

Quand on exécute une telle instruction, on commence par exécuter l'instruction `p`. Si l'exécution de cette instruction lève une exception, alors on interrompt l'exécution de l'instruction `p` et on exécute l'instruction `q`. Ainsi, si `x` est une clé de `l`, on affiche la valeur associée à cette clé et, si ce n'est pas le cas, on affiche la chaîne de caractères "Pas dans la liste".

Dans cet exemple, toutes les exceptions déclenchent l'exécution de l'instruction `System.out.println("Pas dans la liste")`; . Il est aussi possible, en Java, de définir des exceptions distinctes les unes des autres et de n'exécuter de cette instruction que quand une certaine exception est levée.

Dans une séquence d'instructions `{p1 p2}`, si l'instruction `p1` lève une exception, alors l'instruction `p2` n'est pas exécutée. De ce point de vue, l'instruction `throw` ressemble beaucoup à l'instruction `return`.

7.4 La propagation des exceptions

On peut utiliser la fonction `assoc` pour écrire une fonction `assocplus` qui renvoie la valeur associée à une clé dans une liste, plus 1. Et comme à chaque fois que l'on utilise la fonction `assoc`, on doit indiquer ce qu'il faut faire quand la fonction `assoc` lève une exception. On peut, par exemple, décider que la fonction `assocplus` elle-même lève une exception dans ce cas.

```
static int assocplus
    (final int x, final List l) throws Exception {
    try {return assoc(x,l) + 1;}
    catch (Exception e) {throw e;}}
```

Il n'est, cependant, pas nécessaire d'être aussi explicite, car si, dans une fonction `g`, on utilise une fonction `f` qui peut lever une exception, sans la protéger par un `try`, alors quand `f` lève une exception, c'est aussi le cas de la fonction `g`. Ainsi, la fonction `assocplus` peut s'écrire plus simplement

```
static int assocplus
    (final int x, final List l) throws Exception {
    return assoc(x,l) + 1;}
```

Et l'instruction

```
try {System.out.println(assocplus(x,l));}
catch (Exception e) {System.out.println("Pas dans la liste");}
```

affichera la valeur associée à `x` dans `l` plus 1 si cette clé apparaît dans la liste et la chaîne de caractères "Pas dans la liste" sinon.