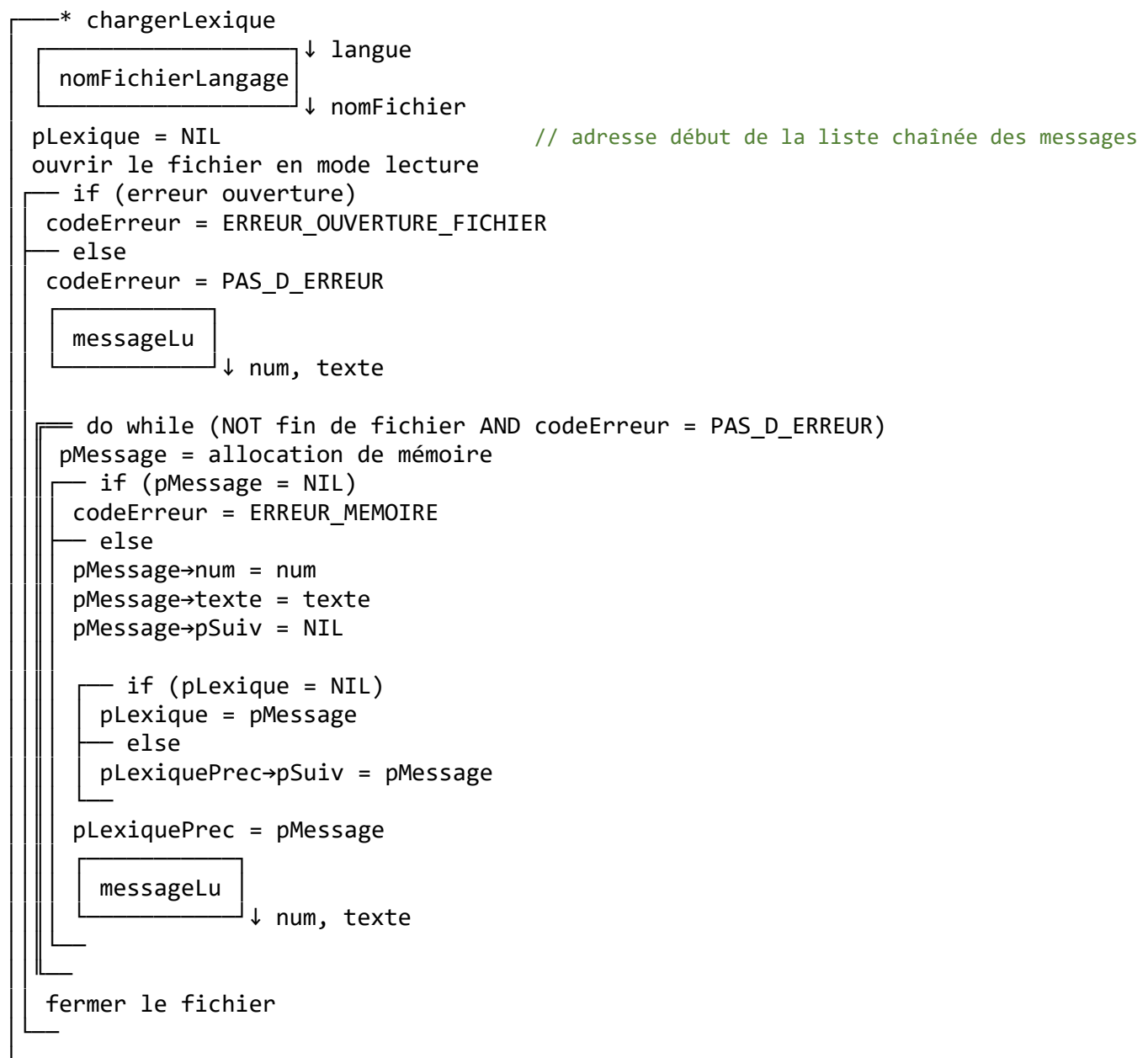
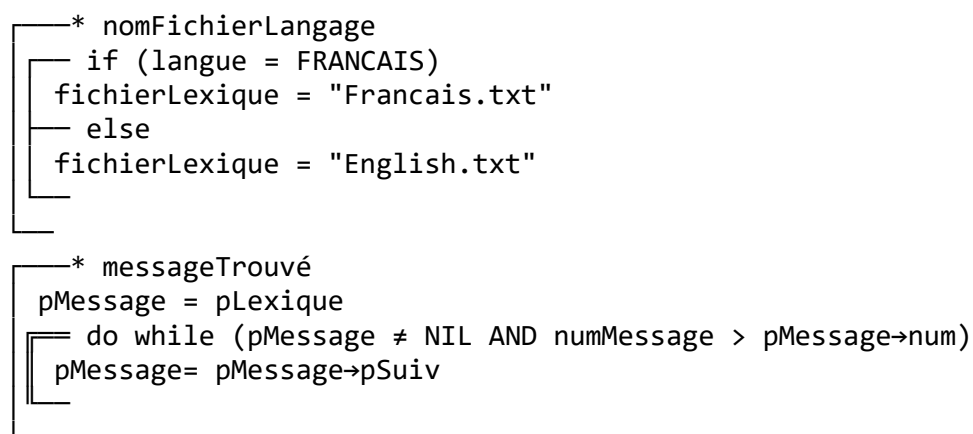




