

## ЛЕКЦИЯ 12

### ТЕМА: ГРАФИЧЕСКИЕ ВОЗМОЖНОСТИ ЯЗЫКА

1. Использование графики в языке VB
2. Объекты Line и Shape
3. Графические методы языка

#### 1. Использование графики в языке VB

Язык Visual Basic предоставляет три основных способа использования графики:

1) не входя в среду Visual Basic, можно нарисовать в каком-нибудь графическом редакторе то, что вам нужно, или найти на диске файл с подходящим рисунком и поместить его на форме или в графическом окне (PictureBox)

2) форма, графическое окно PictureBox и некоторые другие объекты обладают графическими методами, которые позволяют рисовать на объектах точки, отрезки, прямоугольники, окружности, эллипсы, дуги и секторы;

3) в окне инструментов ToolBox имеются два элемента управления: Line (отрезок) и Shape (фигура). Если их на этапе проектирования поместить на форму и придать им нужные свойства, то можно получить различные рисунки без программирования, в отличие от предыдущего способа.

Visual Basic 6.0 поддерживает (воспринимает и работает с ними) графические файлы следующих типов:

Расширение файла	Краткое описание
.BMP, .DIB	Фотографии и любые другие растровые изображения.
.ICO	Значки ( <b>пиктограммы, иконки</b> ) - очень маленькие
.CUR	Значки курсоров - очень маленькие
.WMF, .EMF	Метафайлы Windows - векторные изображения
.GIF	Растровые изображения, применяются в Интернете
.JPG, .JPEG	Растровые изображения (часто пониженного качества, зато экономные), применяются в Интернете

#### 2. Объекты Line и Shape

Рассмотрим второй способ использования графики в Visual Basic.

Возьмем в Toolbox объект **Line** (Линия) и проведем наискосок мышкой по форме. На форме разместится отрезок прямой. Мы можем его перемещать за маркеры или ухватившись острием мышиного курсора за середину. Поэкспериментируйте со следующими свойствами линии:

Свойство	Смысл
BorderWi	Толщина линии

dth	
BorderColor	Цвет линии
BorderStyle	Стиль линии (сплошная, штриховая и т.п.). Предварительно сделайте толщину = 1.
X1, Y1, X2, Y2	Координаты крайних точек отрезка

### 3. Графические методы

В дополнение к графическим элементам управления в Visual Basic существует набор графических методов:

Метод	Назначение
Circle	Рисует окружность, дугу или эллипс
Cls	Очищает область вывода
Line	Рисует линии
PaintPicture	Рисует (загружает) изображение
Point	Возвращает цвет точки
Print	Выводит текст
Pset	Размещает точку

В графических методах используются функции вычисления цвета RGB и QBColor. Рассмотрим сначала эти функции.

#### Функции цвета

Функция RGB возвращает числовое представление цвета (тип возвращаемого значения при этом long) в соответствии с интенсивностью трех цветовых составляющих: R (Red— красный), G (Green — зеленый) и B (Blue — синий). Интенсивность этих трех основных цветов задается шкалой чисел от 0 до 255. Синтаксис этой функции такой:

RGB (Red, Green, Blue)

где:

- Red — интенсивность красного цвета от 0 до 255;
- Green — интенсивность зеленого цвета от 0 до 255;
- Blue — интенсивность синего цвета от 0 до 255.

Для примера в табл. 11.2 показаны часто используемые цвета, разложенные по интенсивностям RGB-цветов.

**Таблица 12.1. Цвета в RGB-представлении**

Цвет	Интенсивность красного	Интенсивность зеленого	Интенсивность синего
Черный	0	0	0
Белый	255	255	255
Красный	255	0	0
Зеленый	0	255	0

Синий	0	0	255
Бирюзовый	0	255	255
Пурпурный	255	0	255
Желтый	255	255	0
Серый	192	192	192

Например, если вы хотите сделать фон формы бирюзовым, необходимо присвоить свойству BackColor формы следующее значение:

`Form.BackColor = RGB (0,255,255)`

Для удобства работы со стандартными цветами используется функция `QBColor`, которая возвращает числовое представление одного из 16 цветов, показанных в табл. 12.2.

**Таблица 12.2.** Цвета, возвращаемые функцией *QBColor*

Номер цвета	Цвет
0	Черный
1	Синий
2	Зеленый
3	Бирюзовый
4	Красный
5	Пурпурный
6	Желтый
7	Белый
8	Серый
9	Светло-синий (голубой)
10	Светло-зеленый (салатный)
11	Светло-бирюзовый
12	Светло-красный (алый)
13	Светло-пурпурный
14	Светло-желтый
15	Ярко-белый

Синтаксис этой функции выглядит очень просто:

`QBColor (NumberColor)`

где `NumberColor` — числовое значение (номер цвета), определяющее цвет (табл. 12.2).

Для того чтобы установить для формы бирюзовый цвет фона, достаточно написать следующий код:

`Form.BackColor = QBColor(3)`

Метод *Circle*

Метод `circle` используется следующим образом:

`object.Circle [Step] (x, y), radius, [color, start, end, aspect]`

где:

- `object` — объект, в котором применяется метод. Если объект не указан, то по умолчанию используется форма;

- **step** — ключ, определяющий привязку центра окружности, дуги или эллипса к координатам, возвращаемым свойствами **currentx** и **currentY** объекта;

- **x, y** — числа, определяющие координаты центра окружности, дуги или эллипса в единицах свойства **scaleMode** объекта;

- **radius** — число, определяющее радиус окружности, дуги или эллипса в единицах свойства **ScaleMode** объекта;

- **color** — значение типа **Long**, задающее цвет линии. Если параметр не указан, применяется значение свойства **ForeColor**. Для задания цвета можно использовать функции **RGB ()** или **QBColor**;

- **start, end** — при рисовании дуги или части эллипса задает позицию начала и конца дуги в радианах: от 2 пи до -2 пи радиан;

- **aspect** — задает коэффициент "эллиптичности" окружности. По умолчанию этот коэффициент равен 1.0, что соответствует окружности.

### Метод **Line**

Метод **Line** предназначен для рисования линий и имеет следующий синтаксис:

**object.Line [Step] (x1, y1) [Step] - (x2, y2), [color], [B] [F]**

где:

- **object** — объект, в котором применяется метод. Если объект не указан, то по умолчанию используется форма;

- **step** — ключ, определяющий привязку начала линии к координатам, возвращаемым свойствами **CurrentX** и **CurrentY** объекта;

- **x1, y1** — координаты начала линии. При отсутствии этих параметров начало привязывается к значениям свойств **CurrentX** и **CurrentY** объекта;

- **step** — ключ, определяющий привязку координат конца линии к началу, то есть координаты конца заданы относительно координат начала линии;

- **x2, y2** — координаты конца линии;

- **color** — задает цвет линии в палитре **RGB**. Если параметр не указан, то используется свойство **ForeColor** объекта;

- **в** — задает рисование прямоугольника, при этом координаты означают координаты левого верхнего и правого нижнего углов;

**F**— задает заполнение прямоугольника цветом линии рисования.