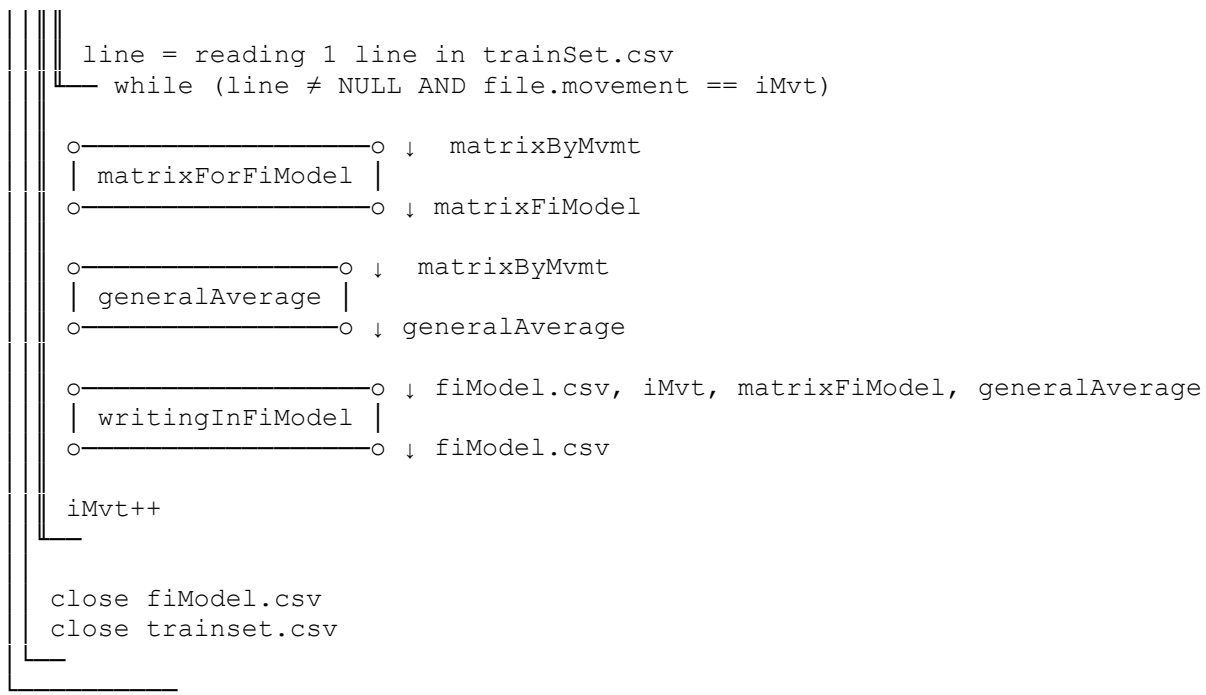
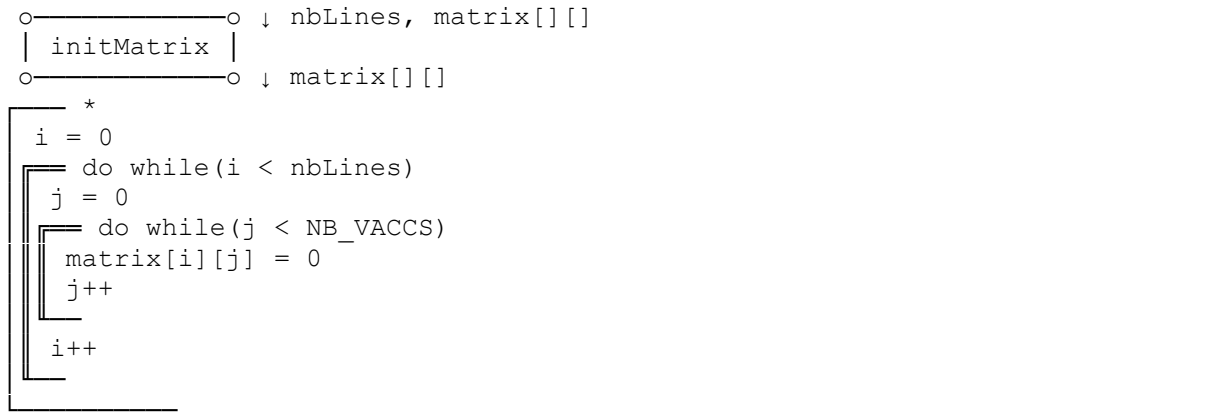
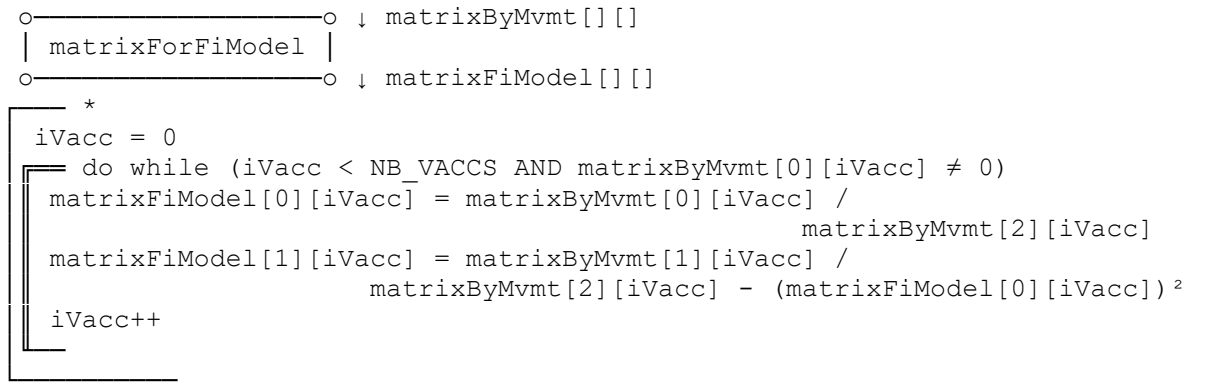
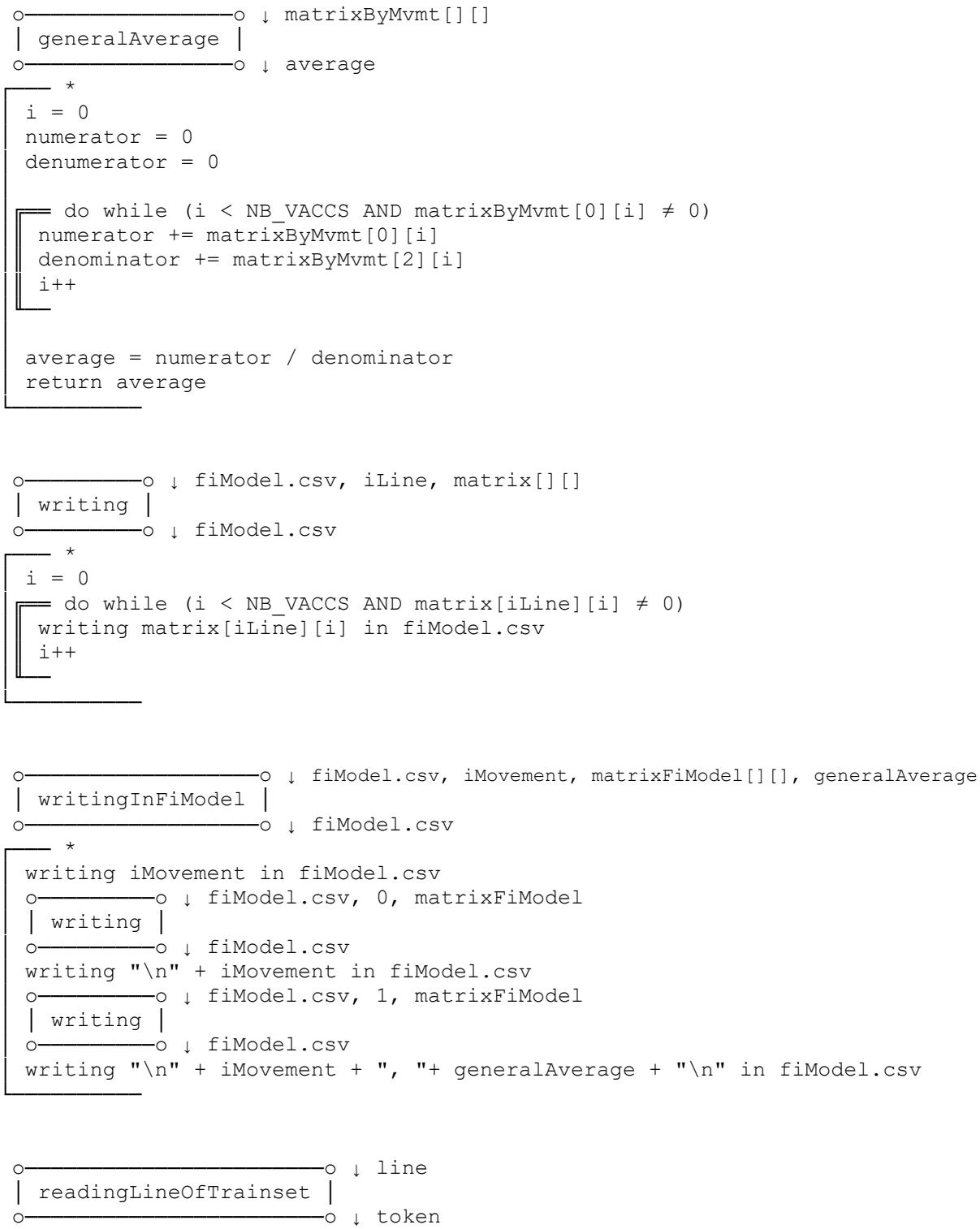
```
// LINE = 10000
 // NB_VACCS = 600
 // NB_MOVEMENTS = 6
 // NB_LINES_MATRIX_BYMVT = 3
 // NB_LINES_MATRIX_FIMODEL = 2

 // struct file {
 //    movement,
 //    gender,
 //    index,
 //    vAccs[NB_VACCS]
 // }
 o──────o
 │ Main │
 o──────o
┌─── *
│ open trainset.csv in reading
│ open fiModel.csv in writing
│
│┌── if(trainset.csv == NULL OR fiModel.csv == NULL)
│ output "Erreur d'ouverture de trainset.csv ou de fiModel.csv"
│
│├── else
│ writing "Mouvement" in fiModel.csv
│
│ i = 0
│╔══ do while (i < NB_VACCS)
│ writing ", vAcc" + i + 1 in fiModel.csv
│ i++
│╚══
│ writing "\n" in fiModel.csv
│
│ line = reading 1 line in trainSet.csv
│ iMvt = 1
│╔══ do while(iMvt ≤ NB_MOVEMENTS)
│
│ o──────────────o ↓ NB_LINES_MATRIX_BYMVT, matrixByMvmt
│ │ initMatrix │
│ o──────────────o ↓ matrixByMvmt
│
│ line = reading 1 line in trainSet.csv
│
│╔══ do
│ // reading data of line
