

Algorithmique & Programmation

Fichiers et persistance des données

yann.secq@univ-lille.fr

ALMEIDA COCO Amadeu, BIRLOUEZ Martin, BONEVA Iovka,
DELECROIX Fabien, LEPRETRE Éric, MARSHALL-BRETON Christopher,
ROUZÉ Timothé, SECQ Yann, SOW Younoussa, SUE Yue

Tableau de promo

Souhait d'un tableau d'étudiant·e·s
(triés !) pour une année de BUT

- Les participant·e·s sont caractérisés par leur nom, prénom, numéro étu et une moyenne
- Souhait d'externaliser ces données afin qu'elles soient pérennes entre deux exécutions du programme
- Comment procéder ?

Tableau de promo

- Toujours la même démarche
 - Analyser et modéliser les données
 - Analyser et décomposer les traitements

Tableau de promo

- Qu'est-ce qui caractérise un-e étudiant-e
 - Nom, prénom : chaînes de caractères
 - N° étu : entier
 - Moyenne : double
- Création d'un type Etudiant avec des champs
- Et newEtudiant ainsi que toString(Etudiant)

Tableau de promo : traitements

- Quels sont les principaux traitements ?
- Charger un fichier avec les données de Etudiants existants
- Importer ces données dans une structure de données afin de pouvoir les trier sur les scores (et les mettre à jour éventuellement)
- Sauvegarder la SDD dans le fichier initial

Type Etudiant

```
class Etudiant {  
    String nom;  
    String prenom;  
    int numEtu;  
    double moyenne;  
}
```

```
// Dans le fichier du programme principal
```

```
Etudiant newEtudiant(String nom, String prénom, String pseudo, int num,  
double moyenne) {  
    Etudiant nouveau = new Etudiant();  
    nouveau.nom      = nom;  
    nouveau.prenom   = prenom;  
    nouveau.numEtu   = num;  
    nouveau.moyenne  = moyenne;  
    return nouveau;  
}
```

```
String toString(Etudiant e) {  
    return j.pseudo + "(" + nom + "," + prenom + ") : " +  
moyenne);  
}
```

Load and store

```
import extensions.CSVFile;

// Dans le fichier du programme principal

Etudiant[] load(String nomFichier) {
    CSVFile EtudiantsAsString = loadCSV(nomFichier);
    Etudiant[] Etudiants = new
Etudiant[rowCount(EtudiantsAsString)];
    for (int idxJ=0; idxJ < length(Etudiants); idxJ++) {
        String nom      = getCell(EtudiantsAsString, idxJ, 0);
        String prenom    = getCell(EtudiantsAsString, idxJ, 1);
        double moyenne   = stringToDouble(EtudiantsAsString,
idxJ, 2);
        int numEtu       = stringToInt(getCell(EtudiantsAsString,
idxJ, 3));
        Etudiant courant = newEtudiant(nom, prenom, numEtu,
moyenne);
        Etudiants[idxJ]  = courant;
    }
    return Etudiants;
}

// introduire des fonctions facilitant la mise à jour des Etudiants
```

Update

// Dans le fichier du programme principal

```
Etudiant trouver(Etudiant[] etudiants, String pseudo) {  
    Etudiant etu = null;  
    int idx = 0;  
    while (idx < length(etudiants) && etudiants[idx].numEtu !=  
numEtu) {  
        idx++;  
    }  
    if (idx < length(etudiants)) {  
        etu = etudiants[idx];  
    }  
    return etu;  
}  
  
boolean setMoyenne(Etudiant[] etudiants, int numEtu, double  
moyenne) {  
    boolean succes = false;  
    Etudiant etu = trouver(etudiants, numEtu);  
    if (etu != null) {  
        etu.moyenne = moyenne;  
        succes = true;  
    }  
    return succes;  
}
```


Save

```
// Dans le fichier du programme principal
void sauver(Etudiant[] etudiants, String nomFichier) {
    String[][] etudiantsAsString new String[length(etudiants)]
[4];
    for (int idxJ=0; idxJ < length(Etudiants); idxJ++) {
        etudiantsAsString[idxJ][0] = etudiants[idxJ].nom;
        etudiantsAsString[idxJ][1] = etudiants[idxJ].prenom;
        etudiantsAsString[idxJ][2] = "" +etudiants[idxJ].numEtu;
        etudiantsAsString[idxJ][3] = "" +etudiants[idxJ].moyenne;
    }
    saveCSV(etudiantsAsString, nomFichier) ;
}
```