

Fiche I : Types abstraits

Type	Exemple de déclaration	Primitives
<i>pile</i>	var P : pile d'entiers	Action créerPile (\underline{S} P : TPile) Fonction pileVide (P : TPile) : Booléen Fonction valeurSommet (P : TPile) : TInfo Action empiler (\underline{ES} P : TPile ; \underline{E} Elem : TInfo) Action dépiler (\underline{ES} P : TPile)
<i>files</i>	var F : file de caractères	Action créerFile (\underline{S} F : TFile) Fonction fileVide (F : TFile) : Booléen Fonction valeurPremier (F : TFile) : TInfo Action enfiler (\underline{ES} F : TFile ; \underline{E} Elem : TInfo) Action défiler (\underline{ES} F : TFile)
<i>listes séquentielles</i>	var L1 : liste de TEtd	Action créerListe (\underline{S} L : TListe) Fonction adressePremier (L : TListe) : TAdresse Fonction adresseSuivant (L : TListe ; Adr : TAdresse) : TAdresse Fonction valeurElement (L : TListe ; Adr : TAdresse) : TInfo Action modifieValeur (\underline{ES} L : TListe ; \underline{E} Adr : TAdresse ; \underline{E} Elem : TInfo) Action insérerEnTête (\underline{ES} L : TListe ; \underline{E} : Elem : TInfo) Action insérerAprès (\underline{ES} L : TListe ; \underline{E} : Elem : TInfo ; \underline{E} Adr : TAdresse) Action supprimerEnTête (\underline{ES} L : TListe) Action supprimerAprès (\underline{ES} L : TListe ; \underline{E} Adr : TAdresse)
<i>tables internes</i>	var TableEtd = table de TEtd et de clé Nom <i>Nom doit être un champ de TEtd...</i>	Action créerTable (\underline{S} T : TTable) Action valeurElement (\underline{E} T : TTable ; \underline{E} Clé : TClé ; \underline{S} Elem : TInfo ; \underline{S} OK : booléen) Action valeurElementSup (\underline{E} T : TTable ; \underline{E} Clé : TClé ; \underline{S} Elem : TInfo ; \underline{S} OK : booléen) Action ajouteValeur (\underline{ES} T : TTable ; \underline{E} Elem : TInfo ; \underline{S} OK : booléen) Action supprimeValeur (\underline{ES} T : TTable ; \underline{E} Clé : TClé ; \underline{S} OK : booléen) Action modifieValeur (\underline{ES} T : TTable ; \underline{E} Elem : TInfo ; \underline{S} OK : booléen) Action premièreValeur (\underline{E} : T : TTable ; \underline{S} Elem : TInfo ; \underline{S} OK : booléen) Action valeurSuivante (\underline{E} : T : TTable ; \underline{S} Elem : TInfo ; \underline{S} OK : booléen)
<i>tables externes (fichiers séquentiels indexés)</i>	Var FS1 : fichier séquentiel indexé de TEtd et de clé Nom <i>Nom doit être un champ de TEtd...</i>	Action ouvrirFichier (\underline{ES} Fic : TFicSI ; <mode_ouverture>) Action fermerFichier (\underline{ES} Fic : TFicSI) Action lireFichierDirect (\underline{E} Fic : TFicSI ; \underline{E} Clé : TClé ; \underline{S} Elem : TInfo) Action lireFichierSup (\underline{E} Fic : TFicSI ; \underline{E} Clé : TClé ; \underline{S} Elem : TInfo) Action lireFichierSuivant (\underline{E} Fic : TFicSI ; \underline{S} Elem : TInfo) Action écrireFichierDirect (\underline{ES} Fic : TFicSI ; \underline{E} Elem : TInfo) Action réécrireFichierDirect (\underline{ES} Fic : TFicSI ; \underline{E} Elem : TInfo) Action supprimerFichierDirect (\underline{ES} Fic : TFicSI ; \underline{E} Clé : TClé) Fonction étatFichier (Fic : TFicSI) : TEtat <i>Modes d'ouverture</i> Idem fichiers séquentiels + modification (mises à jour) <i>Valeurs retournées par la fonction EtatFichier</i> <ul style="list-style-type: none"> FdF : fin de fichier, après LireFichierSup ou LireFichierSuivant. Echec, Succès : échec ou succès de l'opération, après toute opération.

Exercices

1. On suppose qu'une expression arithmétique est codée sous forme d'une chaîne de caractères.
 - a. Ecrire une action permettant de récupérer un à un les éléments d'une telle expression (entiers, parenthèses ou opérateurs).
 - b. Ecrire une action permettant d'évaluer une expression postfixée.
 - c. Ecrire une action permettant de convertir une expression infixe en une expression postfixée.
2. Ecrire une action permettant de « retourner » une file (le premier élément devient dernier, etc.).
3. Ecrire une action permettant de calculer la moyenne d'une liste d'étudiants.
4. Ecrire une action permettant de supprimer dans une liste d'étudiants tous les étudiants dont la moyenne est inférieure à 10.
5. Ecrire une action permettant de « retourner » une liste d'étudiants.
6. Ecrire une action permettant de concaténer deux listes d'étudiants.
7. Ecrire un ensemble d'actions ou de fonctions permettant de gérer une liste triée d'étudiants.
8. Ecrire une action permettant de fusionner deux listes d'étudiants triées.
9. On considère les déclarations suivantes :

```

Type          TPersonne = entité (  NumPers : entier
                                Nom, Adresse : chaîne )
TVéhicule = entité (              Immatriculation, Marque, Type : chaîne )
TPossède = entité (              Possession : entité (  NumPers : entier
                                Immatriculation : chaîne )
                                DateAchat : TDate )
TTablePers = table de TPersonne et de clé NumPers
TTableVeh = table de TVéhicule et de clé Immatriculation
TTablePos = table de TPossède et de clé Possession
  
```

On suppose que les tables TabP (de type TTablePers), TabV (de type TTableVeh) et TabPos (de type TTablePos) existent.

- a. Ecrire une action qui affiche les immatriculations des véhicules d'une marque donnée.
 - b. Ecrire une action qui affiche les immatriculations des véhicules d'une marque donnée, ainsi que le nom de leur propriétaire.
 - c. Ecrire une action qui affiche le nom des personnes possédant au moins un véhicule ainsi que la marque et le type du ou des véhicule(s) qu'elles possèdent.
10. On considère les déclarations suivantes :


```

Const          CNbNotes = 10
Type          TEtudiant = entité (  Nom, Prénom : chaîne
                                Notes : tableau [CNbNotes] de réels )
TEficEtd = fichier séquentiel indexé de TEtudiant et de clé Nom
      
```

Ecrire un ensemble de fonctions et/ou d'actions permettant de gérer efficacement un fichier de type TEficEtd.