ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Points sur le projet informatique 2020

NORA IZRI – DÉPARTEMENT IBO

ESILV
LÉONARD
DE VINCI

ÉCOLE
D'INGÉNIEURS
PARIS-LA DÉFENSE

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Points

Organisation :

-Conception/réflexions

-Implémentation ➔ programmation c#

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Fichiers en C#

NORA IZRI – DÉPARTEMENT IBO

ESILV
LÉONARD
DE VINCI

ÉCOLE
D'INGÉNIEURS
PARIS-LA DÉFENSE

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Pourquoi des fichiers?

Stockage de l'information de manière persistante.

- un fichier est identifié par un chemin, un nom et un type (extension)
- les données contenues dans un fichier sont organisées ➔ cette organisation dépend des données : texte, image, son etc.

L'extension (type) donne une indication sur la nature et l'organisation des données: texte brut (txt), document text formaté (doc, docx, tex, …), image (jpg, png, gif, …), etc.

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Récupération dans un tableau de string

```
string[] lecturefichier = File.ReadAllLines("Des.txt");
for (int i = 0; i < lecturefichier.Length; i++)
{
        Console.WriteLine(lecturefichier[i]);
}
File.WriteAllLines("DesSortie1.txt", lecturefichier);
```

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Fichiers « txt »

Fichier « txt » ➔ texte brut

Enregistrer le nom du fichier complet dans une variable « string »

```
string nomFichier = "C:\\Documents\monFichier.txt";
```

- Lecture/écriture ligne par ligne du début à la fin du fichier.
- Chaque ligne lue ou écrite sera contenue dans un string.
- Les entrées et sorties de données depuis ou vers le fichier constituent un flux de données.
- Pour lire ou écrire un flux de données, on utilise de nouveaux type d'objets (classes StreamWriter et StreamReader).

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Ecriture fichier « txt »

1. Définir le nom du fichier
2. Générer un objet flux de sortie ***StreamWriter*** vers le fichier
3. Ecrire dans le flux en utilisant la méthode WriteLine()
4. Fermer le fichier à la fin des écriture

Exemple > Ecriture fichier « txt »

- **S**treamWriter est une classe ➔ Attention le **S** en majuscule
- fichierEcriture est un objet de type StreamWriter ➔ accès à des méthodes préécrites
- Ne pas oublier la fermeture du fichier **Close()** sinon le fichier ne sera pas enregistré.

```
string nomFich = "C:\\Documents\\sortie.txt" ; \\ Etape 1
StreamWriter fichEcriture = new StreamWriter (nomFich, true); \\ Etape 2
fichEcriture.WriteLine ("1ere ligne") ; // ECRIRE UNE LIGNE ➔ Etape 3
fichEcriture.WriteLine ("2eme ligne") ;
\\ etc...
fichEcriture.Close() ; // ON FERME LE FICHIER ➔ Etape 4
```

true → ajouter des données si le fichier existe

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Lecture fichier « txt »

1. Définir le nom du fichier
2. Générer un objet flux de lecture ***StreamReader*** vers ce fichier
3. Lire dans le flux avec la méthode ReadLine() de l'objet flux
4. Fermer le fichier à la fin des lectures

Exemple > Lecture fichier « txt »

- StreamReader est une classe ➔ Attention le **S** en majuscule
- fichLecture est un objet de type StreamReader ➔ accès à des méthodes préécrites.

```
string nomFich = "C:\\Documents\\sortie.txt" ; \\ Etape 1
StreamReader fichLecture = new StreamReader (nomFich); \\ Etape 2
string ligne= " ";
 while(fichLecture.Peek()>0){
      ligne=fichLecture.ReadLine(); //lecture d'une ligne
      Console.WriteLine("Lu : " + ligne);
}
fichLecture.Close() ; // ON FERME LE FICHIER   ➔ Etape 4
```

Librairie à indiquer

- Librairie **obligatoire** à mentionner en haut du programme :

```
using System.IO;
```

➔ File, StreamReader, StreamWriter

Fichiers « **csv** »

- Séparation des données par des « ; »
- Organisation des données ligne par ligne. Chaque ligne contient des colonnes.
- Exemple : nous souhaitons enregistrés les informations de 3 clients :
  - Cotillard ;Marion ;37 ;45765 ;rue des vagues ;Lyon
  - Rochefort ;Jean ;70 ;56678 ;rue des perdrix ;Marseille
  - Blier ;Bernard ;68 ;33457 ;rue Leonard ;Paris

