

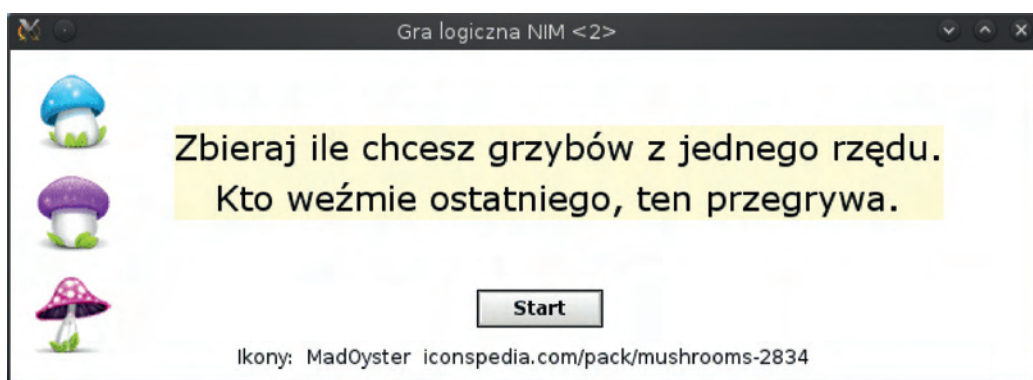
# Implementacja Modułu B2.8

## Gra logiczna NIM

✎ Stanisław Ubermanowicz, Piotr Fiorek

**Z**aprojektuj grę logiczną NIM. Program losuje w każdym rzędzie od 1 do 10 grzybów. Gracz rywalizuje z komputerem. Podczas ruchu można brać dowolną liczbę grzybów, ale tylko z jednego rzędu. Przegrywa ten, kto musi zabrać ostatniego grzyba. Strategia wygranej polega na tym, aby utrzymywać parzystość grup binarych.

### Widok po uruchomieniu



### Widok w trakcie gry



## Widok okna projektu



## Projektowanie obiektów przykładowej implementacji

Na formularzu umieszczamy obiekty tekstowe TLabel, które będą pojawiały się w zależności od stanu gry. Na jednym z pól umieszczamy tekst instrukcji dla gracza, inne będzie służyło do informowania, kto wygrał. Dodajemy obiekt kontenera obrazków TImageList, obiekt przycisku TButton i teksty instrukcji dla gracza.

W formularzu właściwości Object Inspector ustawiamy nazwy poszczególnych elementów:

- » „Link” dla obiektu typu TLabel prezentującego adres internetowy;
- » „Opis” dla obiektu typu TLabel przedstawiającego opis gry;
- » „Info” dla obiektu typu TLabel prezentującego informację o wygranej lub przegranej;
- » „Obrazki” dla obiektu typu TImageList przechowującego obrazki kolejnych grzybków;
- » „Start” dla obiektu typu TButton będącego przyciskiem rozpoczynającym grę;

**UWAGA:** Do zapisywania implementacji nie wolno używać tych samych nazw dla pliku z kodem (\*.pas) oraz dla pliku projektu (\*.lpi’).

