

## PROG2 - Programmation impérative

### TP 07 – Les enregistrements

Julien Sopena

Février 2008

#### Exercice 1 : Coordonnées

##### Question 1

Déclarer (ainsi que les constantes nécessaires) :

- un type **t\_point** permettant de stoker les coordonnées  $x$  et  $y$
- un type **t\_rectangle** permettant de stoker les coordonnées des sommets Sud-Ouest et Nord-Est d'un rectangle (horizontal).
- un type **t\_tabPoints** permettant de stoker un ensemble de points dans un tableau.

```
const
  NB_MAX_POINTS = 50;
type
  t_indicePoints = 1..NB_MAX_POINTS;
  t_point = record
    x : real;
    y : real;
  end;
  t_rectangle = record
    SO : t_point;
    NE : t_point;
  end;
  t_tabPoints = array [t_indicePoints] of t_point;
```

##### Question 2

Écrire un sous-programme **saisirPoint** qui permette de saisir au clavier les coordonnées  $x$  et  $y$  d'un point.

```
procedure saisirPoint (var p : t_point);
begin
```

```

write ('Entrez la coordonnée x : ');
readln (p.x);
write ('Entrez la coordonnée y : ');
readln (p.y);
end ;

```

### Question 3

Écrire un sous-programme **afficherPoint** qui permette d'afficher les coordonnées  $x$  et  $y$  d'un point sous la forme :

(x,y)

```

procedure afficherPoint (p : t_point) ;
begin
  write('(',p.x :0 :2,',',p.y :0 :2,')');
end ;

```

### Question 4

Écrire un sous-programme **distancePoints** qui calcule la distance entre deux points.

```

function distancePoints (p1, p2 : t_point) : real ;
begin
  distancePoints := sqrt(sqr(p1.x-p2.x)+sqr(p1.y-p2.y));
end ;

```

### Question 5

Écrire un sous-programme **milieuPoints** qui calcule le milieu de deux points.

```

procedure milieuPoints (p1, p2 : t_point; var pm : t_point) ;
begin
  pm.x := (p1.x + p2.x) / 2 ;
  pm.y := (p1.y + p2.y) / 2 ;
end ;

```

### Question 6

Écrire un sous-programme **afficherRectangle** qui permette d'afficher les coordonnées  $x$  et  $y$  de tous les sommets d'un rectangle sous la forme :

SO = (x,y)  
 SE = (x,y)  
 NE = (x,y)  
 NO = (x,y)

```

procedure afficherRectangle (r : t_rectangle) ;
begin
  with r do
  begin

```

```

writeln('SO = (',SO.x :0 :2,', ',SO.y :0 :2,')');
writeln('SE = (',NE.x :0 :2,', ',SO.y :0 :2,')');
writeln('NE = (',NE.x :0 :2,', ',NE.y :0 :2,')');
writeln('NO = (',SO.x :0 :2,', ',NE.y :0 :2,')');
end ;
end ;

```

### Question 7

Écrire un sous-programme **estDansRectangle** qui teste si un point est contenu dans un rectangle.

```

function estDansRectangle (p : t_point; r : t_rectangle) : boolean ;
begin
  estDansRectangle := (r.SO.x <= p.x) and (p.x<=r.NE.x) and
    (r.SO.y <= p.y) and (p.y<=r.NE.y) ;
end;

```

### Question 8

Écrire un sous-programme **saisirTabPoints** qui permette de remplir un tableau avec les coordonnées d'un ensemble de points. Ce sous-programme demandera préalablement à l'utilisateur de renseigner le nombre de points à saisir. Attention le sous-programme devra s'assurer que l'utilisateur entre au moins un point.

```

procedure saisirTabPoints (var t : t_tabPoints; var n : t_indicePoints);
var
  i : t_indicePoints ;
begin
  repeat
    write ('Enrer un nombre de point (entre 1 et ',NB_MAX_POINTS,') : ');
    readln (n);
  until (1<n) and (n<NB_MAX_POINTS);
  for i := 1 to n do
    begin
      writeln ('Point ',i);
      saisirPoint(t[i]);
    end ;
  end;

```

### Question 9

Écrire un sous-programme **cadre** qui retourne les coordonnées d'un rectangle contenant tous les points d'un tableau passé en argument.

```

procedure cadre (t : t_tabPoints; n : t_indicePoints; var r : t_rectangle);
var
  i : t_indicePoints ;
begin
  r.SO := t[1];
  r.NE := t[1];

```

```

for i := 2 to n do
  begin
    if (r.SO.x > t[i].x) then
      r.SO.x := t[i].x;
    if (r.SO.y > t[i].y) then
      r.SO.y := t[i].y;
    if (r.NE.x < t[i].x) then
      r.NE.x := t[i].x;
    if (r.NE.y < t[i].y) then
      r.NE.y := t[i].y;
    end ;
  end ;

```

### Question 10

A l'aide de l'ensemble des sous-programmes, écrire un programme qui, à partir d'une liste de points saisis au clavier, affiche :

- les coordonnées d'un cadre contenant tous les points saisis.
- la longueur de la diagonale de ce cadre.
- les coordonnées du centre de ce cadre.

```

var
  tabPoints : t_tabPoints;
  n : t_indicePoints ;
  rect : t_rectangle;
  centre : t_point ;
begin
  saisirTabPoints(tabPoints, n);
  cadre(tabPoints, n, rect);
  writeln ('Les coordonnÃ©es du cadre contenant tous les points sont :');
  afficherRectangle(rect);
  write ('La longueur de la diagonale du cadre est : ');
  writeln (distancePoints(rect.SO,rect.NE));
  milieuPoints(rect.SO,rect.NE,centre);
  write ('Le centre du cadre est : ');
  afficherPoint(centre);
  readln ;
end.

```