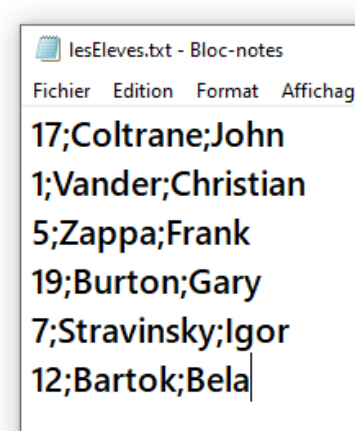
Un fichier est une collection d'informations (enregistrements) stockée sur un support de mémoire durable (disque dur magnétique, disque SSD, clé USB…) dans le but de mémoriser ces informations entre deux exécutions d'un programme.

Un fichier séquentiel est un fichier "texte pur" où les informations sont enregistrées ligne par ligne, par exemple voici un extrait du contenu du fichier "lesEleves.txt" :



L'accès aux enregistrements se fait de façon séquentielle, c'est à dire qu'à la première lecture du fichier on accède au 1er enregistrement, à la deuxième lecture au deuxième enregistrement et ainsi de suite jusqu'à la fin du fichier. Il n'est donc pas possible d'accéder directement au 33ème enregistrement, il faut lire les 32 autres avant !

1) Les modes d'exploitation

Dès l'ouverture d'un fichier il faut préciser la façon dont on va l'exploiter :

- Lecture (L), // consultation du fichier
- Ajout (A), // ajout en fin de fichier
- Ecriture (E), // on écrase tout et on ajoute

Voici comment ouvrir un fichier en C# dans les 3 modes :

Nécessite l'import de la bibliothèque IO : using System.IO ;

Déclarer le fichier en mode ajout :

```
FileStream fs = new FileStream("lesEleves.txt", FileMode.Append, FileAccess.Write);
```

Déclarer le fichier en mode écriture (le contenu initial est préalablement vidé) :

```
FileStream fs = new FileStream("lesEleves.txt ", FileMode.Create, FileAccess.Write);
```

Déclarer le fichier en mode lecture :

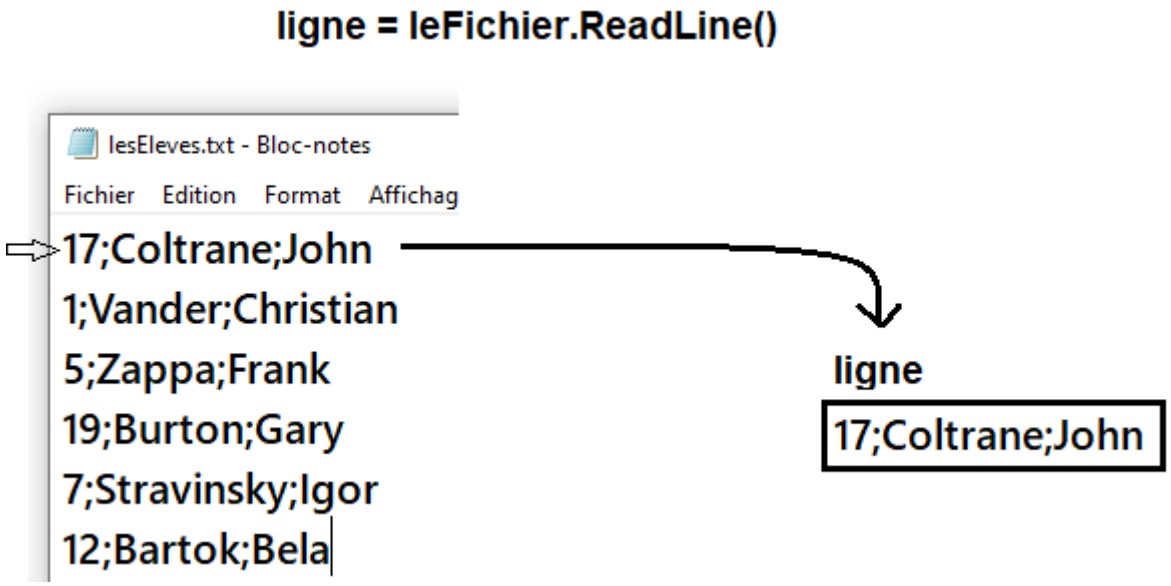
```
FileStream fs = new FileStream("lesEleves.txt ", FileMode.Open, FileAccess.Read);
```

2) Les commandes algorithmiques d'exploitation de fichier

a) Lire le contenu d'un fichier existant

Lire la première ligne du fichier :

```
//ouvrir le fichier en mode lecture
FileStream fs = new FileStream("lesEleves.txt", FileMode.Open, FileAccess.Read);
//ouvrir un flux sur ce fichier en mode lecture
StreamReader leFichier = new StreamReader(fs);
string ligne = leFichier.ReadLine();
MessageBox.Show(ligne);
leFichier.Close();
```



