Часть графического изображения, которое не изменяет содержимое экрана называется *прозрачным*. Функция **DrawIcon**, которая может создавать изображения может содержать прозрачные части. Можно также получить функции при помощи **BitBlt**, но есть и некоторые дополнительные советы:

```
procedure DrawTransparentBitmap(DC: HDC; hBmp : HBITMAP ; xStart: integer; yStart : integer; cTransparentColor : COLORREF);  var bm: BITMAP; cColor: COLORREF; bmAndBack, bmAndObject, bmAndMem, bmSave:  HBITMAP; bmBackOld, bmObjectOld, bmMemOld, bmSaveOld: HBITMAP; hdcMem, hdcBack, hdcObject, hdcTemp, hdcSave:  HDC; ptSize: TPOINT; begin hdcTemp := CreateCompatibleDC(dc); SelectObject(hdcTemp, hBmp); // Выбор точечного изображения GetObject(hBmp, sizeof(BITMAP), @bm); ptSize.x := bm.bmWidth; // получить ширину изображения ptSize.y := bm.bmHeight; // получить высоту изображения DPtoLP(hdcTemp, ptSize, 1); // Преобразование из устройства // в логические точки // Создать DC для хранения временных данных. hdcBack := CreateCompatibleDC(dc); hdcObject := CreateCompatibleDC(dc); hdcMem := CreateCompatibleDC(dc); hdcSave := CreateCompatibleDC(dc); // Создать изображение для каждого DC. // Монохромный DC bmAndBack := CreateBitmap(ptSize.x, ptSize.y, 1, 1, nil); // Монохромный DC bmAndObject := CreateBitmap(ptSize.x, ptSize.y, 1, 1, nil); bmAndMem := CreateCompatibleBitmap(dc, ptSize.x, ptSize.y); bmSave := CreateCompatibleBitmap(dc, ptSize.x, ptSize.y); // Каждый DC должен выбрать объект изображения // для хранения пикселов. bmBackOld := SelectObject(hdcBack, bmAndBack); bmObjectOld := SelectObject(hdcObject, bmAndObject); bmMemOld := SelectObject(hdcMem, bmAndMem); bmSaveOld := SelectObject(hdcSave, bmSave); // Установить нужный режим отображения. SetMapMode(hdcTemp, GetMapMode(dc)); // Сохраняем битмап здесь, но он будет переписан. BitBlt(hdcSave, 0, 0, ptSize.x, ptSize.y, hdcTemp, 0, 0, SRCCOPY); // Установить цвет фона источника DC cColor := SetBkColor(hdcTemp, cTransparentColor); // Создать маску при помощи BitBlt // из исходного битмапа в монохромный. BitBlt(hdcObject, 0, 0, ptSize.x, ptSize.y, hdcTemp, 0, 0, SRCCOPY); // Установить цвет фона исходного DC // назад в оригинальный. SetBkColor(hdcTemp, cColor); // Создаем обратную маску. BitBlt(hdcBack, 0, 0, ptSize.x, ptSize.y, hdcObject, 0, 0, NOTSRCCOPY); // Копировать фон главного DC в конечное изображение. BitBlt(hdcMem, 0, 0, ptSize.x, ptSize.y, dc, xStart, yStart, SRCCOPY); // Поместить маску, куда будет помещен битмап. BitBlt(hdcMem, 0, 0, ptSize.x, ptSize.y, hdcObject, 0, 0, SRCAND); // Маска цветного пикселя на битмапе. BitBlt(hdcTemp, 0, 0, ptSize.x, ptSize.y, hdcBack, 0, 0, SRCAND); // XOR битмап с фоном на конечном DC. BitBlt(hdcMem, 0, 0, ptSize.x, ptSize.y, hdcTemp, 0, 0, SRCPAINT); // Копировать конечное изображение на экран. BitBlt(dc, xStart, yStart, ptSize.x, ptSize.y, hdcMem, 0, 0, SRCCOPY); // Помещаем оригинал обратно. BitBlt(hdcTemp, 0, 0, ptSize.x, ptSize.y, hdcSave, 0, 0, SRCCOPY); // Удалить из памяти Bitmapы. DeleteObject(SelectObject(hdcBack, bmBackOld)); DeleteObject(SelectObject(hdcObject, bmObjectOld)); DeleteObject(SelectObject(hdcMem, bmMemOld));
```

DeleteObject(SelectObject(hdcSave, bmSaveOld)); // Удалить из памяти DC. DeleteDC(hdcMem); DeleteDC(hdcBack); DeleteDC(hdcObject); DeleteDC(hdcSave); DeleteDC(hdcTemp); end;

Ниже приводится пример применения функции **DrawTransparentBitmap**.

procedure TForm1.Button1Click(Sender: TObject); begin DrawTransparentBitmap(Form1.Canvas.Handle, Image1.Picture.Bitmap.Handle, 10, 10, clWhite); end;