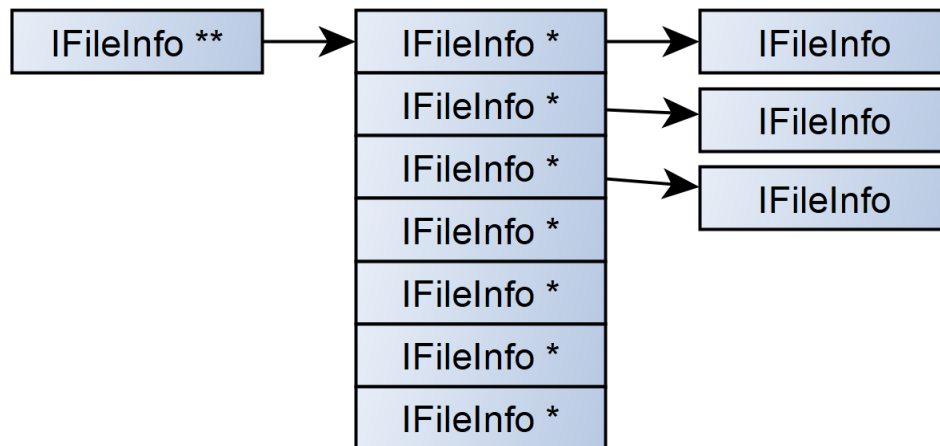


# Travail 1 – SuperDir – Étape 2

La fonction `FindFiles` retourne une valeur du type `IFileInfo **`. Regardons ce type de plus près. `IFileInfo` est le nom d'une classe. Souvent, les développeurs utilisent des noms de classe qui débute par un `I` pour identifier que cette classe est en fait une interface. Les deux `*` nous indiquent que nous avons affaire à un tableau de pointeur vers des instances de la classe `IFileInfo`.



La case complètement à gauche dans le diagramme représente la valeur retournée par la fonction `FindFiles`. Celle-ci est l'adresse d'une case mémoire contenant l'adresse d'une instance de `IFileInfo`. Cette case mémoire est suivie d'autres cases mémoire similaires contenant aussi l'adresse d'une instance de `IFileInfo`. Ici, le diagramme ne montre que 7 de ces cases mémoire alors que l'énoncé du travail 1 vous demande d'en allouer 100.

Dans le diagramme, les 4 dernières cases ne pointent pas vers une instance de `IFileInfo`. Ces cases contiennent la valeur **NULL**. Cela indique tout simplement qu'il n'y a pas assez de fichiers dans le répertoire pour créer d'autres instances de `IFileInfo`.

## 1 Création du tableau

La fonction doit créer le tableau. Un tableau peut être dynamique ou statique. Cependant, comme la fonction doit retourner le tableau, il faut absolument créer un tableau dynamique. Un tableau statique dans une fonction est une variable locale. Il cesserait donc d'exister aussitôt que la fonction retournerait et il serait donc impossible de le retourner.

```
IFileInfo ** lFiles = new IFileInfo * [100];
```

Cette ligne de code crée le tableau demandé. Comme toute mémoire dynamique, ce tableau n'est pas initialisé par le langage C++. Il faut donc initialiser chacun des 100 pointeurs à **NULL**.

## 2 Création des instances de IFileInfo

Une fois le tableau créer, il faut chercher les fichiers qui se trouve dans le répertoire et pour chacun d'eux, créer une instance d'une des classes suivantes.

- FileInfo\_CPP
- FileInfo\_EXE
- FileInfo\_H
- FileInfo\_Other

Pour savoir laquelle des 4 classes utiliser, il faut regarder l'extension du fichier.

```
lFiles[0] = new FileInfo_CPP;
```

La ligne de code précédente crée une instance de la classe FileInfo\_CPP et place l'adresse de celle-ci dans le tableau.

Une fois l'instance créer, il faut l'initialiser en appelant la méthode Init. La méthode Init est déclarée au niveau de la classe IFileInfo, comme une méthode purement virtuelle. Il faut aussi la déclarer au niveau de la classe FileInfo\_Base. Le code de celle-ci doit se trouver dans la classe FileInfo\_Base (dans le fichier source FileInfo\_Base.cpp).

Ce code est très simple, il copie simplement le nom du répertoire et le nom du fichier dans les variables membre défini dans la classe FileInfo\_Base.

## 3 Sources d'informations

# 1	Listing the Files in a Directory <a href="https://docs.microsoft.com/en-us/windows/win32/fileio/listing-the-files-in-a-directory">https://docs.microsoft.com/en-us/windows/win32/fileio/listing-the-files-in-a-directory</a> (2022-02-10)
-----	---