/\*\*\*\*\*\* LEXIQUE \*\*\*\*\*/



Les modules suivants concernent le lexique et seront appelés par d'autres modules que *chargerLexique*.



```

* afficherMessage
  ↓ pLexique, numMessage
  messageTrouvé
  ↓ pMessage
  sortir pMessage→texte

```

// module qui permet dans un menu d'affichage de choix de déterminer le nombre de choix possibles

// NBMAXCHOIXMENU constante déterminée pour l'ensemble du programme

```

* nbChoixMenu
  maxChoix = 0
  do while (pMenu ≠ NIL and pMenu→num < debMenu + NBMAXCHOIXMENU)
    maxChoix++
    pMenu= pMenu→pSuiv

```

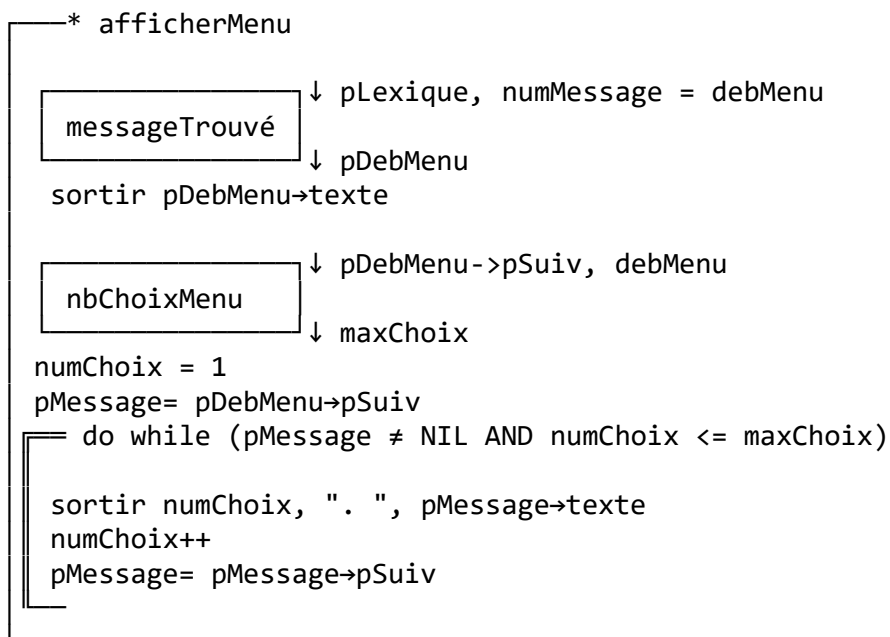
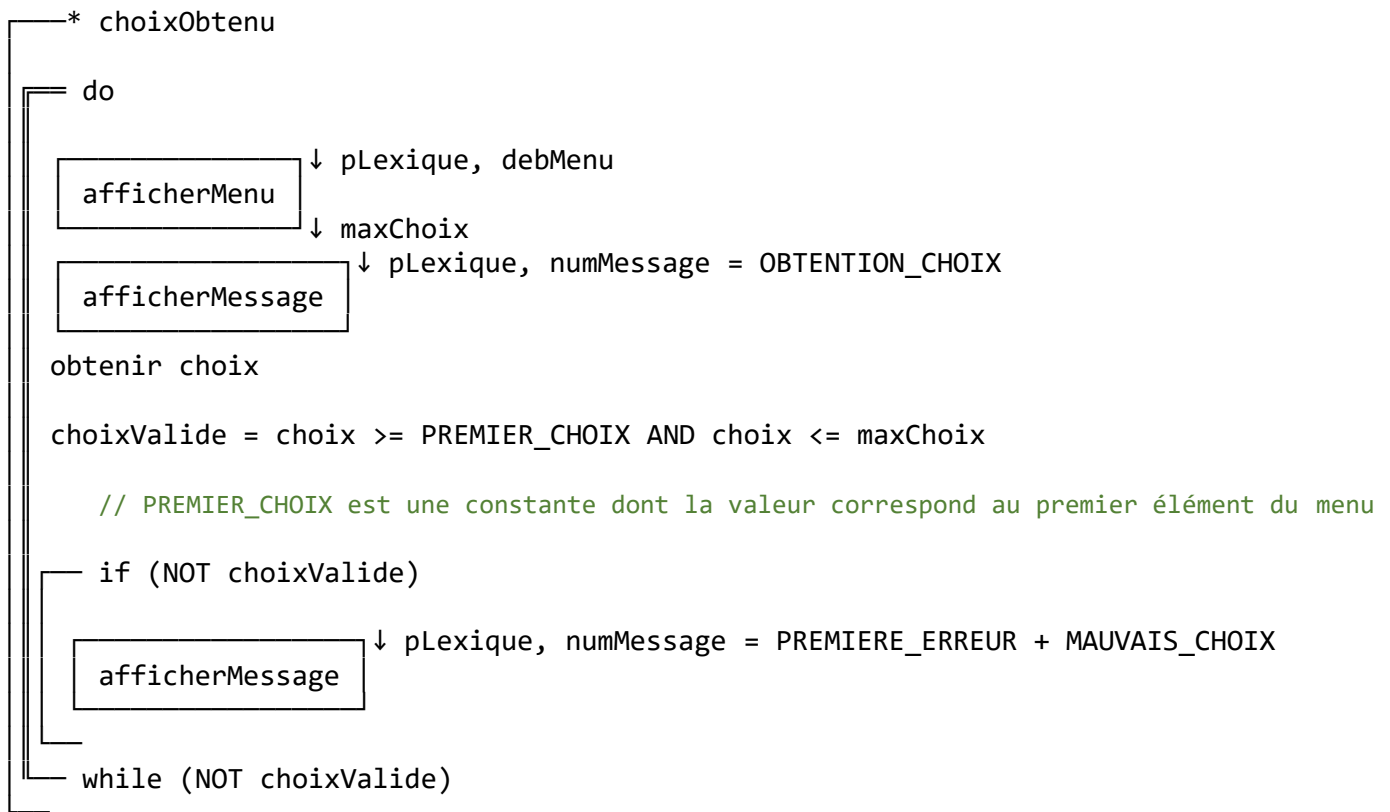
```

/***** INTERFACE *****/

