

Étapes supplémentaires :

Etape 1 :

On va créer un programme qui sauvegarde et affiche les clients du fichier "compte.dat"

Analyse du programme principale	Algorithme
Résulta : Proc affiche(f) F=Proc Création(f,n) N="Donnez le nombre de clients"	0) Début compte 1) Ecrire ("donner le nombre de clients"), lire(n) 2) Proc création (f,n) 3) Proc affiche(f) 4) Fermer(f) 5) Fin compte

Tableau de déclaration des nouveaux types

Type																	
ident=enregistrement code_b:chaîne de 5 caractère code_r: chaîne de 20 caractère Fin ident client=record nom: chaîne de 30 caractère prenom: chaîne de 30 caractère id:ident solde:réel Fin client fiche=fichier de client	Tableau de déclaration des objets																
	<table border="1"> <thead> <tr> <th>Objets</th> <th>Type/nature</th> <th>Rôle</th> </tr> </thead> <tbody> <tr> <td>f</td> <td>fiche</td> <td>fichier des clients</td> </tr> <tr> <td>n</td> <td>Entier</td> <td>Taille du fichier</td> </tr> <tr> <td>affichage</td> <td>procédure</td> <td>-----</td> </tr> <tr> <td>création</td> <td>procédure</td> <td>-----</td> </tr> </tbody> </table>	Objets	Type/nature	Rôle	f	fiche	fichier des clients	n	Entier	Taille du fichier	affichage	procédure	-----	création	procédure	-----	
Objets	Type/nature	Rôle															
f	fiche	fichier des clients															
n	Entier	Taille du fichier															
affichage	procédure	-----															
création	procédure	-----															

Analyse de la procédure création	Algorithme
DEF Proc création (var f :fiche ; n :entier) Résulta : f f[]=pour i de 1 à n faire Avec cl faire écrire('Donner le nom:'), lire(nom); écrire ('Donner le prénom:'); lire(prenom); écrire ('Donner le code banque:'); lire (id.code_b); écrire ('Donner le code région:'); lire(id.code_r); écrire ('Donner la solde:'); lire(solde); Fin avec Ecrire(f,cl) Fin pour Recréer(f) Associer (f, 'c:\4si\compte.dat')	0) DEF Proc création (var f:fiche ; n :entier) 1) Associer (f, 'c:\4si\compte.dat') 2) Recréer(f) 3) pour i de 1 à n faire Avec cl faire écrire('Donner le nom:'), lire(nom); écrire ('Donner le prénom:'); lire(prenom); écrire ('Donner le code banque:'); lire (id.code_b); écrire ('Donner le code région:'); lire(id.code_r); écrire ('Donner la solde:'); lire(solde); Fin avec Ecrire(f,cl) Fin pour 4) fin création

Tableau de déclaration des objets locaux :

Objets	Type/nature	Rôle
i	Entier	Compteur
cl	client	-----

Analyse de la procédure affiche	Algorithme
DEF Proc Affiche (var f :fiche) Résulta : tantque non(fin_fichier (f))faire Lire (f,cl) Avec cl faire écrire('Nom :',nom) écrire ('Prénom :',prenom) écrire ('Code banque :',id.code_b) écrire ('Région :',id.code_r) écrire ('Solde :',solde) Fin avec Fin tant que Ouvrire(f)	0) DEF Proc Affiche (var f :fiche, n :entier) 1) Ouvrire(f) 2) Tantque non(fin_fichier (f))faire Lire (f,cl) Avec cl faire écrire('Nom :',nom) écrire ('Prénom :',prenom) écrire ('Code banque :',id.code_b) écrire ('Région :',id.code_r) écrire ('Solde :',solde) Fin avec Fin tant que 3) Fin Affiche

Tableau de déclaration des objets locaux :

Objets	Type/nature	Rôle
cl	client	Enregistrement point

Traduction en Pascal :

<pre> program compte; uses wincrt; type ident=record code_b:string[5]; code_r:string[20]; end; client=record nom:string[40]; prenom:string[40]; id:ident; solde:integer; end; fiche=file of client; Var cl:client; f:fiche; n:integer; {-----Procédure CREATION-----} procedure creation(var f:fiche ; n:integer); var i:integer; cl:client; begin assign (f,'c:\4si\compte.dat'); rewrite(f); for i:=1 to n do begin with cl do begin write('Donner le nom:'); readln(nom); write('Donner le prénom:'); readln(prenom); write('Donner le code banque:'); readln(id.code_b); write('Donner le code région:'); readln(id.code_r); write('Donner la solde:'); readln(solde); end; write(f,cl); writeln('-----'); end; end; </pre>	<pre> end; {-----Module AFFICHE-----} procedure affiche (var f:fiche); var cl:client; begin reset(f); while not(Eof(f))do begin read(f,cl); with cl do begin writeln('Nom :',nom); writeln('Prénom :',prenom); writeln('Code banque :',id.code_b); writeln('Région :',id.code_r); writeln('Solde :',solde); writeln('====='); end; end; end; {-----Prog PRINCIPAL-----} Begin write('donner le nombre de client: '); readln(n); creation(f,n); affiche(f); end. </pre>
--	---

Etape 2 : Exécuter le programmer et enregistrer le fichier “compte.dat”.

