

## Сапер

{ (с) Культин Н.Б., 2000-2003 Версия для книги Delphi в примерах и задачах. Для вывода справочной информации используется функция WinHelp, а сама справочная информация находится в файле Saper.hlp}

```
unit saper_1;  
interface
```

```
uses  
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
  Menus, StdCtrls, OleCtrls;
```

```
type  
  TForm1 = class(TForm)  
    MainMenu1: TMainMenu;  
    N1: TMenuItem;  
    N2: TMenuItem;  
    N3: TMenuItem;  
    N4: TMenuItem;  
  
    procedure Form1Create(Sender: TObject);  
    procedure Form1Paint(Sender: TObject);  
    procedure Form1MouseDown(Sender: TObject; Button: TMouseButton;  
      Shift: TShiftState; X, Y: Integer);  
    procedure N1Click(Sender: TObject);  
    procedure N4Click(Sender: TObject);  
    procedure N3Click(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;
```

```
var  
  Form1: TForm1;
```

```
implementation
```

```
uses saper_2;
```

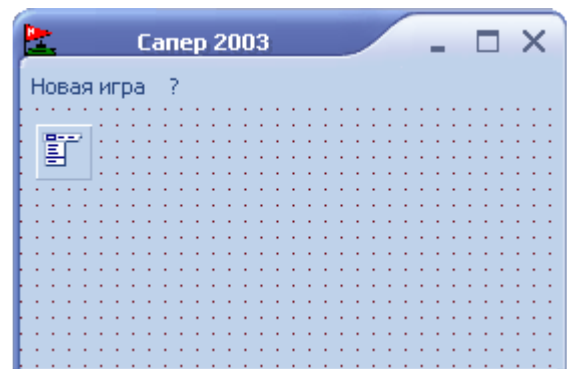
```
{ $R *.DFM }
```

```
*****
```

```
const  
  MR = 10; // кол-во клеток по вертикали  
  MC = 10; // кол-во клеток по горизонтали  
  NM = 10; // кол-во мин  
  W = 40; // ширина клетки поля  
  H = 40; // высота клетки поля
```

```
var  
  Pole: array[0..MR+1, 0..MC+1] of integer; // минное поле  
  // значение элемента массива:  
  // 0..8 - количество мин в соседних клетках  
  // 9 - в клетке мина  
  // 100..109 - клетка открыта  
  // 200..209 - в клетку поставлен флаг
```

```
nMin : integer; // кол-во найденных мин  
nFlag : integer; // кол-во поставленных флагов  
status : integer; // 0 - начало игры; 1- игра; 2 -результат
```



```

Procedure NewGame(); forward; // генерирует новое поле
Procedure ShowPole(Canvas : TCanvas; status : integer); forward; // Показывает поле
Procedure Kletka(Canvas : TCanvas; row, col, status : integer); forward; // выводит содержимое клетки
Procedure Open( row, col : integer); forward; // открывает текущую и все соседние клетки, в которых нет мин
Procedure Mina(Canvas : TCanvas; x, y : integer); forward; // Рисует мину
Procedure Flag( Canvas : TCanvas; x, y : integer); forward; // Рисует флаг

```

```
// выводит на экран содержимое клетки
```

```

Procedure Kletka(Canvas : TCanvas; row, col, status : integer);
var
  x,y : integer; // координаты области вывода
begin
  x := (col-1)* W + 1;
  y := (row-1)* H + 1;
  if status = 0 then
    begin
      Canvas.Brush.Color := clLtGray;
      Canvas.Rectangle(x-1,y-1,x+W,y+H);
      exit;
    end;

  if Pole[row,col] < 100 then
    begin
      Canvas.Brush.Color := clLtGray; // не открытые - серые
      Canvas.Rectangle(x-1,y-1,x+W,y+H);
      // если игра завершена (status = 2), то показать мины
      if (status = 2) and (Pole[row,col] = 9)
        then Mina(Canvas, x, y);
      exit;
    end;

  // открываем клетку
  Canvas.Brush.Color := clWhite; // открытые белые
  Canvas.Rectangle(x-1,y-1,x+W,y+H);
  if ( Pole[row,col] = 100 )
    then exit; // клетка открыта, но она пустая
  if ( Pole[row,col] >= 101) and (Pole[row,col] <= 108 ) then
    begin
      Canvas.Font.Size := 14;
      Canvas.Font.Color := clBlue;
      Canvas.TextOut(x+3,y+2,IntToStr(Pole[row,col] -100 ));
      exit;
    end;
  if ( Pole[row,col] >= 200 ) then
    Flag(Canvas, x, y);
  if (Pole[row,col] = 109 ) then // на этой мине подорвались!
    begin
      Canvas.Brush.Color := clRed;
      Canvas.Rectangle(x-1,y-1,x+W,y+H);
    end;

  if ( (Pole[row,col] mod 10) = 9) and (status = 2) then
    Mina(Canvas, x, y);
end;