```



<sup>1</sup> Numéro de référence de ce texte dans le fichier de langues correspondant



### Exemple :

```

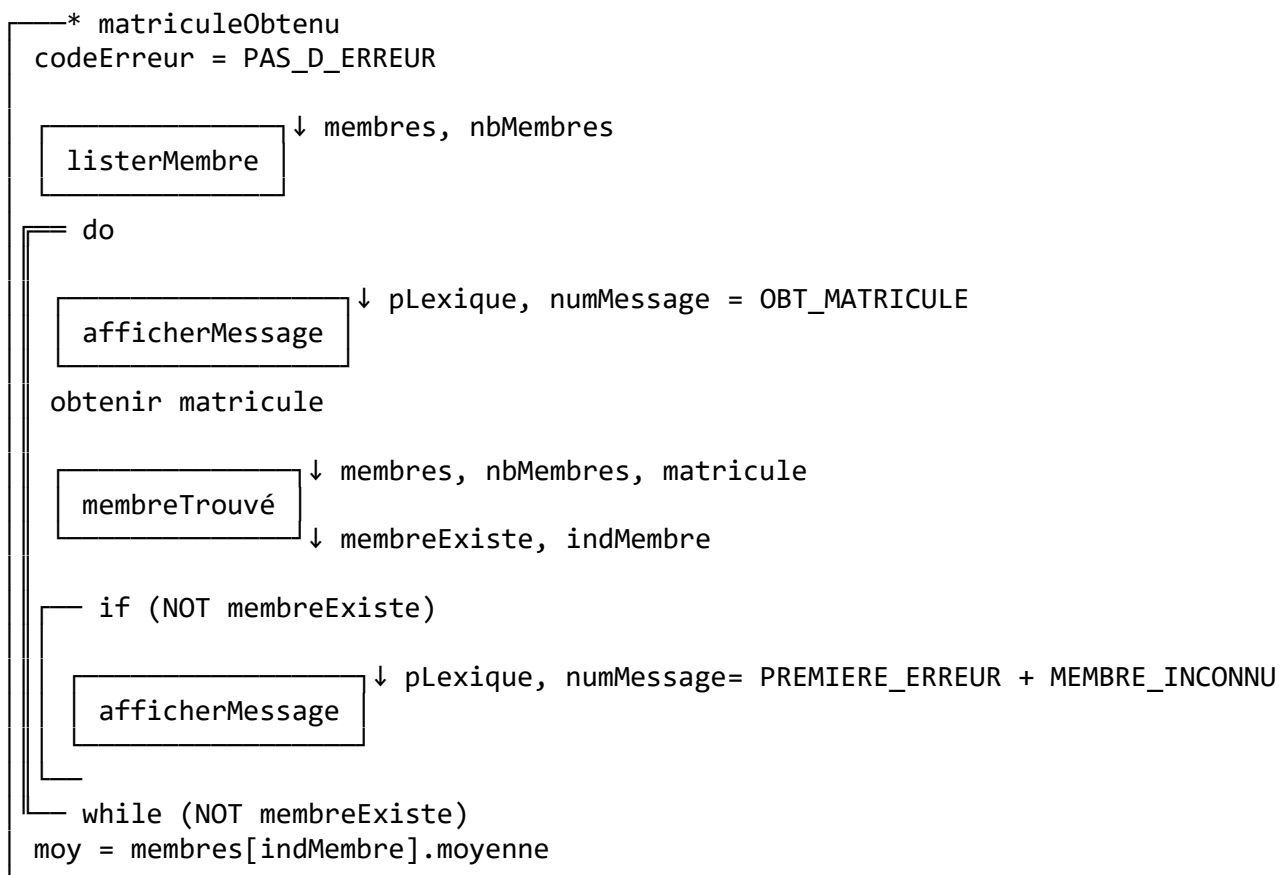