│ o──────────────────────────o ↓ line
│ │ readingLineOfTrainset │
│ o──────────────────────────o ↓ token
│
│ i = 0
│╔══ do while (token ≠ NULL AND i < NB_VACCS)
│ file.vAccs[i] = atof(token)
│ token = strtok(NULL, ",")
│ i++
│╚══
│
│ iVacc = 0
│╔══ do while (iVacc < NB_VACCS AND file.vAccs[iVacc] > 0)
│ matrixByMvmt[0][iVacc] += file.vAccs[iVacc] // sum for averages
│ matrixByMvmt[1][iVacc] += (file.vAccs[iVacc])² //sum for standard deviation
│ matrixByMvmt[2][iVacc]++ // number of occurences by Vacc
│ iVacc++
│╚══
```


line = reading 1 line in trainSet.csv
while (line ≠ NULL AND file.movement == iMvt)
matrixForFiModel
↓ matrixByMvmt
↓ matrixFiModel
generalAverage
↓ matrixByMvmt
↓ generalAverage
writingInFiModel
↓ fiModel.csv, iMvt, matrixFiModel, generalAverage
↓ fiModel.csv
iMvt++
close fiModel.csv
close trainset.csv


initMatrix
↓ nbLines, matrix[][]
↓ matrix[][]
*
i = 0
do while(i < nbLines)
j = 0
do while(j < NB_VACCS)
matrix[i][j] = 0
j++
i++


matrixForFiModel
↓ matrixByMvmt[][]
↓ matrixFiModel[][]
*
