

УДК 004.4

РАЗРАБОТКА КЛАССА ДЛЯ РЕШЕНИЯ КОМПЛЕКСА ЗАДАЧ РАБОТЫ С БАЗАМИ ДАННЫХ

В.В. Коновальчук, 1 курс

Научный руководитель – Ю.М. Вишняков, к.т.н., доцент

Полесский государственный университет

Современные подходы к разработке приложений баз данных основываются на использовании посредников, к которым относится, в частности, технология ADO .NET. Эта технология предоставляет программисту набор классов, позволяющий решать широчайший спектр задач взаимодействия программного приложения с различными базами данных.

Набор классов, которые в основном приходится использовать программисту, не так велик. Это Connection, Command, DataReader, DataSet, DataAdapter. Поскольку использование данных классов предполагает их многократное использование в коде, было принято решение на их основе создать

новый класс ComplexQuery, способствующий упрощению работы с базой данных и более простому пониманию реализации.

Интерфейс класса:

```
public class ComplexQuery
{
//объявление основных переменных
private static string ConnectionString =
"Provider=Microsoft.Jet.OLEDB .4.0;Data Source=BD.mdb"; // строка подключения

private static DataSet ds; // виртуальная таблица данных

private static string QS; // значение базы данных формата string

private static double QD; // значение базы данных формата double

private static int QI; // значение базы данных формата int

public static void LoadComboBox(ComboBox bx, string text) // Загрузка объектов в ComboBox

public static void LoadTable(DataGridView dg,string query) // загрузка таблиц

public static void Query(string query) // основа для запросов к базе данных

public static void SearchDataGridView(DataGridView dg, string text) //Метод поиска

public static string QueryString(string query) //функция получения значения string

public static int QueryInt(string query)//функция получения значения int

public static double QueryD(string query) // функция получения значения double

public static void SortDate(DataGridView dg, DateTimePicker dt,DateTimePicker dt2,string pr) //сортировка по
дате таблицы, по выбранным значения DataSet
}
```

Основные задачи, решаемые разработанным классом:

- исполнение произвольного запроса;
- занесение данных в таблицу;
- занесение данных в ComboBox;
- занесение данных в переменные;
- адаптивный поиск;
- сортировка.

Примеры использования:

Загрузка таблицы:

```
ComplexQuery.LoadTable(Table, "Select * from Table");
```

Загрузка информации в ComboBox:

```
ComplexQuery.LoadComboBox(ComboBox,"Select Id as id, title as naim From Table");
(id, naim – обязательные атрибуты запроса):
```

Адаптивный поиск по таблице:

```
ComplexQuery.SearchDataGridView(Table, text);
```

Добавление данных:

```
ComplexQuery.Query(@"INSERT INTO Table (Title) VALUES" + "{0}",Text);
```

Удаление данных:

```
ComplexQuery.Query(@"Delete From Table Where ID = {0};", ID);
```

Изменение данных:

```
ComplexQuery.Query(@"Update Table set Title='{0}' where ID={1} ",Text, ID);
```

Получение данного:

```
ID = ComplexQuery.QueryInt(@"SELECT Max(ID) From Table");
```

Как показывают примеры использования данного класса, при разработке многофункциональных приложений сокращение объема кода может достигать 40–50%.