Le langage Java
- Piles et Files-

guest stars: les fichiers et les exceptions

Lire des fichiers en Java

- Chemin d'accès à un fichier
  - Sa position sur le disque dur
  - Classe « Path » permet d'avoir beaucoup d'information

- Création d'un objet Path:

```
Path cheminFichier = Paths.get("~./PI2/truc.txt");
```

- Services:

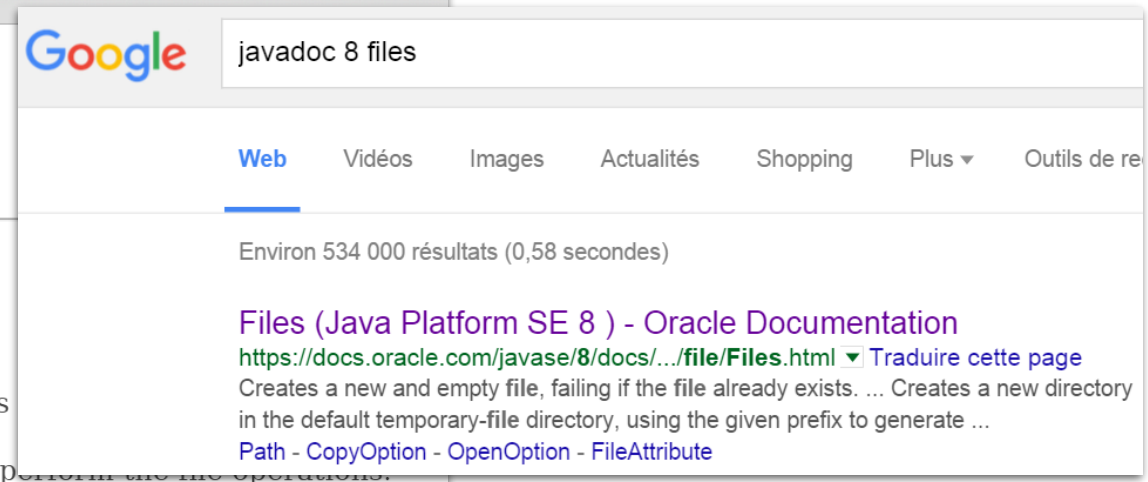
```
Files.size(cheminFichier);  // taille en octets
Files.getLastModifiedTime(cheminFichier); // date de dernière modification
Files.isDirectory(cheminFichier); // test si répertoire ou pas
…
```

https://docs.oracle.com/javase/8/docs/api/java/nio/file/Files.html

Applications Accédez rapidement à vos favoris en les ajoutant à la barre de favoris. Importer mes favoris maintenant...

Class Files

java.lang.Object
    java.nio.file.Files




```
public final class Files
extends Object
```

This class consists exclusively of static methods that operate on files, directories, or other types

In most cases, the methods defined here will delegate to the associated file system provider to p

Since:

1.7

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method and Description
static long	**copy**(**InputStream** in, **Path** target, **CopyOption**... options) Copies all bytes from an input stream to a file.
static long	**copy**(**Path** source, **OutputStream** out) Copies all bytes from a file to an output stream.
static **Path**	**copy**(**Path** source, **Path** target, **CopyOption**... options) Copy a file to a target file.
static **Path**	**createDirectories**(**Path** dir, **FileAttribute**<?>... attrs) Creates a directory by creating all nonexistent parent directories first.
static **Path**	**createDirectory**(**Path** dir, **FileAttribute**<?>... attrs)

Interlude: récupération d'erreur

- Si on écrit une erreur:
  - Ex: `int x = 5 / 0;`

- On obtient:

```
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at cm5_prep.Erreur.main(Erreur.java:7)
```

  - C'est une **Exception**
  - Ca fait planter le programme (pas bien)
  - …sauf si on la gère (bien !)

Gestion d’exception

- Identifier les blocs de code pouvant déclencher une exception
- Les entourer avec :

```
try {
    …
}
```

- « Attraper » les exceptions ensuite :

```
catch (Exception e) {
    // traitement
}
```

Exemple

```
public static void main(String[] args) {
	try {
		int x = 5/0;
	} catch (Exception e) {
		System.out.println("Quelque chose s'est mal passé...");
	}
	System.out.println("Mais le programme n'a pas planté !");
}
```

…plus de détails au tableau…

Retour sur la lecture d’un fichier

- Beaucoup de choses peuvent mal se passer !
  - Erreur dans le nom de fichier, fichier absent, fichier corrompu…
- On est **obligé** de combiner lecture et traitement d’exception
- Les exceptions de fichier sont de type IOException

Code de lecture/écriture d'un fichier

- *…fait au tableau…*

Doc détaillée en anglais ici: http://docs.oracle.com/javase/tutorial/essential/io/file.html

Un problème

- Dans quelle structure de données stocker le contenu du fichier ?

Structures de données dynamiques

- Nous connaissons les tableaux et les classes
- La taille est fixée au départ
- Pas adapté pour des données dont la taille n'est pas connue à l'avance

- Structures de données dynamiques
  - Pas de taille connue à l'avance
  - Peuvent grandir / rapetisser à volonté
  - Element de base pour de très nombreuses applications
  - Basées sur le mécanisme des références que vous connaissez bien maintenant !

Pile

- *…Schéma au tableau…*

Element d'une Pile

- Classe simple qui contient :
  - Une valeur à stocker
  - Une référence sur l'élément précédent dans la pile

- Code:

```
class Element {
        int v ;
        Element precedent ;

        Element(int v, Element precedent) {
                this.v = v ;
                this.precedent = precedent ;
        }
}
```

Classe Pile

- On va faire une « pile » d'éléments
- Que stocker comme attribut(s) ?
  - …

- Méthodes nécessaires :
  - boolean estVide() : dit si la Pile est vide ou non
  - void empile(int x) : ajoute x en haut de la Pile
  - int depile() : retire la valeur du haut de la pile, et la renvoie

La classe File

- Pile : Last In First Out (LIFO)

- On veut maintenant faire une File : First In First Out (FIFO).
- Comment faire ?

  - *… au tableau…*