

```

* Phase 3
pFiModel = ouvrir "fiModel.csv" en lecture
pFiTest = ouvrir "testSet.csv" en lecture
if(pFiModel ≠ NULL AND pFiTest ≠ NULL)
  ↓ pFiModel
  | convertFileToTable |
  ↓ models

// Chaque ligne du fichier sera représentée par la variable lineTestSet
lineTestSet = 1
/* Plutôt que de détailler la structure de la ligne (par exemple pour obtenir une valeur d'accélération : lire les nombres entre deux virgules), nous allons
simplement utiliser la notation lineTestSet.movement, lineTestSet.VacX (où le X représente le numéro de ce vecteur accélération), la notation n'est pas correcte mais
permet de se représenter aisément les choses */
nbTests = 0
do while(!EOF(pFiTest))
  // Pour chaque ligne
  movement = lineTestSet.movement
  iVac = 1
  // iVac représente la ième valeur de vecteur d'accélération sur la ligne en cours
  | EuclidianDistances initialisation |
  ↓ euclidianDistances
  do while(iVac < 601 and lineTest.iVac ≠ '\n')
    // Tant que l'on a pas lu les 600 valeurs de Vac et
    // que la valeur d'accélération lue est différente d'un retour à ligne (si tel est le cas : la ligne est incomplète pour les 60s demandées)
    iModel = 0
    do while(iModel < 6)
      difference = lineTest.iVac - models[iModel].(iVac-1)
      euclidianDistances[iModel] += difference * difference
      iModel++
    iVac++
  // Une fois la ligne complètement lue
  iMovement = 0
  bestMovement = 0
  bestEuclidianDistance = MAX_VALUE
  do while(iMovement < 6)
    euclidianDistances[iMovement] = sqrt(euclidianDistances[iMovement])
    if(bestEuclidianDistance > euclidianDistances[iMovement])
      bestMovement = iMovement+1
      bestEuclidianDistance = euclidianDistances[iMovement]
    iMovement++
  realClasses[nbTests] = movement
  estimateClasses[nbTests] = bestMovement
  nbTests++

| displayResultsForEachClass |
↓ realClasses, estimateClasses, nbTests
| displayAccuracy |
↓ realClasses, estimateClasses, nbTests
| displayClass |
↓ realClasses, estimateClasses, nbTests
Fermer "fiModel.csv"
Fermer "fiTest.csv"
else
  Afficher "Erreur d'ouverture"

```

```

| convertFileToTable |
↓ pFiModel
↓ models
*
lineModel = 1
| Models initialisation |
↓ models
do while (!EOF(pFiModel) AND lineModel ≤ 6)
  iVac = 1
  // iVac représente la ième valeur de vecteur d'accélération sur la ligne en cours
  models[lineModel-1].move = lineModel
  do while(iVac ≤ 600)
    // A priori, on aura toujours 600 valeurs d'accélération
    models[lineModel-1].v_acc[iVac-1] = lineModel.iVac
    iVac++
  lineModel++

```

```

| EuclidianDistances initialisation |
↓ euclidianDistances
*
i = 0
do while(i < 6)
  euclidianDistances[i] = 0
  i++

```

```

| Models initialisation |
↓ models
*
i = 0
do while(i < 6)
  models[i] = 0
  i++

```