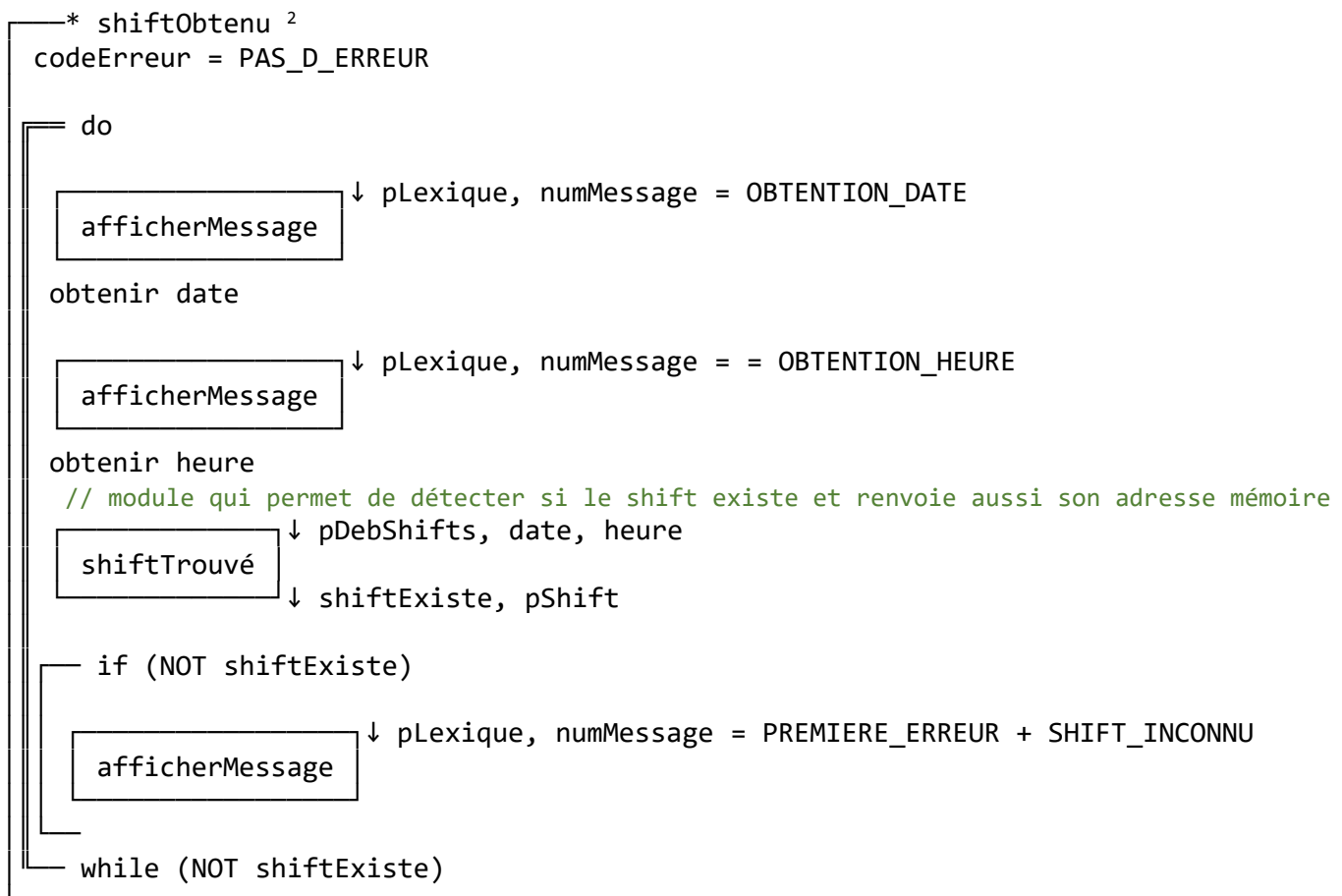
1001MENU PRINCIPAL
1002AJOUTER UNE INSCRIPTION
1003SUPPRIMER UNE INSCRIPTION
1004IMPRIMER
1005QUITTER