## Kod źródłowy implementacji

{ Część elementów kodu generowana jest automatycznie }

```
unit gra;
{$mode objfpc}{$SH+}
interface
uses
  Classes, SysUtils, FileUtil, LResources, Forms, Controls, Graphics, Dialogs,
  ExtCtrls, StdCtrls;
type
  { TForm1 }
  TForm1 = class(TForm)
    Link: TLabel;
    Start: TButton;
    Opis: TLabel;
    Obrazki: TImageList;
    Info: TLabel;
  procedure StartClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure Rzdad1Click(Sender: TObject);
  end;
  { Zmienne reprezentujące obiekty na formularzu }
  { Procedury obsługujące grę }
```

```

procedure Rząd2Click(Sender: TObject);
procedure Rząd3Click(Sender: TObject);
procedure Rząd1Action(grzyb: Integer);
procedure Rząd2Action(grzyb: Integer);
procedure Rząd3Action(grzyb: Integer);
procedure SprawdźKoniec();
procedure PCMove();

privat
    ile1, ile2, ile3: Integer;
    tura: Integer;
    koniec: Boolean;
    Grzyb1: array[1..10] of TImage;
    Grzyb2: array[1..10] of TImage;
    Grzyb3: array[1..10] of TImage;

    { private declarations }
    { Wskazują, ile grzybów jest w danej chwili w rządzie }
    { Zmienna, która określa liczbę tur i to, czy był ruch }
    { Zmienna określająca czy nastąpił koniec gry }
    { Lista obiektów pierwszego rzędu }
    { Lista obiektów drugiego rzędu }
    { Lista obiektów trzeciego rzędu }

public
end;
    { public declarations }

var
    Form1: TForm1;
    i: Integer;
    { Zmienna pomocnicza, iteracyjna }

implementation
{ TForm1 }
    { Procedura wywoływana tylko raz podczas tworzenia okna }
procedure TForm1.FormCreate(Sender: TObject);
begin
    for i:= 1 to 10 do
        begin
            { Twórz wszystkie elementy tablicy }
            Grzyb1[i]:= TImage.Create(self);
            Grzyb1[i].Parent:= self;
            Grzyb1[i].Visible:= False;
            Obrazki.GetBitmap(0, Grzyb1[i].Picture.Bitmap);
            Grzyb1[i].Top:= 16;
            Grzyb1[i].Left:= (64*i)-48;
            Grzyb1[i].Tag:= i;
            Grzyb1[i].OnClick:= @Rząd1Click;
            { twórz element }
            { przypisz go do okna }
            { ukryj go }
            { przypisz mu obrazek }
            { określ jego położenie od górnej krawędzi ekranu }
            { określ jego położenie od lewej krawędzi ekranu }
            { przypisz mu numer porządkowy }
            { określ procedurę obsługującą kliknięcie na obiekt }
        end;

    for i:= 1 to 10 do
        begin
            { drugi rząd - wszystko tak jak wyżej }
            Grzyb2[i]:= TImage.Create(self);
            Grzyb2[i].Parent:= self;
            Grzyb2[i].Visible:= False;
            Obrazki.GetBitmap(1, Grzyb2[i].Picture.Bitmap);
            Grzyb2[i].Top:= 80;
            Grzyb2[i].Left:= (64*i)-48;
            Grzyb2[i].Tag:= i;
            Grzyb2[i].OnClick:= @Rząd2Click;
        end;

    for i:= 1 to 10 do
        begin
            { i trzeci rząd jak wyżej }
            Grzyb3[i]:= TImage.Create(self);
            Grzyb3[i].Parent:= self;
            Grzyb3[i].Visible:= False;
            Obrazki.GetBitmap(2, Grzyb3[i].Picture.Bitmap);
        end;
end;