```

```
// Показывает поле
```

```
Procedure ShowPole(Canvas : TCanvas; status : integer);
```

```

var
  row,col : integer;
begin
  for row := 1 to MR do
    for col := 1 to MC do
      Kletka(Canvas, row, col, status);
    end;
  end;

```

// рекурсивная функция открывает текущую и все соседние клетки, в которых нет мин

```

Procedure Open( row, col : integer);
begin
  if Pole[row,col] = 0 then
    begin
      Pole[row,col] := 100;
      Kletka(Form1.Canvas, row,col, 1);
      Open(row,col-1);
      Open(row-1,col);
      Open(row,col+1);
      Open(row+1,col);
      //примыкающие диагонально
      Open(row-1,col-1);
      Open(row-1,col+1);
      Open(row+1,col-1);
      Open(row+1,col+1);
    end
  else
    if (Pole[row,col] < 100) and ( Pole[row,col] <> -3 ) then
      begin
        Pole[row,col] := Pole[row,col] + 100;
        Kletka(Form1.Canvas, row, col, 1);
      end;
    end;
end;

```

// новая игра - генерирует новое поле

```

procedure NewGame();
var
  row,col : integer; // координаты клетки
  n : integer; // количество поставленных мин
  k : integer; // кол-во мин в соседних клетках
begin
  // Очистим элементы массива, соответствующие клеткам игрового поля.
  for row :=1 to MR do
    for col :=1 to MC do
      Pole[row,col] := 0;
    end;
  end;

  // расставим мины
  Randomize(); // инициализация ГСЧ
  n := 0; // кол-во мин
  repeat
    row := Random(MR) + 1;
    col := Random(MC) + 1;
    if ( Pole[row,col] <> 9) then
      begin
        Pole[row,col] := 9;
        n := n+1;
      end;
  until (n = NM);

```

```

// для каждой клетки вычислим кол-во мин в соседних клетках
for row := 1 to MR do
  for col := 1 to MC do
    if ( Pole[row,col] <> 9 ) then
      begin
        k :=0 ;
        if Pole[row-1,col-1] = 9 then k := k + 1;
        if Pole[row-1,col] = 9 then k := k + 1;
        if Pole[row-1,col+1] = 9 then k := k + 1;
        if Pole[row,col-1] = 9 then k := k + 1;
        if Pole[row,col+1] = 9 then k := k + 1;
        if Pole[row+1,col-1] = 9 then k := k + 1;
        if Pole[row+1,col] = 9 then k := k + 1;
        if Pole[row+1,col+1] = 9 then k := k + 1;
        Pole[row,col] := k;
      end;
status := 0; // начало игры
nMin := 0; // нет обнаруженных мин
nFlag := 0; // нет флагов
end;

// Рисует мину
Procedure Mina(Canvas : TCanvas; x, y : integer);
begin
  with Canvas do
    begin
      Brush.Color := clGreen;
      Pen.Color := clBlack;
      Rectangle(x+16,y+26,x+24,y+30);
      Rectangle(x+8,y+30,x+16,y+34);
      Rectangle(x+24,y+30,x+32,y+34);
      Pie(x+6,y+28,x+34,y+44,x+34,y+36,x+6,y+36);
      MoveTo(x+12,y+32); LineTo(x+26,y+32);
      MoveTo(x+8,y+36); LineTo(x+32,y+36);
      MoveTo(x+20,y+22); LineTo(x+20,y+26);
      MoveTo(x+8, y+30); LineTo(x+6,y+28);
      MoveTo(x+32,y+30); LineTo(x+34,y+28);
    end;
end;

// Рисует флаг
Procedure Flag( Canvas : TCanvas; x, y : integer);
var
  p : array [0..3] of TPoint; // координаты флажка и нижней точки древка
  m : array [0..4] of TPoint; // буква М
begin
  // зададим координаты точек флажка
  p[0].x:=x+4; p[0].y:=y+4;
  p[1].x:=x+30; p[1].y:=y+12;
  p[2].x:=x+4; p[2].y:=y+20;
  p[3].x:=x+4; p[3].y:=y+36; // нижняя точка древка
  m[0].x:=x+8; m[0].y:=y+14;
  m[1].x:=x+8; m[1].y:=y+8;
  m[2].x:=x+10; m[2].y:=y+10;
  m[3].x:=x+12; m[3].y:=y+8;
  m[4].x:=x+12; m[4].y:=y+14;
  with Canvas do