```

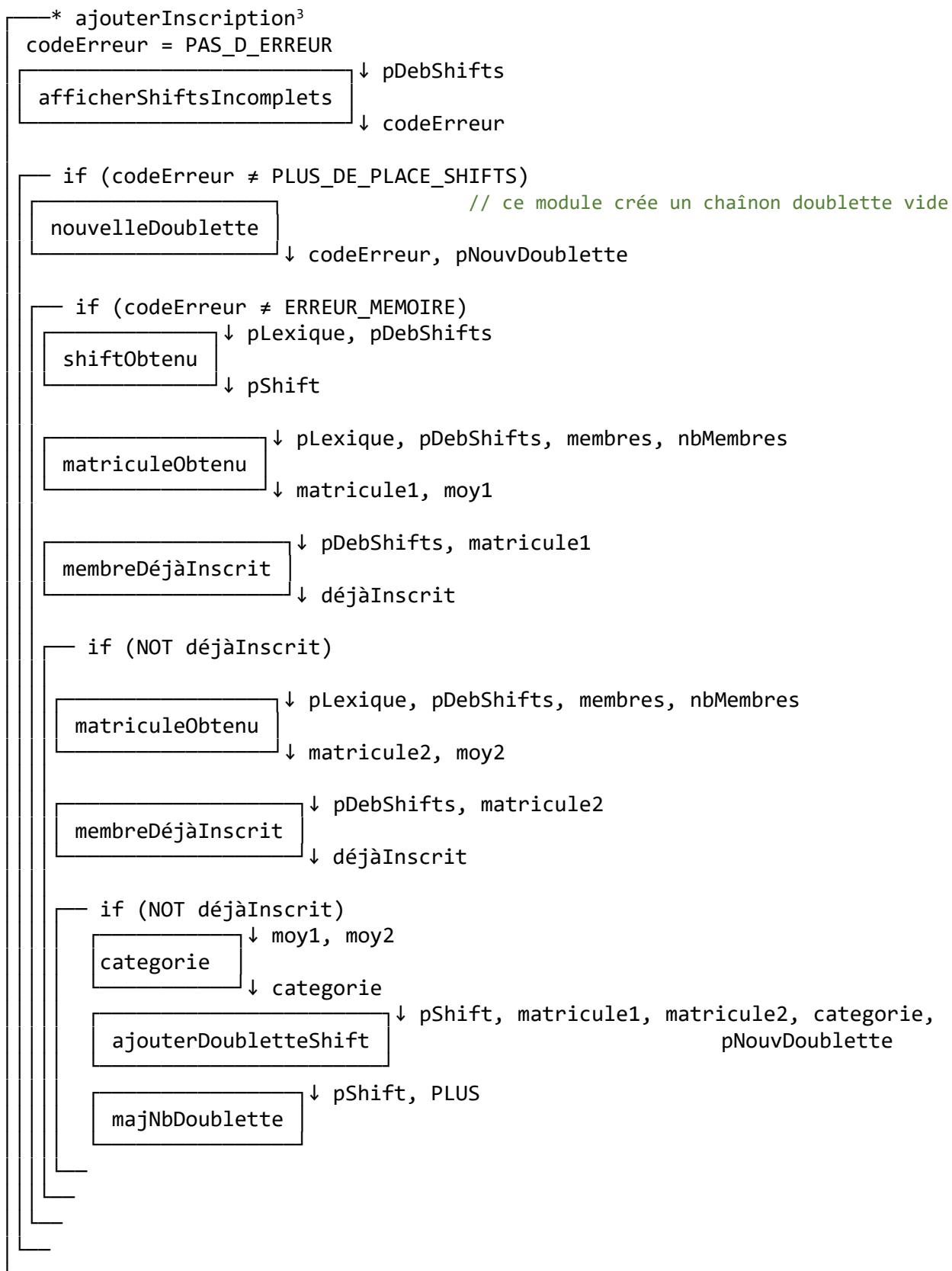
Donnera :

