

Лабораторная работа №14

Передача данных HTML-формы на ASPX-страницу методами класса Request. Чтение и запись cookie-файлов

Выполните следующие задания

1. Создайте новый проект **ASP.NET Web Application**, добавьте в него страницу шаблона **HTML page**.
2. Задайте имя странице и введите HTML-разметку для создания формы. Для атрибута формы *method* установите значение *post*. Добавьте на форму 2 текстовых поля: *Имя* и *Группа* и кнопку: *Отправить* (тип *submit*). Для создания объектов используйте тег *input*. Задайте имена объектам.
3. Создайте еще одну Web-страницу шаблона **Web Form**. Задайте ей имя и укажите это имя для свойства *action* HTML-формы, созданной в п.2.
4. Для извлечения переданных данных используйте метод:

```
Request.Form.Get("имя объекта HTML-формы")
```

Полученное значение необходимо сохранить в объявленной переменной:

```
string Имя = Request.Form.Get("имя объекта HTML-формы");
```

5. Вывести результат с помощью объекта *Response*

```
Response.Write("Получено имя: " + Имя + @"<br />");
```

Выведите на экран переданное имя и название группы.

Задание для самостоятельного выполнения:

Создайте таблицы *Поезда* (Номер, Ст_отправления, Ст_назначения) и *Расписание* (Номер_поезда, Дата, Время). Создайте HTML-форму, на которой пользователь вводит станцию отправления и назначения, дату. На второй странице получает список всех поездов, которые отправляются в указанную дату по необходимому маршруту.

Следующая задача состоит в программировании возможности читать cookie-файлы при загрузке страницы и записывать cookie-файл при нажатии пользователем командной кнопки. Web-страница предлагает ввести данные, которые запоминает на сутки в cookie-файле. В течение суток, каждый раз вызывая данную страницу, в текстовых полях будут отображаться введенные сведения. Реализуем задачу:

1. Создадим проект **ASP.NET Web Application**

2. Добавим Web-форму

3. Поместим на форме 2 текстовых поля и кнопку

4. Запишем введенное в текстовые поля значения в cookie-файле

4.1. Создадим объект класса `HttpCookie`

```
HttpCookie Cookie1 = new HttpCookie("имя раздел");
```

4.2. Запишем в раздел пары: ключ – значение

```
Cookie1["ключ"] = значение;
```

Количество пар может только ограничиваться размерами файла

4.3. Установите срок хранения информации – 1 день

```
Cookie1.Expires = DateTime.Now.AddDays(1);
```

4.4. Добавьте раздел в cookie-файле

```
Response.Cookies.Add(Cookie1);
```

5. Считывание значений cookie-файлов

5.1. Объявить объект класса `HttpCookie`

```
HttpCookie Cookie1;  
Cookie1 = Request.Cookies.Get("имя раздела");
```

- 5.2. Если указанное cookie не найдено завершить работу, иначе считать значение в текстовое поле или любую переменную

```
TextBox1.Text = Cookie1["ключ"];
```

Задание для самостоятельного выполнения:

Добавьте в проект, созданный в лабораторной работе №13 возможность хранить в cookie-файлах введенные логин и пароль на третье Web-форме.

Отображение времени на форме с использованием технологии AJAX.

1. Поместите на форму объект **ScriptManager**, а также **UpdatePanel**, на который расположите объекты **Timer** и метку **Label**
2. В обработчике события **Tick** объекта **Timer** – присвоим метке значение текущего времени
3. При загрузке страницы задайте интервал времени для объекта **Timer**, равный одной секунде (свойство **Interval**)

Задание для самостоятельного выполнения:

Разместите таймер на четвертой Web-форме, созданной в лабораторной работе №12.

Формирование изображения методами класса Graphics и вывод его в Web-форму

1. Создадим проект **ASP.NET Web Application**
2. Добавим Web-форму
3. На форму добавим объект **Image**

4. Создадим точечное изображения размером 215x35 точек с глубиной цвета 24

```
Bitmap Рисунок = new Bitmap(215, 35, System.Drawing.Imaging.  
PixelFormat.Format24bppRgb);
```

5. Создаем новый объект класса *Graphics*

```
Graphics Графика = Graphics.FromImage(Рисунок);
```

6. Задаем заливку поверхности

```
Графика.Clear(Color.цвет);
```

7. Объявляем переменную и присваиваем ей текущую дату

8. Разворачиваем рисунок на 356 градусов по часовой стрелке

```
Графика.RotateTransform(356.0F);
```

9. Выводим на изображение текстовую строку

```
Графика.DrawString(Дата, new Font("Times New Roman", 14,  
FontStyle.Regular), Brushes.Red, x, y);
```

x,y – координаты левого верхнего угла строки

10. Определяем физический путь файла для текущего Web-узла, сохраняем изображение в файле каталога

```
Рисунок.Save(Request.PhysicalApplicationPath + "имя файла",  
System.Drawing.Imaging.ImageFormat.Jpeg);
```

11. Задаем цвет и ширину рамки рисунка

12. Указываем виртуальный путь к файлу изображения

```
Image1.ImageUrl = Request.ApplicationPath + "имя файла";
```

13. Освобождаем ресурсы

```
Рисунок.Dispose(); Графика.Dispose();
```

Задание для самостоятельного выполнения:

Для предотвращения вредоносного использования интернет-сервисов (в частности регистраций, автоматических отправок сообщений) используется процедура капча – компьютерное тестирование, используемое для того, чтобы определить кем является пользователь: человеком или

компьютером. Одна из задач капча – распознавание символов на картинке и последующего их ввода для проверки.

На форме регистрации пользователя, созданной в лабораторной работе №13, добавьте объект **Image**. Сгенерируйте число от 0 до 5 и выведите его на изображение (картинку):

```
var Генератор = new Random();  
int Число = Генератор.Next(0, 5);
```

Добавьте еще одно текстовое поле, куда пользователь будет вводит проверочный код. Если сгенерированный код совпадет с введенным, регистрация успешно завершится, иначе будет объявлено об ошибке.