

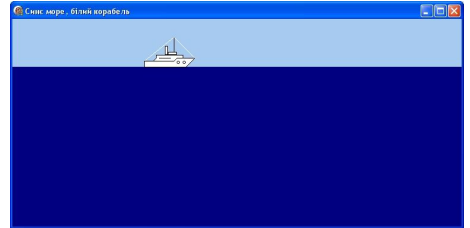
Практична робота №18

Тема: Програмування створення найпростішої мультиплікації.

Мета: Навчитися використовувати графічні процедури та функції.

Хід роботи

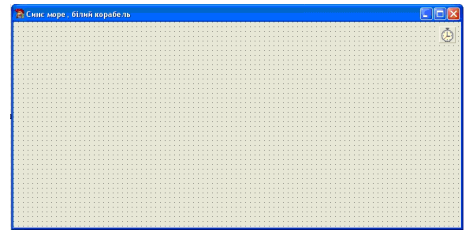
1. Створити у своїй робочій папці каталог PRACT18_1.
2. Запустити середовище Delphi та створити новий додаток.
3. Виконати File – Save All, вказавши пунктом призначення створений Вами каталог PRACT18_1, під іменами, які пропонує середовище Delphi.



Умова задачі. Написати програму, по поверхні вікна якої переміщується кораблик.

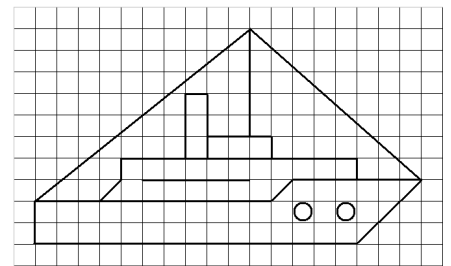
Деякі зауваження щодо проектування форми.

На формі розміщено лише компонент таймер (Timer1).



Деякі зауваження щодо підготовки до створення програми.

Перед створення програмного коду доцільно на аркуші в клітинку створити зображення кораблика, враховуючи значення «зафіксованих» точок – вузлів координатної сітки. Вважаючи, що одна клітинка має розміри 5 пікселів на 5 пікселів, можна виявити залежність між координатами кінців відрізків, за допомогою яких, в основному, будується корпус кораблика.



Деякі зауваження щодо створення програмного коду.

У заголовку форми доцільно відображати, окрім назви практичної роботи, також прізвища та імена виконавців.

У програмі використовуються три процедури:

```
type
  TForm1 = class(TForm)
    Timer1: TTimer;
  procedure FormPaint(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
```

Перед початком роботи програми доцільно обумовити координати базової точки корабля:

implementation

```
{ $R *.dfm }
var x,y: integer; //координати корабля (базової точки)
```

Рекомендується створити процедуру користувача, яка буде малювати кораблик (чи витирати його). У ній будуть використовуватися координати базової точки, значення True (видимий кораблик) чи False (невидимий кораблик). У цій же процедурі доцільно визначити значення кроку зміни – двох чисел, які визначають розмір кораблика. У нашому випадку обидва числа рівні 5. Щоб збільшити (чи зменшити) розмір кораблика, потрібно лише змінити ці значення.

```

procedure parohod (x,y: integer; mode: boolean);
const dx=5; dy=5;
var
  p1: array [1..7] of TPoint; //корпус кораблика
  p2: array [1..8] of TPoint; //надбудова над корпусом кораблика
  pc, bc: TColor;           //колір олівця та кисті
begin