MENU PRINCIPAL

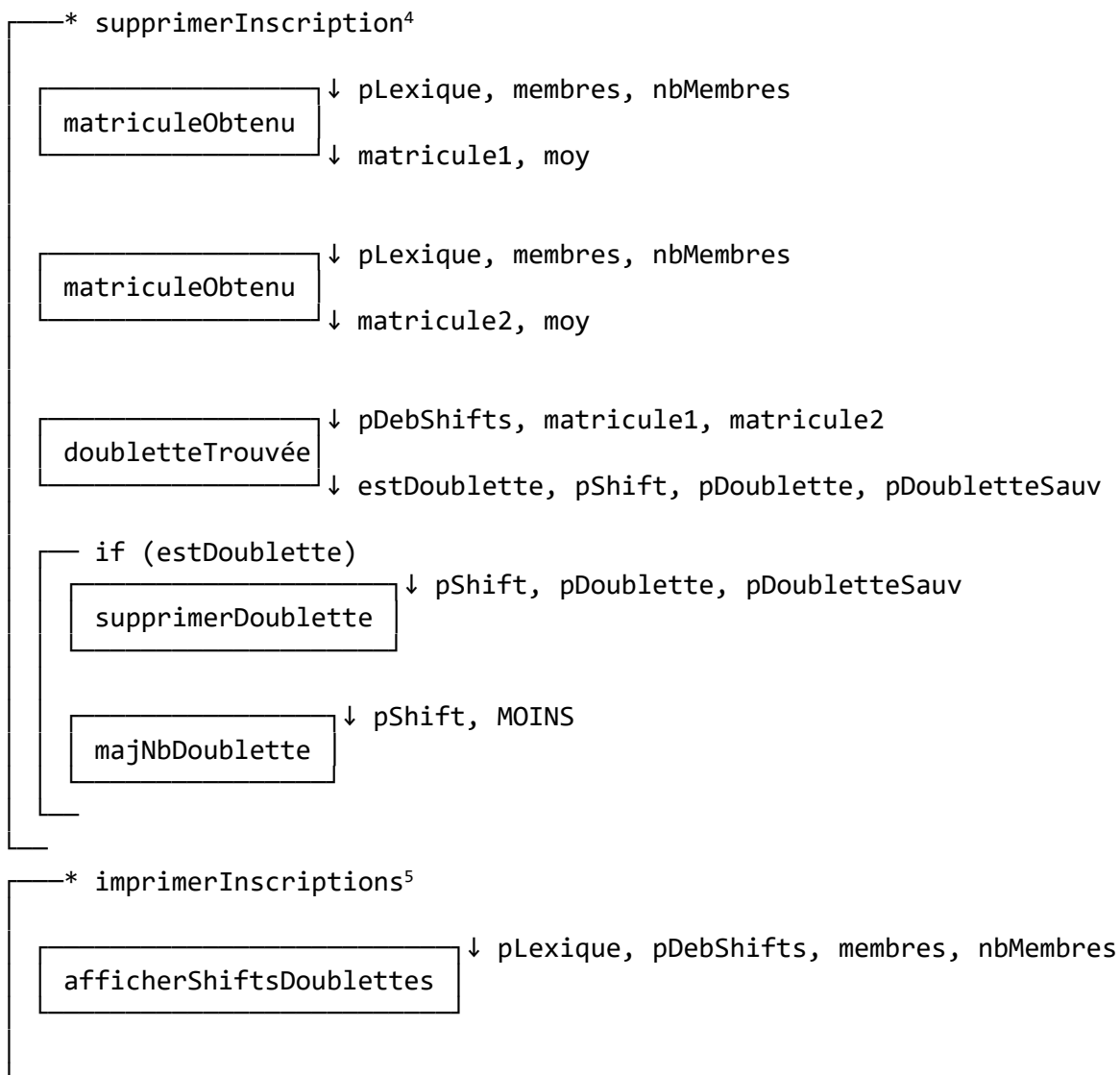
1. Ajouter une inscription
2. Supprimer une inscription...



