**Convention de Nommage pour Java - pour Revue QA**

**A. Introduction**

Cette Convention de Nommage pour Java a été élaborée pour assurer la cohérence, la lisibilité et la maintenabilité du code au sein de notre équipe de développement. Elle vise à standardiser les noms des éléments du code pour optimiser la compréhension et réduire les erreurs potentielles.

**B. Principes Généraux**

1. **Clarté** : Les noms doivent décrire précisément le rôle de l'élément dans le programme.
2. **Consistance** : Appliquer uniformément les conventions à travers tout le projet.
3. **Lisibilité** : Les noms doivent être compréhensibles par toute l'équipe de développement.
4. **Langue** : Utiliser la langue qui convient le mieux au contexte et à l'équipe. Bien que l'anglais soit souvent préféré pour les noms des éléments du code dans un cadre international, il est permis d'utiliser le français dans des projets ou contextes où l'équipe est principalement francophone, en veillant toujours à maintenir la lisibilité et la cohérence.

**C. Règles de Nommage**

**C.1. Classes et Interfaces**

1. **Classes** : Utilisez le format PascalCase, en choisissant des noms représentatifs d'une substance ou d'un concept.  
   Exemples : GestionnaireEmploye, TraitementCommande
2. **Interfaces** : Utilisez PascalCase, sans le préfixe "I" traditionnel.  
   Exemples : RécupérateurDeDonnées, Persistable
3. **Classes Abstraites** : Utilisez PascalCase avec le préfixe "Abstract" recommandé.  
   Exemples : FormeAbstraite

**C.2. Méthodes**

1. **Méthodes d'instance** : Utilisez camelCase, en utilisant des verbes ou des phrases verbales pour décrire l'action réalisée.  
   Exemples : calculerSalaire(), validerEntrée()
2. **Méthodes booléennes** : Utilisez des préfixes comme "est", "a", ou "peut" pour indiquer un état ou une capacité.  
   Exemples : estActif(), aPermission()

**C.3. Variables et Champs**

1. **Variables d'instance** : Utilisez camelCase.  
   Exemples : prenom, nombreEmployes
2. **Constantes** : Utilisez MAJUSCULES\_AVEC\_UNDERSCORES.  
   Exemples : MAX\_UTILISATEURS
3. **Variables locales** : Utilisez camelCase, de manière descriptive et concise.  
   Exemples : montantTotal, nomUtilisateur

**C.4. Paramètres des Méthodes**

1. **Paramètres** : Utilisez camelCase, avec des noms clairs exprimant l'intention.  
   Exemples : processerCommande(String idCommande, int quantite)

**C.5. Paquets**

1. Utilisez des minuscules, séparées par des points.  
   Exemples : com.entreprise.produit.gestion

**C.6. Enumérations**

1. **Nom de l'enum** : Utilisez PascalCase.
2. **Constantes d'enum** : Utilisez MAJUSCULES.

Exemples : public enum Statut { EN\_ATTENTE, APPROUVÉ, REJETÉ }

**C.7. Types Génériques**

1. Utilisez une seule lettre majuscule pour les paramètres génériques.  
   Exemples : T pour type, E pour élément.

**C.8. Exceptions**

1. Utilisez PascalCase et ajoutez le suffixe "Exception".  
   Exemples : ExceptionEntréeInvalide

**C.9. Classes de Test**

1. Utilisez PascalCase et le suffixe "Test".  
   Exemples : GestionnaireEmployeTest

**D. Considérations Spéciales**

1. **Abréviations** : À éviter, sauf si couramment reconnues (ex. URL, HTTP).
2. **Collections** : Les noms des collections doivent être au pluriel.  
   Exemple : List<String> nomsEmployes
3. **Objets individuels** : Les noms des objets doivent être au singulier.  
   Exemple : String nomEmploye
4. **Nombres Magiques** : Les valeurs numériques doivent être définies comme des constantes nommées.  
   Exemple : public static final int MAX\_RETRIES = 5;

**E. Exemple de Code Conforme**

public class GestionnaireEmploye {

private static final int MAX\_EMPLOYES = 100;

private String prenom;

private int idEmploye;

public GestionnaireEmploye(String nom, int id) {

this.prenom = nom;

this.idEmploye = id;

}

public String getPrenom() {

return prenom;

}

public void setPrenom(String prenom) {

this.prenom = prenom;

}

public double calculerSalaire(double salaireDeBase) {

double prime = 0.1 \* salaireDeBase; // Prime de 10%

return salaireDeBase + prime;

}

public void afficherDetailsEmploye() {

System.out.println("Nom de l'employé : " + prenom);

System.out.println("ID de l'employé : " + idEmploye);

}

}

**F. Points d'Attention pour la Revue QA**

Lors de la revue du code, merci de porter une attention particulière aux éléments suivants :

1. **Cohérence des noms** : Vérifier que les noms des classes, méthodes, variables, et autres éléments respectent bien les conventions établies.
2. **Clarté des noms** : S'assurer que chaque nom est descriptif et qu’il ne comporte pas d’abréviations non standard.
3. **Conventions de casse** : Vérifier que PascalCase et camelCase sont appliqués correctement.
4. **Absence d'abréviations non reconnues** : Les abréviations doivent être évitées sauf si elles sont courantes et comprises de tous.
5. **Commentaires explicatifs** : Vérifier que le code complexe est correctement commenté pour faciliter la compréhension.
6. **Utilisation des constantes** : Vérifier que les valeurs magiques sont remplacées par des constantes nommées et respectent la convention de nommage des constantes.

**G. Conclusion**

Cette convention de nommage a pour objectif de garantir une uniformité dans le code et d’assurer qu'il soit facile à comprendre, à maintenir et à tester. En appliquant cette convention, nous favorisons la collaboration efficace entre les membres de l’équipe de développement, ce qui conduit à une meilleure qualité de code. Nous encourageons la stricte adhésion à cette convention afin de maintenir un code propre et de haute qualité.

**Auteur** : Stephanie Lee  
**Date** : le 12 décembre, 2024