## Лабораторная работа 6

### *Основы GDI*

### Подготовка к лабораторной работе

Для выполнения лабораторной работы необходимо ознакомиться (по литературе или в сети) и сделать краткое описание следующих функций и параметров, а также используемых событий.

1. Дескриптор контекста устройства и функции для его получения:
   * **e.Graphics**.

Класс [Graphics](http://msdn.microsoft.com/ru-ru/library/system.drawing.graphics(v=vs.100).aspx) предоставляет методы рисования на устройстве отображения.

* + **this.CreateGraphics()**;

Создает объект [Graphics](https://msdn.microsoft.com/ru-ru/library/system.drawing.graphics(v=vs.110).aspx) для элемента управления.

* + **Graphics.FromHwnd** ().

Создает новый объект [Graphics](https://msdn.microsoft.com/ru-ru/library/system.drawing.graphics(v=vs.110).aspx) из указанного дескриптора окна.

1. Выбор и создание кисти:
   * **System.Drawing.Brushes**

Кисти для каждого из стандартных цветов.

* + **SolidBrush;**

Определяет кисть одного цвета. Кисти используются для заливки графических фигур, таких как прямоугольники, эллипсы, круги, многоугольники и пути.

* + **System. Drawing. Drawing2D. HatchBrush;**

Определяет прямоугольную кисть с стиль штриховки, цвет фона и цвет переднего плана.

* + **System.Drawing.TextureBrush;**

Каждое свойство TextureBrush класс [Brush](https://msdn.microsoft.com/ru-ru/library/system.drawing.brush(v=vs.110).aspx) объекта, использующего изображение для заполнения внутренней части фигуры.

* + **LinearGradientBrush;**

Заполняет область линейным градиентом.

1. Выбор и создание карандаша.

Стандартные карандаши: **System.Drawing.Pens**

Создание карандаша. Например: p = **new Pen**(Color.Aquamarine, 3);

Ширина: p.**Width** = 5; Цвет: p.**Color** = Color.Coral;

Стиль: p.**DashStyle** = System.Drawing.Drawing2D.DashStyle.DashDot;

1. Функции рисования: **DrawArc, DrawLine, DrawEllipse, DrawRectangle, DrawPolygon, DrawImage, DrawString, FillEllipse, FillRectangle, FillPolygon, FillRegion**.
2. Прямоугольники, регионы, область рисования.

**Rectangle** r=new Rectangle(100,100,50,150);

**Region** reg = new Region(r);

Свойство **Clip** контекста устройства

1. Область отсечения и обновление окна: методы Button.**Refresh**() и **Invalidate();**
2. Цвет (три и четыре параметра)

**Color** col=Color.FromArgb(255,255,100);

Color col=Color.FromArgb(100,255,255,100);

* Метод **Clear** - обновление области рисования и заливка заданным цветом

## Задание

Для выполнения работы создать приложение на языке C#.

1. Придумать маленькую эмблему своей программы и нарисовать ее на всех кнопках. Картинка не должна исчезать после сворачивания или перекрытия окна.

private void DrawEmblem(Button button)

{

Graphics e = Graphics.FromHwnd(button.Handle);

e.DrawEllipse(new Pen(Color.Black, 3), 5, 5, 20, 20);

e.FillRectangle(new SolidBrush(Color.Blue), 10, 10, 5, 5);

e.FillRectangle(new SolidBrush(Color.Red), 10, 15, 5, 5);

e.FillRectangle(new SolidBrush(Color.Green), 15, 10, 5, 5);

e.FillRectangle(new SolidBrush(Color.Yellow), 15, 15, 5, 5);

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

if (t1)

{

foreach (Button button in listButtons)

DrawEmblem(button);

}

}

1. Нарисовать флаг олимпийских игр.

private void DrawFlag(Graphics graphics)