```

Корпус та надбудову над корпусом кораблика будемо будувати за допомогою метода polygon (масив, який складається із декількох точок; координати останньої точки рівні координатам першої точки – лінія замкнена). Для витирання зображення кораблика на попередньому місці використовуються кольори фону (неба та моря). Для малювання кораблика на новому місці – кольори, відмінні від кольору фону.

```

  if not mode then
  begin
    //виперти
    form1.Canvas.Brush.Color:=clNavy;
    form1.Canvas.Pen.Color:=clNavy;
    form1.Canvas.Rectangle(x,y+1,x+17*dx,y-10*dy);
    form1.Canvas.Brush.Color:=clNavy;
    //небо
    if y-10*dy<80 then
    begin
      //кінець мачти на фоні неба
      form1.Canvas.Pen.color:=clSkyBlue;
      form1.Canvas.Brush.color:=clSkyBlue;
      form1.Canvas.Rectangle(x,y-10*dy,x+17*dx,80);
    end;
    exit;
  end;

//малюємо
with form1.canvas do
begin
  pc:=pen.color;
  bc:=brush.color;
  pen.Color:=clBlack;
  brush.color:=clWhite;
  //корпус
  p1[1].X:=x;           p1[1].Y:=y;
  p1[2].X:=x;           p1[2].Y:=y-2*dy;
  p1[3].X:=x+10*dx;     p1[3].Y:=y-2*dy;
  p1[4].X:=x+11*dx;     p1[4].Y:=y-3*dy;
  p1[5].X:=x+17*dx;     p1[5].Y:=y-3*dy;
  p1[6].X:=x+14*dx;     p1[6].Y:=y;
  p1[7].X:=x;           p1[7].Y:=y;
  polygon (p1);

  //надбудова
  p2[1].X:=x+3*dx;       p2[1].Y:=y-2*dy;
  p2[2].X:=x+4*dx;       p2[2].Y:=y-3*dy;
  p2[3].X:=x+4*dx;       p2[3].Y:=y-4*dy;
  p2[4].X:=x+13*dx;      p2[4].Y:=y-4*dy;
  p2[5].X:=x+13*dx;      p2[5].Y:=y-3*dy;
  p2[6].X:=x+11*dx;      p2[6].Y:=y-3*dy;
  p2[7].X:=x+10*dx;      p2[7].Y:=y-2*dy;
  p2[8].X:=x+3*dx;       p2[8].Y:=y-2*dy;
  polygon (p2);

  moveto (x+5*dx,y-3*dy);
  lineto (x+9*dx,y-3*dy);

```

```
//капітанський мостик
rectangle (x+8*dx,y-4*dy,x+11*dx,y-5*dy);
//труба
rectangle (x+7*dx,y-4*dy,x+8*dx,y-7*dy);
//ілюмінатори
ellipse (x+11*dx,y-2*dy,x+12*dx,y-1*dy);
ellipse (x+13*dx,y-2*dy,x+14*dx,y-1*dy);
//мачта
moveto (x+10*dx,y-5*dy);
lineto (x+10*dx,y-10*dy);
//оснастка
pen.Color:=clWhite;
moveto (x+17*dx,y-3*dy);
lineto (x+10*dx,y-10*dy);
lineto (x,y-2*dy);
pen.color:=pc;
end;
end;

//початкове положення кораблика, колір моря, час затримки
procedure TForm1.FormCreate(Sender: TObject);
begin
  x:=0;
  y:=80;
  form1.color:=clNavy;
  timer1.Interval:=50;
end;

procedure TForm1.FormPaint(Sender: TObject);
begin
  canvas.Brush.Color:=clSkyBlue; //небо
  canvas.Pen.Color:=clSkyBlue;
  canvas.Rectangle(0,0,clientwidth,80);
end;

//обробка сигналу таймера
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  parohod (x,y, false); //вигерти малюнок
  if x<form1.clientwidth
  then x:=x+2
  else
  begin //новий рейс
    x:=0;
    y:=random(50)+100;
  end;
  parohod (x,y,true); //намалювати кораблик в іншому місці
end;
```

4. Зберегти отриманий додаток на диску та запустимо його на виконання.
5. Завершити роботу із середовищем Delphi.
6. Із операційної оболонки запустити файл із розширенням exe, який утворився у створеній Вами папці PRACT18_1.
7. Завдання для самостійного виконання.

- 7.1. Створити у своїй папці каталог PRACT18_2, до якого скопіювати файли щойно створеного Вами проекту. Внести до нього зміни, які б дозволяли розв'язати наступну задачу:
 - 7.1.1. Написати програму, по поверхні вікна якої переміщується кораблик, але у зворотному напрямку (справа наліво).
8. Створити звіт із результатами виконання практичної роботи.
9. Навести порядок на робочому місці.

