```
//Tomasz Lubiński (C)2009
// http://www.algorytm.org
//Uporzadkowane rozpraszanie bledow - tablice Bayer'a
//---------------------------------------------------------------------
   -----------
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "math.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------
   -----------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------
   -----------

typedef struct Pixel
{
   Byte b;
   Byte g;
   Byte r;
   Byte reserved;
} TPixel;


/* tablica Bayer'a */
int bayer[16][16];

/* funkcje do wygenerowanie tablicy Bayer'a */
int getX(int i, int level, int shift)
{
   int result = ((i+1) % 2);

   if (level == 1)
   {
      return result + shift;
   }

   return getX(i/4, level-1, shift + result * pow(2, level-1));
}

int getY(int i, int level, int shift)
{
```

```
    int result = (((i+3) % 4) / 2);

    if (level == 1)
    {
        return result + shift;
    }

    return getY(i/4, level-1, shift + result * pow(2, level-1));
}

void prepareBayerTable(int level)
{
    int size, i, x, y;

    size = pow(2, level);

    for (i=0; i<size*size; i++)
    {
        x = getX(i, level, 0);
        y = getY(i, level, 0);

        bayer[x][y] = (i+1);
    }
}

void __fastcall TForm1::Button1Click(TObject *Sender)
{
int i,j,bayerSize;
float p;
TPixel *pixelOrg, *pixelNew, white, black;

//przygotuj wartosci bialy i czarny
white.b = 255;
white.g = 255;
white.r = 255;
black.b = 0;
black.g = 0;
black.r = 0;

//przygotuj obrazy wynikowe
Image2->Canvas->Brush->Color = clWhite;
Image2->Canvas->Rectangle(0, 0, Image2->Width, Image2->Height);
Image2->Picture->Bitmap->PixelFormat = pf32bit;

Image3->Canvas->Brush->Color = clWhite;
Image3->Canvas->Rectangle(0, 0, Image3->Width, Image3->Height);
Image3->Picture->Bitmap->PixelFormat = pf32bit;
```

```
//przygotuj format obrazu zrodlowego
Image1->Picture->Bitmap->PixelFormat = pf32bit;

//pobierz prog
p = StrToInt(Form1->Edit1->Text);

//zwykle progowe
for (j=0; j<230; j++){
        pixelOrg = (TPixel *)Image1->Picture->Bitmap->ScanLine[j];
        pixelNew = (TPixel *)Image2->Picture->Bitmap->ScanLine[j];
        for (i=0; i<248; i++){
                if (p > pixelOrg->r)
                        *pixelNew = black;
                else
                        *pixelNew = white;
                pixelOrg++;
                pixelNew++;
                }
        }
//uporzadkowane rozpraszanie bledow
//przygotuj tablice Bayer'a
prepareBayerTable(ComboBox1->ItemIndex + 1);
bayerSize = pow(2, ComboBox1->ItemIndex + 1);
p /= (float)(bayerSize*bayerSize);
for (j=0; j<230; j++){
        pixelOrg = (TPixel *)Image1->Picture->Bitmap->ScanLine[j];
        pixelNew = (TPixel *)Image3->Picture->Bitmap->ScanLine[j];
        for (i=0; i<248; i++){
                if (p * bayer[i % bayerSize][j % bayerSize] >
                   pixelOrg->r) {
                        *pixelNew = black;
                } else {
                        *pixelNew = white;
                }

                pixelOrg++;
                pixelNew++;
        }
}
}
//---------------------------------------------------------------------
   -----------
void __fastcall TForm1::Label6Click(TObject *Sender)
{
 ShellExecute(Application->Handle,
             "open",
             "http://www.algorytm.org",
             NULL,
```

```
            NULL,
            SW_NORMAL);
}
//---------------------------------------------------------------------
   ------------
```