	A	B	C	D	E	F
1	Cotillard	Marion	37	45765	rue des vagues	Lyon
2	Rochefort	Jean	70	56678	rue des perdrix	Marseille
3	Blier	Bernard	68	33457	rue Leonard	Paris

Exemple > Ecriture fichier « csv »

- Construire chaque ligne à écrire dans le fichier en intercalant un « ; » entre chacune des données (data1, data2 etc) qui constituent une ligne du fichier.
- Ecrire la ligne sur le flux

```
string nomFich = "C:\\Documents\\clients.csv" ; \\ Etape 1
StreamWriter fichEcriture = new StreamWriter (nomFich,true); \\ Etape 2
string ligne = "Cotillard" + ";" + "Marion" + ";" + 37 + ";" + 45795 + ";" + "rue des vagues" + ";" + "Lyon"; \\ Etape 3
fichEcriture.WriteLine (ligne); // ECRIRE UNE LIGNE ➔ Etape 4
fichEcriture.Close() ; // ON FERME LE FICHIER ➔ Etape 5
```

Lecture fichier « csv »

- Ce sont des fichiers texte qui se lisent donc comme des fichiers texte
- Chaque lecture dans le fichier lit un string composé d'une série de data séparées par des '**;**'
- Pour séparer ces informations on utilise une méthode des objets string : la méthode **Split**
- string string.Split(char[]) : On passe en paramètre un tableau de char correspondant à la liste des séparateurs à utiliser (ici uniquement le ';')

Exemple > Lecture fichier « csv »

```
string nomFich = "C:\\Documents\\clients.csv" ;
StreamReader fichLect = new StreamReader (nomFich) ;
char[] sep = new char[1] {';'};
string ligne = "";
string[] datas = new string[6];
While (fichLect.Peek() > 0)
{
    ligne = fichLect.ReadLine() ; // LECTURE D'UNE LIGNE
    Console.WriteLine( "ligne lue : "+ ligne);
    datas = ligne.Split(sep);
}
\\ etc...
```

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

La classe « FILE »

Classe « File »

- Plusieurs méthodes utiles :
  - Existence d'un fichier => bool File.Exists (string nomFich)
  - Supprimer fichier => void File.Delete (string nomFich)
  - Copier fichier => void File.Copy (string source, string dest)
  - Déplacer fichier => void File.Move (string source, string dest)

```
if ( ! File.Exists ("c:\\Douments\\monFichier.txt") ) {
    Console.WriteLine ( "Fichier inexistant") ;
}
```

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Classe « File »

Des méthodes d'instanciation d'objet de contrôle de flux de données (StreamReader, SreamWriter)

. **Lecture de fichier** : StreamReader fichLect = File.OpenText (nomFich) ;

. **Ecriture de fichier** :

- StreamWriter fichEcr = File.CreateText (nomFich) ;
- StreamWriter fichEcr = File.AppendText (nomFich) ;
- si nomFich existe déjà, il sera écrasé par CreateText

```
StreamReader fichLect = File.OpenText(nomFichier) \\
StreamWriter fichEcr = File.CreateText(nomFichier)
StreamWriter fichEcr = File.AppendText(nomFichier)
```

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Exercices

ESILV
ENGINEERING SCHOOL
DE VINCI PARIS

Exercice 1

Ecrire une méthode qui prend en entrées une chaine de caractères représentant le nom d'un fichier et un mot « x » puis retourne le nombre de fois que le mot « x » est présent dans le fichier.

Exercice 2

On considère un fichier contenant ce type d'informations

N° client;**N**om ;**P**rénom;**C**ommande1; **C**ommande2; **C**ommande3; **C**ommande4

1. Ecrire une méthode prenant en entrée une chaîne de caractères correspondant au nom du fichier à traiter et retourne le prix moyen dépensé par chaque client en construisant un nouveau fichier "PrixMoyenCommande.txt" comme suit :
   - Pour chaque client :
     - Calculer le prix moyen de ces 4 commandes.
     - Rajouter dans le nouveau fichier; les informations d'un client (N° client, nom, prénom) ainsi que le prix moyen dépensé pour les commandes.

Exercice 3

On considère un fichier contenant ce type d'informations

N° client;**N**om ;**P**rénom;**M**oyenneCom

2. Ecrire une méthode qui prend en entrée le nom d'un fichier; exemple celui construit à la question précédente "PrixMoyenCommande.txt"
   a. Déterminer et afficher la moyenne la plus basse ainsi que la plus élevée.
   b. Calculer la moyenne de tous les clients en additionnant au fur et à mesure les moyennes individuelles puis en divisant sur le nombre de clients total (nombre de lignes dans un fichier).