<sup>2</sup> Les deux modules de cette page attendent que l'utilisateur entre des données cohérentes. A vous d'adapter votre programme si vous souhaitez quitter les modules à un moment donné.



<sup>3</sup> Ce DA est à ajuster si vous ajustez les modules shiftObtenu et matriculeObtenu !



<sup>4</sup> Même remarque : à adapter si vous adaptez le module matriculeObtenu ; de plus, nous n'avons prévu un message si la doublette n'existe pas !

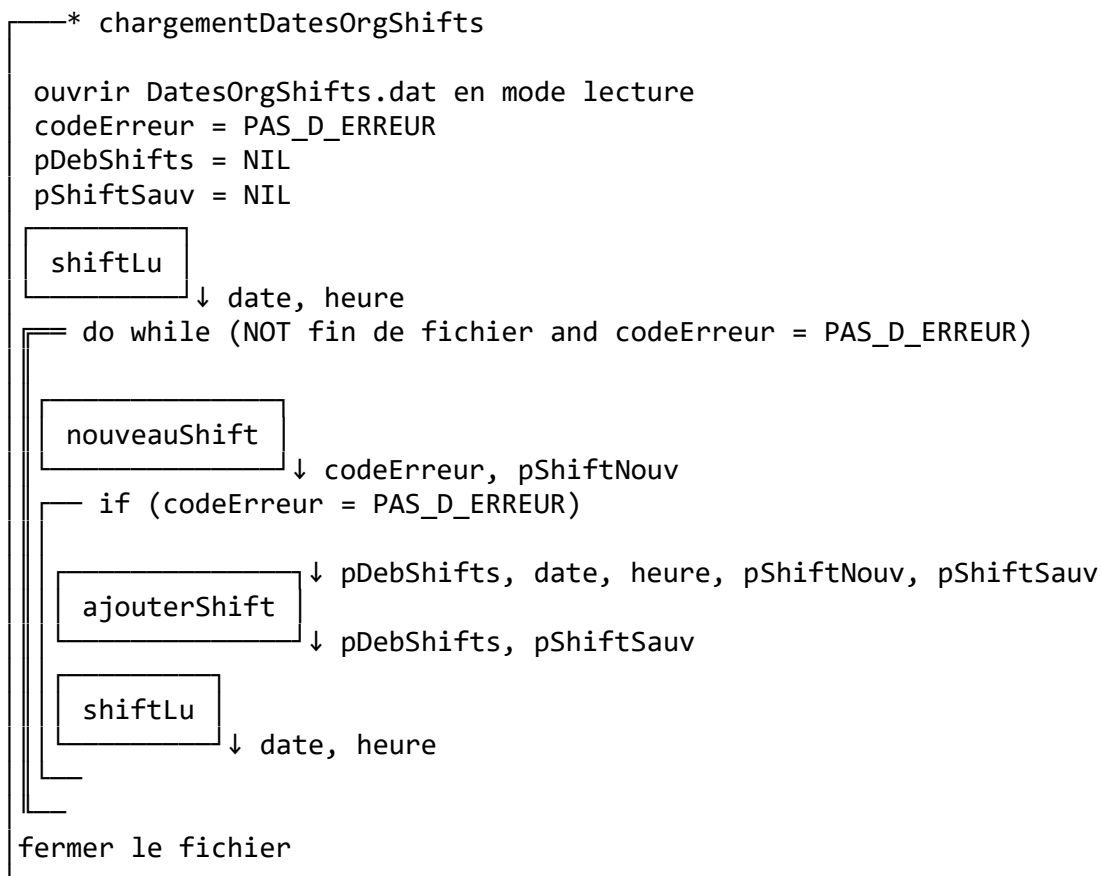
<sup>5</sup> Adaptez vos messages quand les listes sont vides