```

```

    Grzyb3[i].Top:= 144;
    Grzyb3[i].Left:= (64*i)-48;
    Grzyb3[i].Tag:= i;
    Grzyb3[i].OnClick:= @Rzad3Click;
end;
Grzyb1[1].Visible:=True;           { Pokazanie pierwszych obiektów z każdego rzędu }
Grzyb2[1].Visible:=True;
Grzyb3[1].Visible:=True;
end;

{ Procedura obsługująca kliknięcie przycisku START }
procedure TForm1.StartClick(Sender: TObject);
begin
    Start.Visible:= False;           { ukryj przycisk }
    Link.Visible:= False;           { ukryj adres autora ikon }
    Opis.Visible:= False;           { ukryj opis gry }
    Info.Visible:= False;           { ukryj Info o wygranej lub przegranej }
    koniec:= False;                 { ustaw zmienną oznaczającą koniec gry na Fałsz }
    tura:= 1;                         { ustaw licznik tur na pierwszą turę }
    Randomize;                         { ustaw pozycję startową generatora liczb losowych }
    ile1:= Random(10)+1;              { wylosuj liczbę grzybów w pierwszym rzędzie }
    for i:= 1 to ile1 do               { pętla przebiega przez elementy od pierwszego do wylosowanego }
        Grzyb1[i].Visible:= True;     { i ustawia, aby elementy były widoczne }
    ile2:= Random(10)+1;              { to samo dla drugiego rzędu }
    for i:= 1 to ile2 do
        Grzyb2[i].Visible:= True;
    ile3:= Random(10)+1;              { i dla trzeciego }
    for i:= 1 to ile3 do
        Grzyb3[i].Visible:= True;
end;

{ Procedura obsługująca kliknięcie w element z pierwszego rzędu }
procedure TForm1.Rzad1Click(Sender: TObject);
begin
    { Sender jest ogólną reprezentacją obiektu, który został kliknięty. Przekształcamy go na konkretny obiekt,
    w tym przypadku typu TImage i pobieramy z niego wartość atrybutu Tag, która reprezentuje numer obiektu.
    Numer ten od razu przekazujemy do procedury obsługującej kliknięcie w określonym rzędzie (Rzad1Action) }
    Rzad1Action((Sender as TImage).Tag);
    { Sprawdź, czy zmienna 'koniec' została ustawiona na Fałsz, jeśli tak, to obsłuż zakończenie gry }
    if koniec = False then PCMove();
end;

procedure TForm1.Rzad2Click(Sender: TObject);           { jak wyżej tylko dla drugiego rzędu }
begin
    Rzad2Action((Sender as TImage).Tag);
    if koniec = False then PCMove();
end;
procedure TForm1.Rzad3Click(Sender: TObject);           { jak wyżej dla trzeciego rzędu }
begin
    Rzad3Action((Sender as TImage).Tag);
    if koniec = False then PCMove();
end;
procedure TForm1.Rzad1Action(grzyb: Integer);           { procedura zmiany stanów rzędu 1 }
begin
    tura:= tura+1;                                       { zwiększ licznik tur }
    for i:= grzyb to ile1 do                               { wykonaj od grzyba klikniętego aż do ostatniego: }

