

Normes de programmation TourGuide

Ces normes reposent sur l'article *The Elements of Modern Java Style* de DZone.

Les « coziéd braces » placent l'accolade d'ouverture juste à la fin de l'instruction d'origine et l'accolade de clôture sur sa propre ligne. Les mots-clés associés (comme « else » jusqu'à « si ») sont cozy jusqu'à l'accolade de clôture sur la même ligne :

```
if(condition) {  
    statement;  
else {  
    statement2;  
}
```

Pour garder une base de code cohérente, utilisez les conventions de nommage suivantes :

- capitalisez la première lettre des classes et des interfaces ;
- commencez par une lettre minuscule pour methodNames et variableNames, conformément à la convention de nommage CamelCase ;
- nommez toutes vos constantes sous la forme UPPERCASE_WITH_UNDERSCORES.

Utilisez toujours des accolades pour les instructions de bloc, même si elles sont vides ou sur une seule ligne. Cela améliore la lisibilité et évite les problèmes si ces blocs de code sont modifiés ultérieurement. De plus :

- utilisez des parenthèses pour clarifier l'ordre des opérations ;
- utilisez le polymorphisme pour réduire le besoin d'instructions switch ou d'opérations coûteuses, car cela peut entraîner des problèmes de performance si elles sont répétées dans une structure en boucle ;
- si vous vous servez des instructions switch, utilisez toujours une valeur par défaut : case et put break. Les instructions doivent être à la fin de chaque bloc, y compris la valeur par défaut.

Gestion des exceptions

L'une des principales préoccupations d'un développeur doit être de s'assurer que le logiciel ne plante jamais, même en cas d'imprévu. Au moment de l'exécution, de nombreux problèmes peuvent survenir, des données utilisateur non valides aux interruptions du réseau. C'est pour cette raison que tous les cas d'exception potentiels doivent être traités.

Au minimum, les exceptions d'exécution doivent toujours être consignées. Il est très rare qu'une exception ne se produise jamais et puisse être ignorée.

Capturez les exceptions dans un scope d'exception le plus précis possible. Ne wrappez pas un bloc *try* autour d'une section de code et capturez uniquement `java.lang.Throwable`. Certains cas exceptionnels sont plus faciles à capturer que d'autres. Il est préférable de ne pas regrouper les erreurs non récupérables (telles que `java.lang.OutOfMemoryError`) avec des exceptions attendues plus raisonnables (telles que `java.lang.NumberFormatException` lors de la conversion d'une chaîne en un entier).