



Nazwa implementacji: Zegar binarny

Autor:

Piotr Fiorek

Opis implementacji: Implementacja przedstawia zasadę działania zegara decybinarnego.

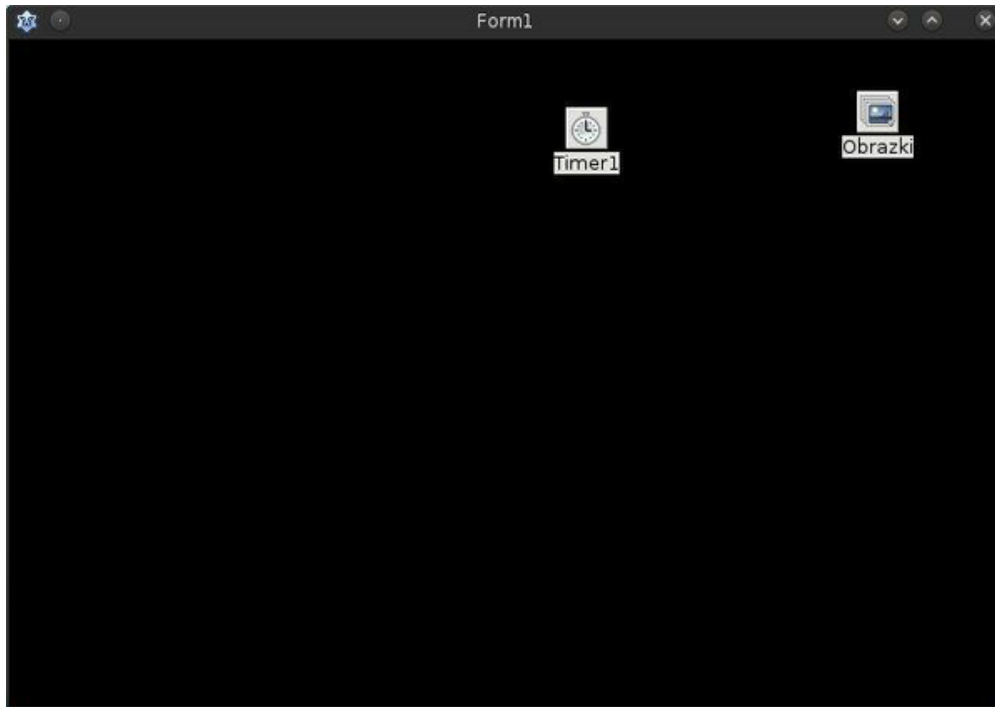
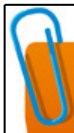
Zaprojektuj zegar binarny, startowany bieżącym czasem systemowym.

Zegar binarny

	dziesiątne godzin	jedności godzin	dziesiątne minut	jedności minut	dziesiątne sekund	jedności sekund		
wartości								
8								
4								
2								
1								
suma pokazuje godzinę	1	3	:	0	1	:	5	7

Zegar binarny składa się z macierzy krutek – cztery rzędy, sześć kolumn. Każda kolumna reprezentuje inną część godziny. Pierwsza kolumna pokazuje dziesiątą część godziny, druga – jednostkową, trzecia – dziesiątą część minuty, czwarta – jednostkową minuty, piąta – dziesiątą sekunda i szósta – jednostkową sekundy. Rzędy z kolei reprezentują wartości – od góry po kolei 8, 4, 2 i 1. Aby odczytać godzinę na takim zegarze, patrzemy po kolei na kolumny i sumujemy wartości lampek, które są w danej kolumnie zapalone.

Okno Formularza:



Należy dodać obiekty QTimer oraz TPictureList. W przypadku TPictureList należy określić jego rozmiary (powinien być taki sam jak rozmiar pól obrazkowych, które zamierzamy stworzyć, w przeciwnym wypadku obrazki będą przycięte) oraz po dwukrotnym kliknięciu w obiekt listę obrazków przedstawiających lampkę zgaszoną, oraz zapaloną.

Aby stworzyć procedurę, która będzie wywoływana z każdym „tyknięciem” timera wystarczy kliknąć na niego dwa razy.

W oknie „Object Inspector” można również ustawić nazwy poszczególnych elementów. W tej implementacji używamy nazw:

- „Obrazki” dla obiektu TPictureList;
- "Timer1" dla obiektu QTimer;

Kod:

```
unit zegar;
```

```
{$mode objfpc}{$H+}
```

```
interface
```

```
uses
```

```
Classes, SysUtils, FileUtil, LResources, Forms, Controls, Graphics, Dialogs,  
ExtCtrls, dateutils;
```

