



## Java : exercice 6.4

### Écrire dans un fichier

Reproduire et exécuter la classe `TestIO` telle que définie dans la partie de cours consacrée aux flux.

Remarques :

- il faudra que la fonction `main` déclare une exception de type `IOException` ; vous pouvez également introduire un bloc `try-catch` pour récupérer l'exception qui se produit si le nom de fichier est vide (`...new FileWriter("") ;`)
- le fichier créé se trouve à la racine du projet Eclipse (faire éventuellement un clic-droit > Refresh sur le projet)

### Lire depuis l'entrée standard

Reproduire et exécuter la classe `TestIO2` telle que définie dans la partie de cours consacrée aux flux.

### Recopie de fichier

Reproduire et exécuter la classe `TestIO3` telle que définie dans la partie de cours consacrée aux flux.

Remarques :

- il faudra que la fonction `main` déclare une exception de type `IOException` ;
- vous devrez également introduire un bloc `try-catch` pour récupérer l'exception qui se produira si le fichier à lire n'existe pas.

### Retour à l'exercice 6.2

On reprend la classe `Exercice62` dans laquelle nous avons commencé à implémenter la bibliothèque `iText`.

```
public class Exercice62 {  
    public static void main(String[] args) {  
        // création du document  
        Document document = new Document();  
        document.addAuthor("moi");  
        document.addCreationDate();  
        document.addTitle("premier document");  
    }  
}
```

Nous créons un bloc `try-catch`

```
try {  
    ...  
} catch (FileNotFoundException e) {  
    e.printStackTrace();  
} catch (DocumentException de) {  
    System.err.println(de.getMessage());  
}
```

avec l'intention de lever deux exceptions : la première si le document n'existe pas, la seconde en cas d'erreur sur le document.

Puis, à l'intérieur de ce bloc

- on crée un flux de sortie vers un fichier  
`FileOutputStream fos = new FileOutputStream("fichier1.pdf");`
- on crée un objet `PdfWriter` qui va « écouter » ce qui se passe sur l'objet `document` et le

## Initiation à Java – GMM4

redirige vers le flux de sortie :

```
PdfWriter.getInstance(document, fos);
```

- on ouvre le document, écrire un paragraphe et le fermer (l'objet de type PdfWriter a pris en compte ces modifications et les a transmises au flux de sortie)

```
document.open();
```

```
document.add(new Paragraph("paragraphe 1"));
```

```
document.close();
```

Le fichier fichier1.pdf est créé !