

## Cohésion et lisibilité

Une conception avec un haut degré de cohésion présente de nombreux avantages dont les deux principaux sont la *lisibilité* et la *réutilisabilité*.

L'exemple proposé précédemment à la section « Cohésion des méthodes » (cohésion de la méthode `printWelcome`) est clairement un exemple où, en augmentant la cohésion, une classe devient plus lisible et ainsi plus facile à comprendre et à maintenir.

La cohésion de la classe présentée dans la section précédente (« Cohésion des classes ») facilite aussi la compréhension. S'il existe une classe `Item` séparée, un programmeur chargé de la maintenance trouvera facilement où il doit commencer à lire le code si une modification des caractéristiques d'un objet est nécessaire. La cohésion des classes améliore également la lisibilité d'un programme.

## Cohésion et réutilisation

Le deuxième avantage principal offert par la cohésion est le potentiel de réutilisation.

L'exemple de la section « Cohésion des classes » met aussi en évidence ce point : en créant une classe séparée `Item`, nous pouvons créer de multiples objets, et donc utiliser le même code pour plus d'un objet.

La réutilisation est aussi un aspect important de la cohésion des méthodes. Considérons une méthode de la classe `Room` possédant la signature suivante :

```
public Room leaveRoom(String direction)
```

Cette méthode peut renvoyer la pièce accessible dans une direction donnée (utilisable comme nouvelle valeur de `currentRoom`) et aussi afficher la description de la nouvelle pièce dans laquelle nous venons juste d'entrer.

Cette conception semble réalisable et on peut effectivement la mettre en œuvre. Dans notre version, nous avons cependant séparé cette tâche en deux méthodes :

```
public Room getExit(String direction)
public String getLongDescription()
```

La première méthode est chargée de retourner la prochaine pièce, alors que la seconde produit la description de la pièce.

L'avantage de cette conception est que les tâches séparées peuvent être réutilisées plus facilement. Par exemple, la méthode `getLongDescription` est non seulement utilisée par la méthode `goRoom`, mais aussi par `printWelcome` et pour implanter la commande `regarder`. Tout cela est possible grâce au fort degré de cohésion. Il ne serait pas envisageable de réutiliser cette méthode en s'appuyant sur la version contenant la méthode `leaveRoom`.