2



```
type
{ TForm1 }

TForm1 = class(TForm)
  // elementy umieszczone na ekranie
  Background: TImage;
  Obrazki: TImageList;
  Timer1: TTimer;
  // deklaracje procedur
  procedure FormCreate(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
  procedure Tik();
private
  // deklaracje zmiennych w programie
  Czas: TDateTime; // bieżący czas
  Godziny, Minuty, Sekundy: Integer; // zmienne przechowujące liczniki czasu
  Cyfra10Godz: array[1..4] of TImage; // tablica przechowująca kratki dziesiątek godzin
  Cyfra01Godz: array[1..4] of TImage; // tablica przechowująca kratki jednostek godzin
  Cyfra10Min: array[1..4] of TImage; // tablica przechowująca kratki dziesiątek minut
  Cyfra01Min: array[1..4] of TImage; // tablica przechowująca kratki jednostek minut
  Cyfra10Sek: array[1..4] of TImage; // tablica przechowująca kratki dziesiątek sekund
  Cyfra01Sek: array[1..4] of TImage; // tablica przechowująca kratki jednostek sekund
public
  { public declarations }
end;

var
  Form1: TForm1;
  i: Integer; // zmienna używana jako iterator

implementation

{ TForm1 }

procedure TForm1.FormCreate(Sender: TObject);
begin
  Czas:=Now(); // pobranie aktualnego czasu do zmiennej
  Godziny:= HourOf(Czas); // pobranie godziny z aktualnego czasu
  Minuty:= MinuteOf(Czas); // pobranie minut z aktualnego czasu
  Sekundy:= SecondOf(Czas); // pobranie sekund z aktualnego czasu

  { dziesiątki godziny }
  for i:=1 to 4 do
  begin
    Cyfra10Godz[i]:= TImage.Create(self); // stworzenie kratki
    Cyfra10Godz[i].Parent:= self; // przypisanie kratki do okna
    Cyfra10Godz[i].Width:= 100; // określenie szerokości kratki
    Cyfra10Godz[i].Height:= 100; // określenie wysokości kratki
    Obrazki.GetBitmap(0, Cyfra10Godz[i].Picture.Bitmap); // przypisanie obrazka do kratki
    Cyfra10Godz[i].Top:= (i*10) + (i-1)*100; // określenie położenia kratki od górnej krawędzi okna
    Cyfra10Godz[i].Left:= 10; // określenie położenia kratki względem lewej krawędzi okna
  end;

  { 1 godziny }
```

