

4

*ALGORITHMIQUE
ET
PROGRAMMATION*

4^{ème} année de l'enseignement secondaire
Science de l'informatique

Professeur
Mohamed TRABELSI

Table des matières

Chapitre 1 : Les enregistrements et les fichiers

Chapitre 2 : La récursivité

Chapitre 3 : Les algorithmes de tri

Chapitre 4 : Les algorithmes récurrents

Chapitre 5 : Les algorithmes d'arithmétique

Chapitre 6 : Les algorithmes d'approximation

Chapitre 7 : Les algorithmes avancés

Chapitre 1

Les enregistrements et les fichiers

Durée : Heures

Type : Théorique / Pratique

A- Les enregistrements**I. Introduction**

Activité 1 : Une société désire saisir et traiter des données se rapportant aux produits qu'elle fabrique. On vous communique la fiche produit utilisée d'habitude par cette société :

Fiche produit	
•	Référence :
•	Nom :
•	Prix unitaire (DT) :
•	Quantité (unité) :
•	Exportation (vrai / faux) :

Quelles structures de données pourrions-nous utiliser pour saisir les données d'un produit ?

1. On peut penser à 5 variables pour chaque produit comme suit :

T.D.O

Objet	Type	Rôle
R	Chaîne	Référence produit
N	Chaîne	Nom produit
PU	Réel	Prix unitaire
Q	Entier	Quantité en stock
E	Booléen	A l'exportation ou non

Problème : Ces variables ne permettent de saisir qu'un seul produit.

2. On peut créer un tableau pour chaque attribut du produit :

Exemple :

T.D.O

Objet	Type	Rôle
Tab ref	Tableau de 100 chaînes	Tableau des références produit
Tab nom	Tableau de 100 chaînes	Tableau des noms produit
...

Problèmes : Les informations concernant un produit sont éparpillées sur plusieurs tableaux. On ne peut pas donc parler de l'entité produit.

En programmation, on dispose d'une structure de données appelée enregistrement (Record) qui pourrait représenter l'entité produit.

II. Définition et déclaration

II.1. Définition

II.2. Déclaration

III. Utilisation des enregistrements

III.1. Affectation

III.2. Lecture

III.3. Ecriture

Activité 2 : (Suite de l'activité 1)

On désire créer un programme qui permet de saisir un produit et de calculer la valeur du stock pour ce produit. Au cas on est en rupture de stock en ce produit le programme affiche le message "produit indisponible actuellement !".

- $V_stock = \text{Prix_unitaire} \times \text{Quantité}$

a. Analyse

Résultat : affichage

Traitement : **Si** `p1.quantité` $\neq 0$ **alors** calculer et afficher `v_stock`

Sinon afficher le message

Fin Si

`P1.référence` = donnée

`P1.nom` = donnée

`P1.pu` = donnée

`P1.quantité` = donnée

`P1.exportation` = donnée

a. Analyse principale :

Résultat : Afficher la liste des références produits et leur PU à l'exportation.

Traitement :

La procédure Affiche () fera le teste pour faire l'affichage des produits à l'exportation.

Remplir **tp1**.

Saisir n

Algorithme

- 0) Début produit_export
- 1) Proc Saisie (n)
- 2) Proc Remplir (**tp1**, n)
- 3) Proc Affiche (**tp1**, n)
- 4) Fin produit_export

T.D.N.T

Type
Produit = Enregistrement
Référence : Chaîne [10]
Nom : Chaîne [50]
Pu : Réel
Quantité : Entier
Exportation : octet
tab_prod = Tableau de 50 produit

T.D.O. Globaux

Objet	Type	Rôle
tp1	tab_prod	Tableau de produit
n	entier	Nombre de produit à saisir
Saisie	Procédure	Saisie de n
Remplir	Procédure	Remplir liste
Affiche	Procédure	Teste et affichage

b. Analyse de la procédure saisie :

Résultat : n saisi

Traitements :

Algorithme

Procédure **saisie** (var n : octet)

c. Analyse de la procédure remplir :

Résultat : **tp1** saisie

Traitements :

Parcours total sur **tp1** grâce à une boucle pour
 Avec **tp1 [i]** faire
 Référence = donnée
 Nom = donnée
 Pu = donnée
 Quantité = donnée
 Exportation = donnée
 Fin avec

Algorithme

0) Procédure **remplir** (var tp1 : tab_prod, n : octet)

1) Pour i de 1 à n faire

 Avec **tp1 [i]** faire

 Lire (Référence)

 Lire (Nom)

 Lire (Pu)

 Lire (Quantité)

 Lire (Exportation)

 Fin Avec

Fin pour

Fin **remplir**

T.D.O. Locaux

Objet	Type	Rôle
i	octet	Compteur

d. Analyse de la procédure affiche :

Résultat : Afficher les produits disponibles.

Traitements :

Grâce à une boucle pour on fera un parcours total sur le tableau **tp1**.

Sur chaque enregistrement de **tp1** on fera le teste suivant :

Si **tp1.exportation = 1** alors afficher **tp1.référence** et **tp1.pu**

Algorithme

0) Procédure **affiche** (tp1 : tab_prod, n : octet)

1) Pour i de 1 à n faire

 Avec **tp1 [i]** faire

 Si **tp1.exportation = 1** alors écrire (**tp1.référence**, **tp1.pu**)

 Fin si

 Fin Avec

Fin pour

2) Fin **affiche**

T.D.O. Locaux

Objet	Type	Rôle
i	octet	Compteur

Pascal (voir fichier : prod2.pas)