

- **Cas n°1** : On tombe sur une exception dans méthode2 introduite par méthode1 et on décide de la traiter : utilisation de *"try{...}catch{...}"*

```
public void méthode2(){
    try
    {
        méthode1();
    }
    catch(MonException exception)
    {
        // Traitement de l'exception
    }
    finally
    {
        //Instruction faite même si on tombe sur l'exception (bloc qui n'est pas
        //obligatoire)
    }
}
```

- **Cas n°2** : On tombe sur une exception dans méthode2 introduite par méthode1 et on décide de la renvoyer : utilisation de *"throws"* (avec un s)

```
public void méthode2() throws MonException{
    méthode1();
    //Comme on a mis "throws MonException", toutes le méthodes qui throws
    //MonException comme méthode1() n'auront pas besoin de traiter
    //MonException au sein de cette méthode.
}
```

- En fait on aurai pu aussi faire le **Cas n°2** avec le **Cas n°1** en décidant de traiter l'exception et de renvoyer la même:

```
public void méthode2(){
    try{
        méthode1();
    }
    catch(MonException exception)
    {
        throw exception;
    }
}
```

Attention: Dans une fonction on utilise le mot clé *"throw"* pour envoyer un objet Exception alors qu'on utilise le mot clé *"throws"* dans la signature d'une fonction.