```

```

    Grzyb1[i].Visible:= False;                                { ukryj grzyba }
    ile1:= grzyb-1;                                          { ustaw nową wartość pozostałych grzybów }
    SprawdźKoniec;                                          { sprawdź czy nastąpił koniec gry }
end;
procedure TForm1.Rzad2Action(grzyb: Integer);                { to samo dla rzędu drugiego }
begin
    tura:= tura+1;
    for i:= grzyb to ile2 do
        Grzyb2[i].Visible:= False;
    ile2:= grzyb-1;
    SprawdźKoniec;
end;
procedure TForm1.Rzad3Action(grzyb: Integer);                { i dla rzędu trzeciego }
begin
    tura:= tura+1;
    for i:= grzyb to ile3 do
        Grzyb3[i].Visible:= False;
    ile3:= grzyb-1;
    SprawdźKoniec;
end;
{ Procedura sprawdzająca, czy wszystkie grzyby zostały zabrane }
procedure TForm1.SprawdzKoniec();
begin
    if ile1+ile2+ile3 < 2 then                                { Sprawdź, czy grzybów jest mniej niż dwa }
    begin
        if (tura mod 2) = 1 then                               { Jeśli była tura gracza }
            Info.Caption:= 'Przegrana'                         { ustaw informację o przegranej }
        else if ile1+ile2+ile3 = 1 then                       { Jeśli pozostał ostatni grzyb }
            Info.Caption:= 'Wygrana'                          { ustaw informację o wygranej }
        else                                                  { a jeśli był ruch komputera }
            Info.Caption:= 'Przegrana';                       { ustaw informację o przegranej }
        koniec:= True;                                       { Ustaw zmienną oznaczającą koniec gry }
        Info.Visible:= True;                                  { Pokaż napis z informacją o tym, kto wygrał }
        Start.Visible:= True;                                { Pokaż przycisk START, dla uruchomienia nowej gry }
    end;
end;
procedure TForm1.PCMove();                                   { Procedura realizująca ruch komputera }
begin
    if (ile1=ile2) and (ile1<2) and (ile3>1) then           {gdy w rzędach 1 i 2 jest}
    begin                                                     {0 lub 1 grzyb, to pozostaw 1 grzyba w rzędzie 3.}
        Rzad3Action(2);                                       { kliknij grzyba 2 w rzędzie 3. }
        Exit;                                                 {po wykonanym ruchu wyjdź z funkcji}
    end
    else if (ile1=ile3) and (ile1<2) and (ile2>1) then     {gry w rzędach 1 i 3 jest}
    begin                                                     {0 lub 1 grzyb, to pozostaw 1 grzyba w rzędzie 2.}
        Rzad2Action(2);                                       {kliknij grzyba 2 w rzędzie 2.}
        Exit;
    end
    else if (ile2=ile3) and (ile2<2) and (ile1>1) then     {gdy w rzędach 2 i 3 jest}
    begin                                                     {0 lub 1 grzyb, to pozostaw 1 grzyba w rzędzie 1.}
        Rzad1Action(2);                                       { kliknij grzyba 2 w rzędzie 1.}
        Exit();
    end;
end;
If (ile1+ile2)=1 then                                       {gdy w rzędach 1 i 2 pozostał łącznie jeden grzyb}

```

```

begin
  Rzad3Action(1);                                     {usuń wszystkie z rzędu 3}
  Exit();
end
else if (ile1+ile3)=1 then                             {gdy w rzędach 1 i 3 pozostał łącznie jeden grzyb}
begin
  Rzad2Action(1);                                     {usuń wszystkie z rzędu 2}
  Exit();
end
else if (ile2+ile3)=1 then                             {gdy w rzędach 2 i 3 pozostał łącznie jeden grzyb}
begin
  Rzad1Action(1);                                     {usuń wszystkie z rzędu 1}
  Exit();
end;
end;

```

*{Sprawdzenie parzystości binarnej w środkowej fazie gry. Szukanie układu dającego parzystość grup. Jeśli dla danej wartości 'i' poniższe wyrażenia logiczne z operatorem XOR się zerują, to jest to poszukiwany układ parzystości, dający szansę na wygraną.}*

```

for i:= 1 to 10 do
begin
  if (ile1>=i) and (((ile1-i) xor ile2 xor ile3) = 0) then
begin
  Rzad1Action(ile1 - i + 1)                           {doprowadź układ do parzystości}
  Exit();                                             {po wykonanym ruchu wyjdź z funkcji}
end
else if (ile2>=i) and ((ile1 xor (ile2-i) xor ile3) = 0) then
begin
  Rzad2Action(ile2 - i + 1);                           {doprowadź układ do parzystości}
  Exit();                                             {po wykonanym ruchu wyjdź z funkcji}
end
else if (ile3>=i) and ((ile1 xor ile2 xor (ile3-i)) = 0) then
begin
  Rzad3Action(ile3 - i + 1);                           {doprowadź układ do parzystości}
  Exit();
end;
end;
end;

```

*{Losowy wybór posunięcia, jeśli strategię wykorzystał gracz i nie ma ruchu wygrywającego}*

```

if (ile1>=ile2) and (ile1>=ile3) and ((Random(10)/10)>0.5) then
  Rzad1Action(Random(ile1) + 1)
else if (ile2>=ile1) and (ile2>=ile3) then
  Rzad2Action(Random(ile2) + 1)
else
  Rzad3Action(ile3);
end;

```

```

initialization
  {$I gra.lrs}

end.

```