iVacc = 0
do while (iVacc < NB_VACCS AND matrixByMvmt[0][iVacc] ≠ 0)
matrixFiModel[0][iVacc] = matrixByMvmt[0][iVacc] /
matrixByMvmt[2][iVacc]
matrixFiModel[1][iVacc] = matrixByMvmt[1][iVacc] /
matrixByMvmt[2][iVacc] - (matrixFiModel[0][iVacc])²
iVacc++


generalAverage
↓ matrixByMvmt[][]
↓ average
i = 0
numerator = 0
denumerator = 0
do while (i < NB_VACCS AND matrixByMvmt[0][i] ≠ 0)
numerator += matrixByMvmt[0][i]
denominator += matrixByMvmt[2][i]
i++
average = numerator / denominator
return average
writing
↓ fiModel.csv, iLine, matrix[][]
↓ fiModel.csv
i = 0
do while (i < NB_VACCS AND matrix[iLine][i] ≠ 0)
writing matrix[iLine][i] in fiModel.csv
i++
writingInFiModel
↓ fiModel.csv, iMovement, matrixFiModel[][], generalAverage
↓ fiModel.csv
writing iMovement in fiModel.csv
writing
↓ fiModel.csv, 0, matrixFiModel
↓ fiModel.csv
writing "\n" + iMovement in fiModel.csv
writing
↓ fiModel.csv, 1, matrixFiModel
↓ fiModel.csv
writing "\n" + iMovement + ", "+ generalAverage + "\n" in fiModel.csv
readingLineOfTrainset
↓ line
↓ token