.FieldCount; i++)
    {
        row[i] = rdr.GetValue(i);
    }
    dt.Rows.Add(row);
}
```

```

}
conn.Close();
dataGridView1.DataSource = dt;
lbRecordCount.Text = "Количество записей: " + String.Format("{0}", dt.Rows.Count);

```

6. После запуска приложение имеет вид, как показано на рисунке 12.4.

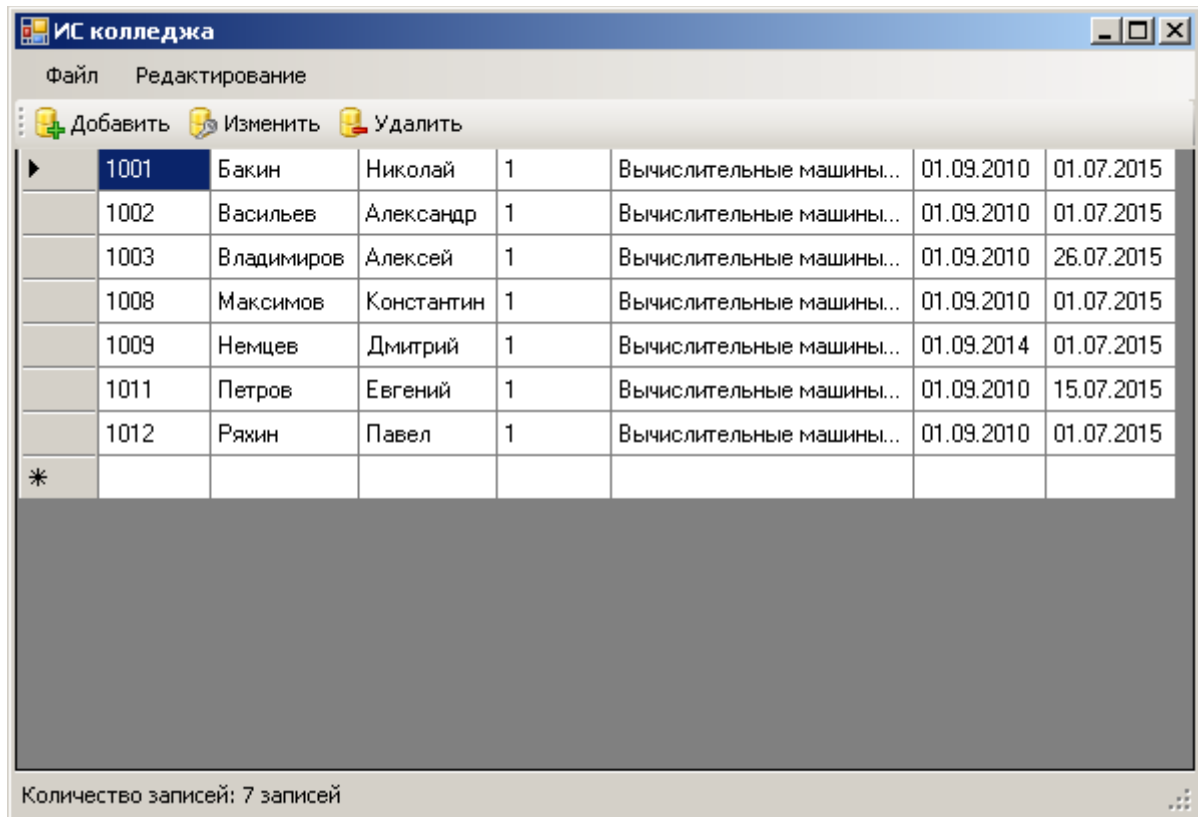


Рисунок 12.4 — Вывод данных в DataGridView

Задание 2: Модифицируйте созданное приложение таким образом, чтобы можно было просматривать все остальные таблицы базы данных. Для обеспечения выбора таблицы можно использовать следующие элементы управления:

- **TabControl** (см. Рисунок 12.5);
- **MenuStrip** (см. Рисунок 12.6);
- **ToolStrip** (см. Рисунок 12.7).

При выборе таблицы одним из предложенных выше способом, необходимо учитывать, что в программном коде на каждую таблицу необходимы свои собственные объекты: SqlCommand, SqlDataAdapter и DataSet.

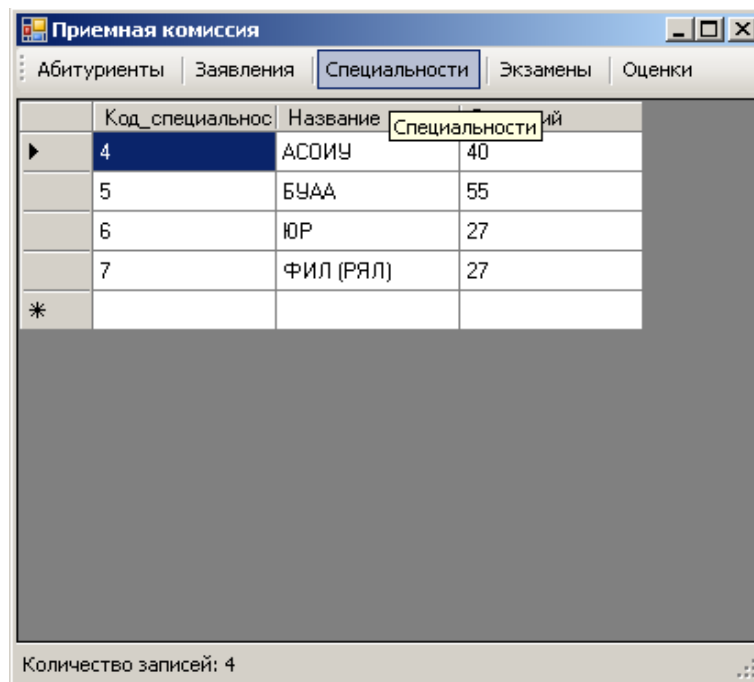


Рисунок 12.7 — Дизайн формы приложения с элементом **ToolStrip**