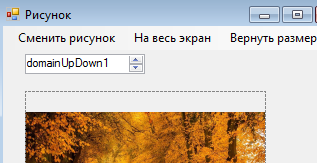
**Цель:** Ознакомление с новыми элементами управления в C#.

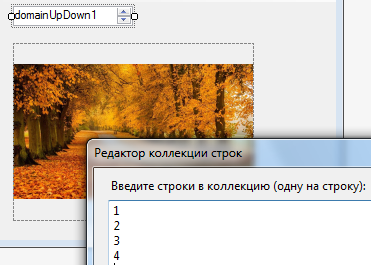
**Задание 1 Смена рисунков. Элемент DomainUpDown**

Подготовить форму вида:



Для которой:

* 1. Для объекта PictureBox1 установите свойство SizeMode=Zoom
  2. Для выполнения операций с рисунками используется горизонтальное меню: Сменить рисунок, На весь экран, Вернуть размер.
  3. Для смены рисунка по его номеру используется элемент DomainUpDown. Его вы найдете в группе элементов «Все формы Windows Forms». Заполните коллекцию его значений (свойство items) номерами от 1 до 4. Это будут номера выбираемых рисунков. Измените его свойство Text на значение 1.



* 1. Создайте для класса всей формы переменную num\_pict, чтобы она использовалась как глобальная:

string num\_pict;

* 1. Выполните двойной щелчок по свободному полю формы (т.е. вызовите обработчик private void Form1\_Load), для которого поместите код:

num\_pict = domainUpDown1.Text;

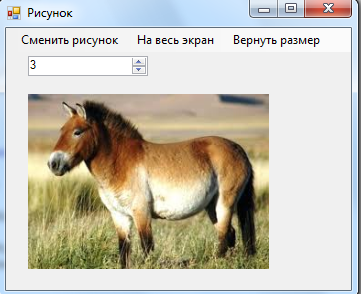
pictureBox1.ImageLocation = num\_pict + ".jpg";

* 1. Напишите обработку кнопки DomainUpDown (будет обрабатываться ее событие SelectedItemChanged), чтобы при выборе нового значения в списке менялось значение переменной num\_pict:

num\_pict = domainUpDown1.Text;

pictureBox1.ImageLocation = num\_pict + ".jpg";

(Для смены рисунка меняем свойство ImageLocation объекта PictureBox1).



Вы видите, что рисунок меняется в соответствии с номером из DomainUpDown1, т.к. в переменной num\_pict сохранилось именно это значение. В данном случае все рисунки имеют расширение jpg и находятся в папке debug (в текущей папке). Если у вас другое расположение рисунков, то внесите соответствующие изменения.

* 1. При выборе пункта меню «на весь экран» размеры текущей формы делаем максимальными и автоматически под них подгоняем размеры рисунка:

this.WindowState = FormWindowState.Maximized;

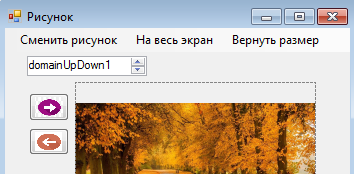
pictureBox1.SizeMode = PictureBoxSizeMode.AutoSize;

* 1. При выборе пункта меню «вернуть размер», размеры текущей формы делаем нормальными и выравниваем размеры рисунка значением Zoom:

this.WindowState = FormWindowState.Normal;

pictureBox1.SizeMode = PictureBoxSizeMode.Zoom;

* 1. Добавьте на форму две кнопки: для перехода к следующему рисунку и перехода к предыдущему рисунку.



Использовать для изображения кнопок будем не текстовые надписи, а изображения, для чего поменяйте свойства:



* 1. К кнопке с изображением «вперед» привяжите код, по которому вы увеличиваете номер рисунка и меняете его. Причем, смена номера рисунка выполняется только в том случае, если текущий номер рисунка еще не достиг значения 4:

int n;

if (num\_pict != "4") {

n = int.Parse(num\_pict);

n++;

num\_pict = n.ToString();

pictureBox1.ImageLocation = num\_pict + ".jpg";

}

else

{

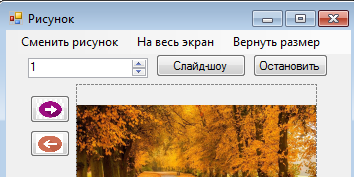
num\_pict = "1";

pictureBox1.ImageLocation = num\_pict + ".jpg";

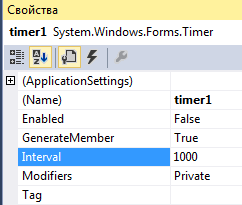
}

Переменная num\_pict строковая, поэтому перед тем, как увеличивать, приходится преобразовывать ее в число. Иначе мы не сможем ее увеличить. Метод ToString, наоборот, превращает числовое значение в строковое. Эти преобразования нам помогут правильно составить имя файла.

* 1. К кнопке с изображением «назад» разработайте код самостоятельно.
  2. Проверьте работу программы. Если вы уже отладили предыдущий фрагмент, то переходите к следующему.
  3. На нашей форме теперь не хватает только кнопок, по которым рисунки будут меняться в автоматическом режиме, по таймеру. Добавьте кнопки «Слайд-шоу» и «Остановить»:



* 1. Добавим компонент Timer («таймер») и настроим только его интервал=1000 (смена картинки раз в минуту). Если хотите, можете увеличить интервал.



* 1. По кнопке «Слайд-шоу» надо включить таймер:

timer1.Enabled=true;

* 1. По кнопке «Остановить» - выключить.



* 1. Щелкните по элементу «таймер» на форме для того, чтобы оформить программный код, который будет выполняться при каждом прерывании таймера:

int n;

n = int.Parse(num\_pict);

if (n==4) //если был 4-й рисунок, начинаем сначала

{ n = 0; }

n++;

num\_pict = n.ToString();

pictureBox1.ImageLocation = num\_pict + ".jpg";

* 1. Проверьте программу.