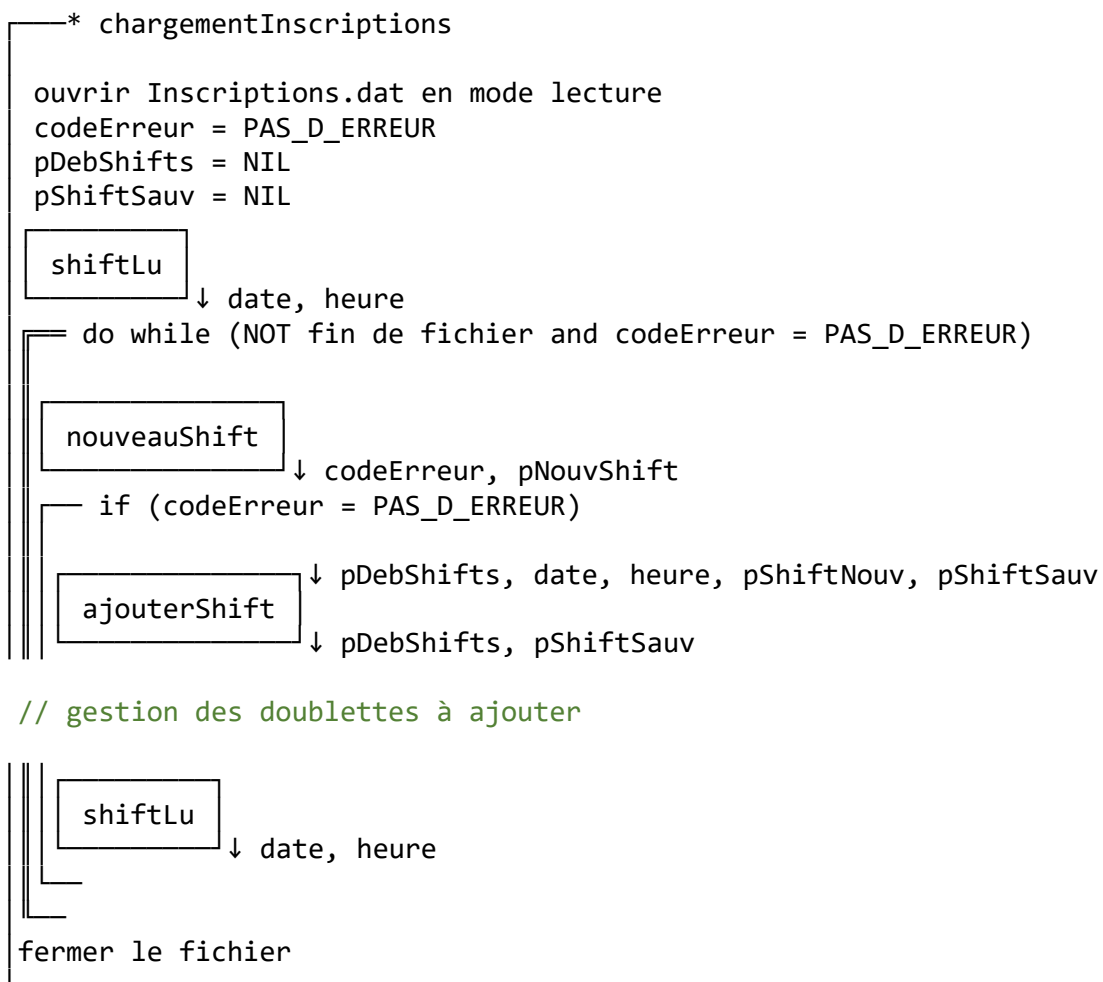


/\*\*\*\*\*\* FICHIER \*\*\*\*\*/

```
* detectionFichierShifts  
  
  codeErreur = PAS_D_ERREUR  
  ouvrir Inscriptions en lecture  
  if (erreur d'ouverture)  
    premiereFois = true  
    ouvrir DatesOrgShifts en lecture  
    if (erreur d'ouverture)  
      codeErreur = ERREUR_OUVERTURE_FICHIER  
    else  
      fermer DatesOrgShifts  
  else  
    premiereFois = false  
    fermer Inscriptions.dat
```

```
* initialiserShifts  
  
  detectionFichierShifts  
    ↓ premiereFois, codeErreur  
  if (codeErreur = PAS_D_ERREUR)  
    if (premiereFois)  
      chargementDatesOrgShifts  
        ↓ codeErreur, pDebShifts  
    else  
      chargementInscriptions  
        ↓ codeErreur, pDebShifts
```





\* sauverShifts

ouvrir Inscriptions.dat en écriture

if (erreur ouverture)

codeErreur = ERREUR\_OUVERTURE\_FICHER

else

codeErreur = PAS\_D\_ERREUR

pShift = pDebShifts

do while (pShift ≠ NIL)

shift.date = pShift → date

shift.heure = pShift → heure

shift.nbDoublettes = pShift → nbDoublettes

↓ shift

ecrireShift

pDoublette = pShift → pDebDoublettes

do while (pDoublette ≠ NIL)

inscription.matricule1 = pDoublette → matricule1

inscription.matricule2 = pDoublette → matricule2

inscription.categorie = pDoublette → categorie

↓ inscription

ecrireInscription

pDoublette = pDoublette→pSuiv

pShift = pShift→pSuiv

fermer Inscriptions.dat