```

```

begin
  // установим цвет кисти и карандаша
  Brush.Color := clRed;
  Pen.Color := clRed;
  Polygon(p); // флажок
  // древко
  Pen.Color := clBlack;
  MoveTo(p[0].x, p[0].y);
  LineTo(p[3].x, p[3].y);
  // буква М
  Pen.Color := clWhite;
  Polyline(m);
  Pen.Color := clBlack;
end;
end;

```

// выбор из меню ? команды О программе

### **procedure TForm1.N4Click(Sender: TObject);**

```

begin
  AboutForm.Top := Trunc(Form1.Top + Form1.Height/2 - AboutForm.Height/2);
  AboutForm.Left := Trunc(Form1.Left + Form1.Width/2 - AboutForm.Width/2);
  AboutForm.ShowModal;
end;

```

### **procedure TForm1.Form1Create(Sender: TObject);**

```

var
  row,col : integer;
begin
  // В неотображаемые эл-ты массива, которые соответствуют клеткам по границе игрового поля
  запишем число -3. Это значение используется функцией Open для завершения рекурсивного процесса
  открытия соседних пустых клеток.
  for row :=0 to MR+1 do
    for col :=0 to MC+1 do
      Pole[row,col] := -3;
  NewGame(); // "разбросать" мины
  Form1.ClientHeight := H*MR + 1;
  Form1.ClientWidth := W*MC + 1;
end;

```

// нажатие кнопки мыши на игровом поле

### **procedure TForm1.Form1MouseDown(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);**

```

var
  row, col : integer;
begin
  if status = 2 // игра завершена
    then exit;
  if status = 0 then // первый щелчок
    status := 1;
  // преобразуем координаты мыши в индексы клетки поля
  row := Trunc(y/H) + 1;
  col := Trunc(x/W) + 1;
  if Button = mbLeft then
    begin
      if Pole[row,col] = 9 then
        begin // открыта клетка, в которой есть мина

```

```

    Pole[row,col] := Pole[row,col] + 100;
    status := 2; // игра закончена
    ShowPole(Form1.Canvas, status);
end
else if Pole[row,col] < 9 then
    Open(row,col);
end
else
    if Button = mbRight then
        if Pole[row,col] > 200 then
            begin
                // уберем флаг и закроем клетку
                nFlag := nFlag - 1;
                Pole[row,col] := Pole[row,col] - 200; // уберем флаг
                x := (col-1)* W + 1;
                y := (row-1)* H + 1;
                Canvas.Brush.Color := clLtGray;
                Canvas.Rectangle(x-1,y-1,x+W,y+H);
            end
        else
            begin // поставить в клетку флаг
                nFlag := nFlag + 1;
                if Pole[row,col] = 9
                    then nMin := nMin + 1;
                Pole[row,col] := Pole[row,col]+ 200; // поставили флаг
                if (nMin = NM) and (nFlag = NM) then
                    begin
                        status := 2; // игра закончена
                        ShowPole(Form1.Canvas, status);
                    end
                else Kletka(Form1.Canvas, row, col, status);
            end;
        end;
end;
end;

```

// Выбор меню Новая игра

**procedure TForm1.N1Click(Sender: TObject);**

```

begin
    NewGame();
    ShowPole(Form1.Canvas,status);
end;

```

// выбор из меню ? команды Справка

**procedure TForm1.N3Click(Sender: TObject);**

```

begin
    // вывести справочную информацию
    Winhelp(Form1.Handle,'saper.hlp',HELP_CONTEXT,1);
end;

```

// обработка события OnPaint

**procedure TForm1.Form1Paint(Sender: TObject);**

```

begin
    // отобразить игровое поле
    ShowPole(Form1.Canvas, status);
end;
end.

```