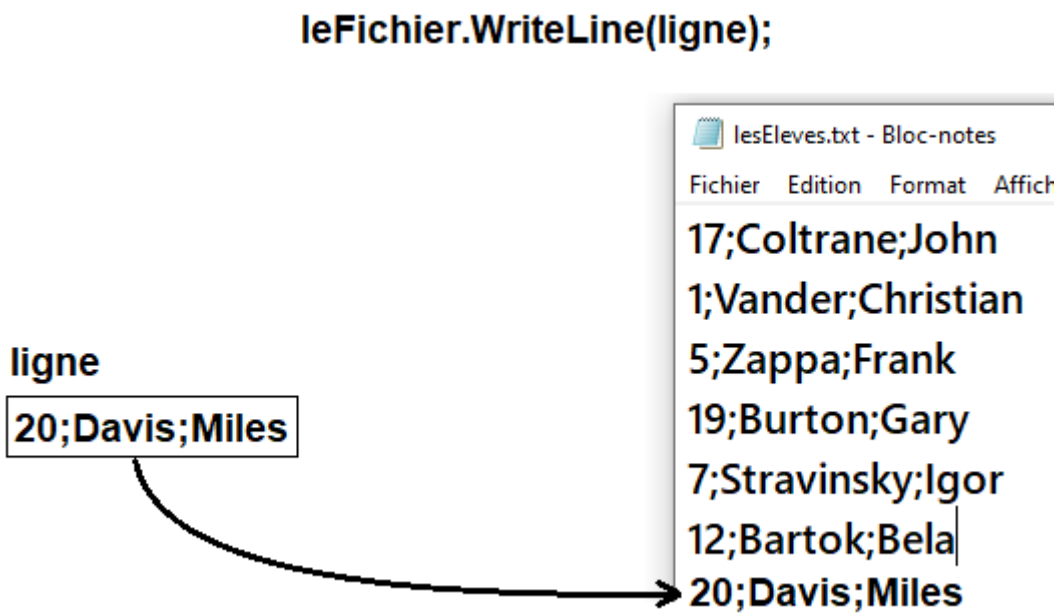
Lire toutes les lignes du fichier :

```
//ouvrir le fichier en mode lecture
FileStream fs = new FileStream("lesEleves.txt", FileMode.Open, FileAccess.Read);
//ouvrir un flux en mode lecture
StreamReader leFichier = new StreamReader(fs);

//Amorce
string ligne = leFichier.ReadLine();
while (ligne != null) {
	MessageBox.Show(ligne);
	//lire l'enregistrement suivant
	ligne = leFichier.ReadLine();
}
leFichier.Close();
```

b) Ecrire dans le fichier (en mode ajout)

```
FileStream fs = new FileStream("lesEleves.txt", FileMode.Append, FileAccess.Write);
StreamWriter leFichier = new StreamWriter(fs);
String ligne;
ligne = "20;Davis;Miles";
leFichier.WriteLine(ligne);
leFichier.Close();
```

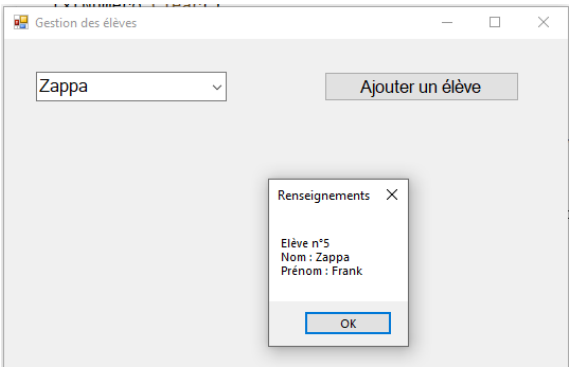


3) Exercice d'application

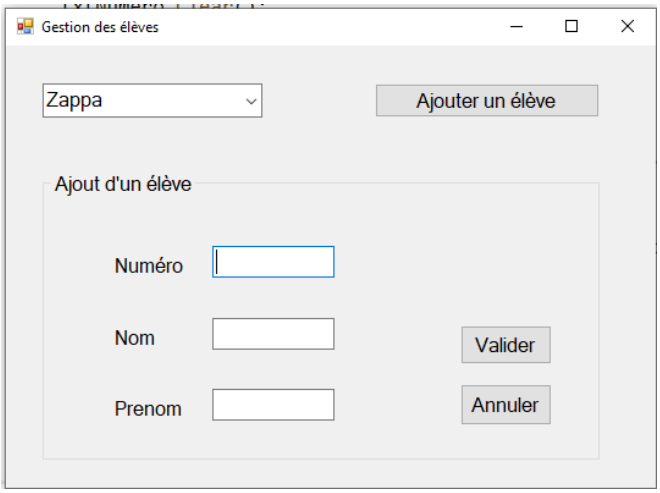
Créer une nouvelle application qui charge une collection de chaînes déclarée en global `List<string> lesEleves;` à partir du fichier **`lesEleves.txt`** fourni.

Remplir ensuite à partir de la collection une comboBox avec le nom des élèves

Le choix d'un élève dans la combo affiche dans un MessageBox ses informations :



Le bouton Ajouter rend visible le GroupBox qui permettra de saisir les informations d'un nouvel élève :



Le bouton Valider ajoute le nouvel élève dans le fichier lesEleves.txt et son nom dans la comboBox.

Dans les 2 cas les boutons Valider et Annuler devront cacher à nouveau le GroupBox.

N'oubliez pas ! Une fois les composants placés sur le formulaire, il faut commencer par les renommer !