**Задание 1**

Разработайте проект, в котором одна машинка едет в направлении слева направо, вторая машинка едет справа налево. Для этого:

1. Установите на форме белый фон (либо подготовьте рисунки с прозрачным фоном)
2. Установите на форму таймер, включите его.
3. Скопируйте картинки с изображением машинок на белом фоне в папку Debug.
4. Самостоятельно напишите код таймера так, чтобы организовать движение двух машинок в противоположных направлениях (аналогично тому, как в предыдущей практической работе двигалась надпись).



**Задание 2**

Разработайте проект, в котором человечек шагает в левую сторону. Для этого:

1. Установите на форме белый фон, скопируйте картинки в папку Debug.
2. Установите на форму таймер, включите его, задайте интервал 300.
3. Объявите для всего класса формы переменную: int k;
4. Пусть при загрузке формы задается начальное значение: k = 0;
5. При прерывании по таймеру должен отрабатывать код, в котором запрограммировано движение человечка (при к=0 ссылаемся на один рисунок, при к=1 – на другой):

if (k == 0)

{pictureBox1.ImageLocation = "чел2.png";

k = 1;}

else

{pictureBox1.ImageLocation = "чел1.png";

k = 0;}

1. Добавьте движение объекта pictureBox1 вправо:

if (pictureBox1.Left > Width) pictureBox1.Left = -pictureBox1.Width;

pictureBox1.Left = pictureBox1.Left + 6;

**Задание 3**

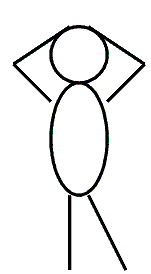
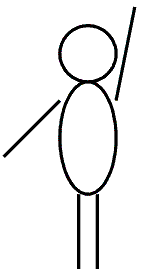
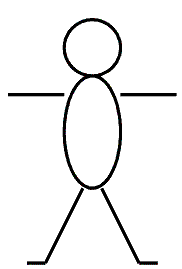
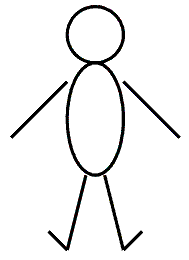
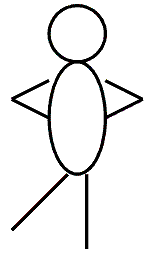
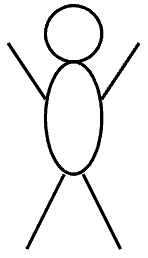
Разработайте проект, в котором изображается движение пляшущего человечка (он должен одновременно и передвигаться):



**Задание 4**

Разработайте проект, в котором изображается, как человечек выполняет зарядку.

1. Поскольку движений более 2-х, то вместо оператора if используйте switch.



switch (k)

{

case 0: pictureBox1.ImageLocation = "11.bmp"; break;

case 1: pictureBox1.ImageLocation = "12.bmp"; break;

case 2: pictureBox1.ImageLocation = "13.bmp"; break;

case 3: pictureBox1.ImageLocation = "14.bmp"; break;

case 4: pictureBox1.ImageLocation = "15.bmp"; break;

case 5: pictureBox1.ImageLocation = "16.bmp"; break;

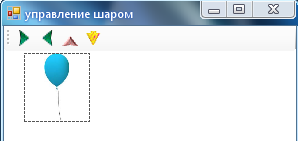
default: k = 0; break;

}

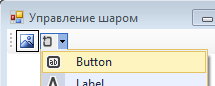
k++;

**Задание 5**

Создайте форму вида:



Где панель инструментов с кнопками создается при помощи элемента ToolStrip1. Вытяните панель. Выберите по стрелке Button и для этой кнопки установите свойство image.



Организуйте 8 кнопок:

* Строго вправо
* Строго влево
* Строго вверх
* Строго вниз
* По диагонали вниз вправо
* По диагонали вниз влево
* По диагонали вверх вправо
* По диагонали вверх влево

При загрузке формы установите шарик в начальное положение и задайте вектор передвижения по горизонтали dx и вертикали dy (глобальные переменные) такой, чтобы обеспечить начальное движение шарика строго вправо:

pictureBox1.Left = 0;

pictureBox1.Top = 0;

dx = 1; //на сколько изменяем положение по горизонтали

dy = 0; // на сколько изменяем положение по вертикали

Далее, организуя обработку кнопок на панели инструментов, имейте в виду, что для y положительные значения – это движение вниз, отрицательные – вверх. А для x – положительные значения – это движение вправо, отрицательные – влево. Если движение строго в одном направлении, то вторая координата должна быть нулевая, либо получается движение по диагонали. Для x нулевое положение – это крайнее левое, а для y – самое верхнее, но ниже верхнего края на высоту картинки.

При выборе кнопки «строго вниз»:

dy = 1;

dx = 0;

При выборе кнопки «строго вверх»:

dy = -1;

dx = 0;

При выборе кнопки «строго влево»:

dy = 0;

dx = -1;

При выборе кнопки «По диагонали вниз влево»:

dx = -1;

dy = 1;

…

Все остальные варианты реализуйте самостоятельно.

Обработчик для таймера:

pictureBox1.Left = pictureBox1.Left + dx;

pictureBox1.Top = pictureBox1.Top + dy;

Запустите программу и управляйте движением шарика по всем 8 кнопкам.

Добавим к предыдущему решению отражение от стенок формы. Т.е. в обработчике таймера допишите еще две строки:

if ((pictureBox1.Left == 0) || (pictureBox1.Left == Width - pictureBox1.Width)) dx = -dx;

if ((pictureBox1.Top == 0) || (pictureBox1.Top== Height - pictureBox1.Height)) dy = -dy;

В этих командах мы указали, что при достижении какой-либо границы мы меняем направление вектора движения (на ему обратное).

Запустив программу, вы увидите, что шарик будет отражаться от стенок слева и справа немного раньше времени, это происходит из-за того, что в применяемом рисунке имеются лишние поля. Если убрать лишний фон из рисунка, то отражение шарика будет выглядеть более правдоподобно.