

Programmation et Structure de Données

Structure de données de Base
Les Structures

Déclaration des Structures

Pb : Variable qui représente un point définit par un nom et deux coordonnées x et y)?

Sol : utiliser les structures.

La **structure** nous permet de désigner sous un seul nom un ensemble de valeurs pouvant être de types différents.

Déclaration :

Point : struct

Debut

nom : chaine de caractères

X, Y : entier;

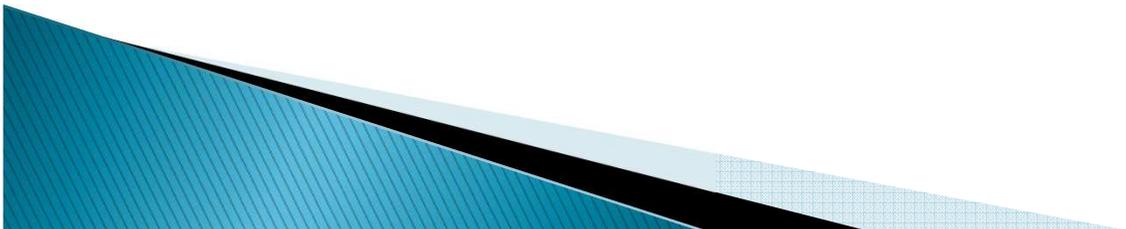
Fin;

Var

P1,P2 : Point;



Définition d'un nouveau type : Point qu'on va utiliser après pour déclarer P1 et P2



Utilisation des Structures

Pour accéder aux éléments d'une structure:

```
P1.nom ← "Point01"; P1.X ← 4 ; P1.Y ← 8;
```

Exemple : on veut représenter un produit défini par désignation, prix et stock puis déclarer un produit P et demander à l'utilisateur de lire ses attributs.

```
Algo Structure_Produit
```

```
Prod : Struct
```

```
  Début
```

```
    Des : chaîne de caractères;
```

```
    prix : réel;
```

```
    stock : entier;
```

```
  Fin;
```

```
Var P1 : Prod;
```

```
  Début
```

```
    écrire (" Donner la désignation  
            du produit:");
```

```
  Lire (P1.Des);
```

```
  écrire (" Donner le prix du  
          produit :");
```

```
  Lire (P1.prix);
```

```
  écrire (" Donner le Stock du  
          produit :");
```

```
  Lire (P1.stock);
```

```
  écrire ("voici votre produit :",  
          P1.Des, " son prix=",  
          P1.prix, " avec un stock ="  
          ,P1.Stock);
```

```
FIN.
```

Utilisation des Structures

Traduire l'exemple précédent en C++:

```
main()
{
    struct Prod
    {
        char des[20];
        float prix;
        int stock;
    };
    Prod P1;
    cout << "Donner la designation du produit:";
    cin >> P1.des;
    cout << "Donner le prix du produit:";
    cin >> P1.prix;
    cout << "Donner le stock du produit:";
    cin >> P1.stock;
    cout << "\nVoici votre produit :" << P1.des << " son prix :
        " << P1.prix << " DA avec un stock : " << P1.stock << endl;
}
```

Utilisation des Structures

Initialisation :

Prod P1 = {"Clavier", 300, 23};

Tableaux de Structures :

➤ **Algorithmique** : Liste : tableau[100] de Prod;

➤ **C++** : Prod Liste[100];

Où Liste est un tableau de produits.

Accéder aux éléments :

Liste[i].des, Liste[i].prix, ...

Structures comportant d'autres structures :

Date : struct

debut

jour : entier ;

mois : entier;

annee : entier;

Fin ;

Personne : struct

Debut

nom : tableau[30] de caract;

pren : tableau[30] de caract;

DN : Date;

Fin ;

Ex :

P : personne;

P.nom ← "Nabil";

P.Pren ← "Ahmed";

P.DN.jour ← 26;

P.DN.mois ← 7;

P.DN.jour ← 1991;

Structure en argument d'un sous programme

Ex1 (passage par valeur):

Écrire un s/prg qui renvoi la distance entre deux points

Point : struct

 Debut

 x,y : réel;

 Fin;

Fonction dist (P1: Point, P2: Point) : réel

 Début

 retourner Sqrt(pow(P1.x-P2.x , 2)+pow(P1.y-P2.y , 2));

 Fin;

Ex2 (passage par référence):

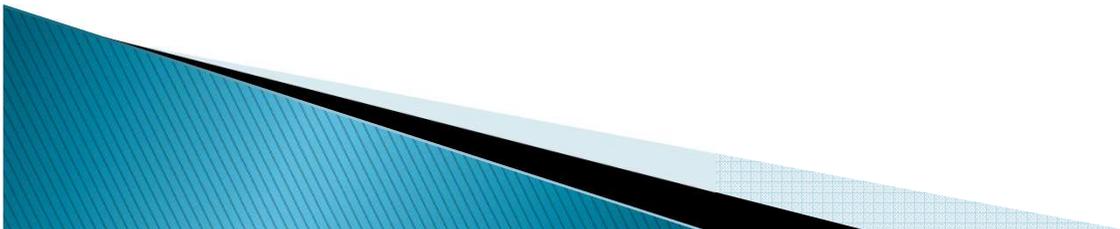
Écrire un s/prg qui permet de déplacer un point horizontalement avec une distance d (changer la coordonnée x)

Procédure déplacer (var ou & P:point, d : réel)

 Début

 P.x = P.x + d;

 Fin;



Structure en argument d'un sous programme

En C++ :

```
// Déclaration :  
struct Point  
{  
    float x,y;  
};  
  
// Fonction Distance :  
float dist(Point P1, Point P2)  
{  
    return sqrt(pow(P1.x-P2.x , 2)+pow(P1.y-P2.y , 2));  
}  
  
//Procédure Déplacer Horizontalement :  
void deplace(Point &P, float d)  
{  
    P.x+=d;  
}
```

Structure comme valeur de retour d'une fonction

Supposant :

```
struct Point
```

```
{
```

```
    float x,y;
```

```
};
```

```
Point P1={2,5}, P2;
```

L'écriture : $P2 = P1$; est juste et équivalente à

```
P2.x = P1.x;    P2.y = P1.y;
```

Écrire un s/prg qui permet de créer un point

```
Point creer (float a, float b)
```

```
{
```

```
    Point P;
```

```
    P.x = a;    P.y = b;
```

```
    return P;
```

```
}
```

Appelle dans le programme principal : $\text{Point A} = \text{creer}(23, 3,5);$

Exemple de synthèse sur les structures

On veut gérer un tableau des produits (max 100). Un produit est défini par un code, une désignation, un prix et un stock. Le code se varie entre 1 et le nombre global des produits, la désignation est une chaîne de caractères.

La gestion se fait grâce au menu suivant qui se réaffiche après chaque opération jusqu'à taper 5 (Quitter) :

1. Ajouter un produit.
2. Supprimer un produit grâce à son code.
3. Afficher tous les produits sous la forme du tableau suivant par exemple :

| Code | Des | prix | stock |
|-------------|--------------|-------------|--------------|
| 1 | Lait | 100 | 23 |
| 2 | Sucre | 90 | 25 |

4. Vente d'un produit dont on connaît le code avec une quantité donnée par l'utilisateur.
5. Quitter

Écrire un programme C++ qui permet de gérer un tableau de produits, utiliser les sous programmes pour réaliser ce programme. ([Le travail à faire](#))