3

**KAPITAŁ LUDZKI**
NARODOWA STRATEGIA SPÓJNOŚCI**UNIA EUROPEJSKA**
EUROPEJSKI
FUNDUSZ SPOŁECZNY



```
for i:=1 to 4 do
begin
  Cyfra01Godz[i]:= TImage.Create(self);
  Cyfra01Godz[i].Parent:= self;
  Cyfra01Godz[i].Width:= 100;
  Cyfra01Godz[i].Height:= 100;
  Obrazki.GetBitmap(0, Cyfra01Godz[i].Picture.Bitmap);
  Cyfra01Godz[i].Top:= (i*10) + (i-1)*100;
  Cyfra01Godz[i].Left:= 120;
end;

{ 10 minuty }
for i:=1 to 4 do
begin
  Cyfra10Min[i]:= TImage.Create(self);
  Cyfra10Min[i].Parent:= self;
  Cyfra10Min[i].Width:= 100;
  Cyfra10Min[i].Height:= 100;
  Obrazki.GetBitmap(0, Cyfra10Min[i].Picture.Bitmap);
  Cyfra10Min[i].Top:= (i*10) + (i-1)*100;
  Cyfra10Min[i].Left:= 230;
end;

{ 1 minuty }
for i:=1 to 4 do
begin
  Cyfra01Min[i]:= TImage.Create(self);
  Cyfra01Min[i].Parent:= self;
  Cyfra01Min[i].Width:= 100;
  Cyfra01Min[i].Height:= 100;
  Obrazki.GetBitmap(0, Cyfra01Min[i].Picture.Bitmap);
  Cyfra01Min[i].Top:= (i*10) + (i-1)*100;
  Cyfra01Min[i].Left:= 340;
end;

{ 10 sekundy }
for i:=1 to 4 do
begin
  Cyfra10Sek[i]:= TImage.Create(self);
  Cyfra10Sek[i].Parent:= self;
  Cyfra10Sek[i].Width:= 100;
  Cyfra10Sek[i].Height:= 100;
  Obrazki.GetBitmap(0, Cyfra10Sek[i].Picture.Bitmap);
  Cyfra10Sek[i].Top:= (i*10) + (i-1)*100;
  Cyfra10Sek[i].Left:= 450;
end;

{ 1 sekundy }
for i:=1 to 4 do
begin
  Cyfra01Sek[i]:= TImage.Create(self);
  Cyfra01Sek[i].Parent:= self;
  Cyfra01Sek[i].Width:= 100;
  Cyfra01Sek[i].Height:= 100;
  Obrazki.GetBitmap(0, Cyfra01Sek[i].Picture.Bitmap);
```





```
Cyfra01Sek[i].Top:= (i*10) + (i-1)*100;  
Cyfra01Sek[i].Left:= 560;  
end;  
  
Tik();  
end;  
  
// funkcja wywoływana co sekundę przez Timer  
procedure TForm1.Timer1Timer(Sender: TObject);  
begin  
    Sekundy:= Sekundy + 1; // zwiększenie liczby sekund  
    Tik(); // wywołanie funkcji przeliczającej czas i zmieniającej stan zegara  
end;  
  
// funkcja przeliczająca czas i zmieniająca stan zegara  
procedure TForm1.Tik();  
var  
    godz10, godz01, min10, min01, sek10, sek01: Integer; // zmienne pomocnicze  
begin  
    if Sekundy > 59 then // sprawdź czy minęła pełna minuta  
        begin  
            Sekundy:= 0; // wyzeruj licznik sekund  
            Minuty:= Minuty + 1; // zwiększ licznik minut  
            if Minuty > 59 then // sprawdź czy minęła pełna godzina  
                begin  
                    Minuty:= 0; // wyzeruj licznik minut  
                    Godziny:= Godziny + 1; // zwiększ licznik godzin  
                    if Godziny > 23 then Godziny:= 0; // jeśli minęła pełna doba wyzeruj licznik godzin  
                end;  
            end;  
        end;  
  
    godz10:= Godziny div 10; // pobierz dziesiątą część godziny  
    if (godz10 and 2) <> 0 then // funkcja logiczna sprawdzająca czy bit reprezentujący cyfrę dwa jest aktywny  
        Obrazki.GetBitmap(1, Cyfra10Godz[3].Picture.Bitmap) // jeśli tak - zapal lampkę  
    else  
        Obrazki.GetBitmap(0, Cyfra10Godz[3].Picture.Bitmap); // jeśli nie - zgaś lampkę  
    if (godz10 and 1) <> 0 then Obrazki.GetBitmap(1, Cyfra10Godz[4].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra10Godz[4].Picture.Bitmap);  
  
    godz01:= Godziny mod 10; // pobierz jedności minut  
    if (godz01 and 8) <> 0 then Obrazki.GetBitmap(1, Cyfra01Godz[1].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Godz[1].Picture.Bitmap);  
    if (godz01 and 4) <> 0 then Obrazki.GetBitmap(1, Cyfra01Godz[2].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Godz[2].Picture.Bitmap);  
    if (godz01 and 2) <> 0 then Obrazki.GetBitmap(1, Cyfra01Godz[3].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Godz[3].Picture.Bitmap);  
    if (godz01 and 1) <> 0 then Obrazki.GetBitmap(1, Cyfra01Godz[4].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Godz[4].Picture.Bitmap);  
  
    min10:= Minuty div 10;  
    if (min10 and 4) <> 0 then Obrazki.GetBitmap(1, Cyfra10Min[2].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra10Min[2].Picture.Bitmap);  
    if (min10 and 2) <> 0 then Obrazki.GetBitmap(1, Cyfra10Min[3].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra10Min[3].Picture.Bitmap);  
    if (min10 and 1) <> 0 then Obrazki.GetBitmap(1, Cyfra10Min[4].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra10Min[4].Picture.Bitmap);  
  
    min01:= Minuty mod 10;  
    if (min01 and 8) <> 0 then Obrazki.GetBitmap(1, Cyfra01Min[1].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Min[1].Picture.Bitmap);  
    if (min01 and 4) <> 0 then Obrazki.GetBitmap(1, Cyfra01Min[2].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Min[2].Picture.Bitmap);  
    if (min01 and 2) <> 0 then Obrazki.GetBitmap(1, Cyfra01Min[3].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Min[3].Picture.Bitmap);  
    if (min01 and 1) <> 0 then Obrazki.GetBitmap(1, Cyfra01Min[4].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Min[4].Picture.Bitmap);
```





```
sek10:= Sekundy div 10;
if (sek10 and 4) <> 0 then Obrazki.GetBitmap(1, Cyfra10Sek[2].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra10Sek[2].Picture.Bitmap);
if (sek10 and 2) <> 0 then Obrazki.GetBitmap(1, Cyfra10Sek[3].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra10Sek[3].Picture.Bitmap);
if (sek10 and 1) <> 0 then Obrazki.GetBitmap(1, Cyfra10Sek[4].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra10Sek[4].Picture.Bitmap);

sek01:= Sekundy mod 10;
if (sek01 and 8) <> 0 then Obrazki.GetBitmap(1, Cyfra01Sek[1].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Sek[1].Picture.Bitmap);
if (sek01 and 4) <> 0 then Obrazki.GetBitmap(1, Cyfra01Sek[2].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Sek[2].Picture.Bitmap);
if (sek01 and 2) <> 0 then Obrazki.GetBitmap(1, Cyfra01Sek[3].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Sek[3].Picture.Bitmap);
if (sek01 and 1) <> 0 then Obrazki.GetBitmap(1, Cyfra01Sek[4].Picture.Bitmap) else Obrazki.GetBitmap(0, Cyfra01Sek[4].Picture.Bitmap);
end;

initialization
  {$I zegar.lrs}

end.
```

UWAGA: Podczas zapisywania projektu nie wolno zapisywać plików ".pas" (plik z kodem) oraz ".lpi" (plik projektu) pod tymi samymi nazwami.