{

graphics.DrawRectangle(new Pen(Color.Black, 1), new Rectangle(0, 0, 310, 190));

graphics.DrawEllipse(new Pen(Color.Blue, 3), new Rectangle(40, 40, 70, 70));

graphics.DrawEllipse(new Pen(Color.Yellow, 3), new Rectangle(80, 80, 70, 70));

graphics.DrawEllipse(new Pen(Color.Black, 3), new Rectangle(120, 40, 70, 70));

graphics.DrawEllipse(new Pen(Color.Green, 3), new Rectangle(160, 80, 70, 70));

graphics.DrawEllipse(new Pen(Color.Red, 3), new Rectangle(200, 40, 70, 70));

graphics.DrawLine(new Pen(Color.Blue, 3), 40 + 70, 40 + 38, 40 + 70, 40 + 50);

graphics.DrawLine(new Pen(Color.Black, 3), 40 + 150, 40 + 38, 40 + 150, 40 + 50);

graphics.DrawLine(new Pen(Color.Green, 3), 40 + 158, 40 + 40, 40 + 166, 40 + 40);

graphics.DrawLine(new Pen(Color.Yellow, 3), 40 + 158 - 82, 40 + 40, 40 + 166 - 82, 40 + 40);

}

1. Создать программу «Игра», похожую на крестики-нолики. Игровое поле – клетки-прямоугольники (пять на пять) одного цвета. При щелчке на клетке цвет меняется (игрок номер 1), следующий щелчок совершает игрок номер 2 (цвет меняется на другой). Выигрывает тот, кто закрашивает 4 клетки подряд. При создании интерфейса для игры использовать текстурные и градиентные кисти, а также режим прозрачности.

private void pictureBox1\_Paint(object sender, PaintEventArgs e)

{

if (t2)

{

DrawFlag(e.Graphics);

}

if (t3)

{

Graphics graphics = e.Graphics;

Pen blackPen = new Pen(Color.Black, 1);

e.Graphics.FillRectangle(Brushes.White, new Rectangle(1, 1, 505, 505));

for (int i = 0; i < 6; ++i)

{

graphics.DrawLine(blackPen, 100 \* i + i, 0, 100 \* i + i, 504);

graphics.DrawLine(blackPen, 0, 100 \* i + i, 504, 100 \* i + i);

}

}

}

private void pictureBox1\_MouseClick(object sender, MouseEventArgs e)

{

if (t3 && start)

{

int xPos = e.X / 101;

int yPos = e.Y / 101;

if (mas[yPos, xPos] != 0)

return;

steps++;

Graphics graphics = pictureBox1.CreateGraphics();

if (firstPlayerStep)

{

//graphics.FillRectangle(new TextureBrush(greenBitmap), new Rectangle(xPos \* 101 + 1, yPos \* 101 + 1, 100, 100));

LinearGradientBrush greenGradientBrush = new LinearGradientBrush(new Point(xPos \* 100, yPos \* 100), new Point((xPos + 1) \* 100, (yPos + 1) \* 100), Color.FromArgb(255, 0, 255, 0), Color.FromArgb(0, 0, 100, 0));

graphics.FillRectangle(greenGradientBrush, new Rectangle(xPos \* 101 + 1, yPos \* 101 + 1, 100, 100));

mas[yPos, xPos] = 1;

label3.Text = "Second player step";

}

else

{

//graphics.FillRectangle(new TextureBrush(redBitmap), new Rectangle(xPos \* 101 + 1, yPos \* 101 + 1, 100, 100));

LinearGradientBrush redGradientBrush = new LinearGradientBrush(new Point(xPos \* 100, yPos \* 100), new Point((xPos + 1) \* 100, (yPos + 1) \* 100),

Color.FromArgb(255, 255, 0, 0), Color.FromArgb(0, 0, 100, 0));

graphics.FillRectangle(redGradientBrush, new Rectangle(xPos \* 101 + 1, yPos \* 101 + 1, 100, 100));

mas[yPos, xPos] = 2;

label3.Text = "First player step";

}

string str = getWinner();

if (str != "")

{

start = false;

label3.Text = str;

}

firstPlayerStep = !firstPlayerStep;

}

}

private string getWinner()

{

for (int i = 0; i < 5; ++i)

for (int j = 0; j < 2; ++j)

if (mas[i, j] > 0)

{

for (int k = j + 1; k < 5; ++k)

{

if (mas[i, k] != mas[i, j])

break;

else

if (k - j == 3)

if (mas[i, j] == 1)

return "First player win";

else

return "Second player win";

}

}

for (int i = 0; i < 2; ++i)

for (int j = 0; j < 5; ++j)

if (mas[i, j] > 0)

{

for (int k = i + 1; k < 5; ++k)

{

if (mas[k, j] != mas[i, j])

break;

else

if (k - i == 3)

if (mas[i, j] == 1)

return "First player win";

else

return "Second player win";

}

}

if(steps >= 25)

{

return "The game is over, nobody won";

}

return "";

}