Travail demandé:

Analyse du problème :

Analyse du programme principale	Algorithme
Résulta : Proc affiche(fr) fr=Proc Création(fr,r) r='Donnez la région :'	0) Début région 1) Ecrire ('' Donnez la région :'), lire(r) 2) Proc Création(fr,r) 3) Proc affiche(fr) 4) Fermer (fr) 5) Fin région

Tableau de déclaration des nouveaux types

Type																	
ident=enregistrement code_b:chaîne de 5 caractère code_r: chaîne de 20 caractère Fin ident client=record nom: chaîne de 30 caractère prenom: chaîne de 30 caractère id:ident solde:réel Fin client fiche=fichier de client	<i>Tableau de déclaration des objets</i>																
	<table border="1"> <thead> <tr> <th>Objets</th> <th>Type/nature</th> <th>Rôle</th> </tr> </thead> <tbody> <tr> <td>fr</td> <td>fiche</td> <td>fichier des clients</td> </tr> <tr> <td>r</td> <td>chaîne</td> <td>Taille du fichier</td> </tr> <tr> <td>affichage</td> <td>procédure</td> <td>-----</td> </tr> <tr> <td>création</td> <td>procédure</td> <td>-----</td> </tr> </tbody> </table>	Objets	Type/nature	Rôle	fr	fiche	fichier des clients	r	chaîne	Taille du fichier	affichage	procédure	-----	création	procédure	-----	
Objets	Type/nature	Rôle															
fr	fiche	fichier des clients															
r	chaîne	Taille du fichier															
affichage	procédure	-----															
création	procédure	-----															

Analyse de la procédure création	Algorithme
DEF Proc création (var fr :fiche ; r :chaîne) Résulta : fr fr[]=tant que (non (fin_fichier(fc))) faire lire(fc,cl) si(cl.id.code_r = r) alors ecrire(fr,cl) Fin tant que Ouvrire (fc) Recréer (fr) Associer (fr, ''c:\4si\région.dat'') Associer (fc, ''c:\4si\compte.dat'')	0) DEF Proc création (var fr :fiche ; r :chaîne) 1) Associer (fr, ''c:\4si\région.dat'') 2) Associer (fc, ''c:\4si\compte.dat'') 3) Recréer(fr) 4) Ouvrire(fc) 5) tant que (non (fin_fichier(fc))) faire lire(fc,cl) si(cl.id.code_r = r) alors ecrire(fr,cl) Fin tant que 6) fermer (fc) 7) fin création

Tableau de déclaration des objets locaux :

Objets	Type/nature	Rôle
fc	fichier	Compteur
cl	client	-----

Analyse de la procédure affiche	Algorithme
DEF Proc Affiche (var fr :fiche) Résulta : tantque non(fin_fichier (fr))faire Lire (fr,cl) Si(cl.solde<=500) alors Avec cl faire écrire('Nom :',nom) écrire ('Prénom :',prenom) écrire ('Code banque :',id.code_b) écrire ('Région :',id.code_r) écrire ('Solde :',solde) Fin avec Fin si Fin tant que Ouvrire(fr)	0) DEF Proc Affiche (var fr :fiche, n :entier) 1) Ouvrire(fr) 2) Tantque non(fin_fichier (fr))faire Lire (fr,cl) Si(cl.solde<=500) alors Avec cl faire écrire('Nom :',nom) écrire ('Prénom :',prenom) écrire ('Code banque :',id.code_b) écrire ('Région :',id.code_r) écrire ('Solde :',solde) Fin avec Fin si Fin tant que 3) Fin Affiche

Tableau de déclaration des objets locaux :

Traduction en Pascal :

Objets	Type/nature	Rôle
cl	client	Enregistrement point

<pre> program region; uses wincrt; type ident=record code_b:string[5]; code_r:string[20]; end; client=record nom:string[40]; prenom:string[40]; id:ident; solde:integer; end; fiche=file of client; Var cl:client; r:string[20]; fc,fr:fiche; {-----Procédure CREATION-----} procedure creation(var fr:fiche; r:string); var i:integer; cl:client; fc:fiche; begin assign (fc,'c:\4si\compte.dat'); assign (fr,'c:\4si\region.dat'); rewrite(fr); reset(fc); while not(Eof(fc))do begin read(fc,cl); if(cl.id.code_r=r)then write(fr,cl); end; close(fc); end; </pre>	<pre> {----- Module AFFICHE -----} procedure région (var fr:fiche); var cl:client; begin reset(fr); while not(Eof(fr))do begin read(fr,cl); if(cl.solde<=500)then with cl do begin writeln('====='); writeln('Nom :',nom); writeln('Prénom :',prenom); writeln('Code banque :',id.code_b); writeln('Région :',id.code_r); writeln('Solde :',solde); end; end; end; end; {-----Prog PRINCIPAL-----} Begin write ('région= '); readln(r); creation(fr,r); affiche(fr); close(fr) ; end. </pre>
--	--