

Les bonnes pratiques du développement

A. Git et Jira

a) Nommage des branches

- **feature**: Ajout d'une nouvelle fonctionnalité
- **bugfix**: Correction d'un bug
- **hotfix**: Correction d'un bug critique
- **chore**: Nettoyage du code
- **experiment**: Expérimentation de fonctionnalités

Pour créer une nouvelle branche pour une fonctionnalité :
feature/nom-de-la-fonctionnalité

bugfix/nom-du-bug/numero-du-tiket pour un bug

hotfix/nom-du-bug/numero-du-tiket pour un bug critique

b) Nommage des messages de commits

Le message du commit doit être précis et clair de sorte à comprendre facilement les modifications apportées au code.

Il doit contenir :

- Le type du commit :
 - **feat** pour ajouter une nouvelle fonctionnalité
 - **fix** pour la correction d'un bug
 - **build** pour un changement lié aux dépendances
 - **refactor** si le changement est lié au refactoring
 - **test** : ajout de nouveaux test unitaires
 - **revert** : annulation de commit
- Le scope : la partie de l'architecture impactée (Controller, donnée etc.)
- Le sujet : courte description du commit
- Le message du commit : détail du commit
- Le footer : référence au ticket Jira

Pour nous assurer de l'application de ces règles de commit, nous installerons un hook de git.

B. Développement

Notation des variables et attributs

Un nom de variable ou d'un attribut est un nom principal suffisamment éloquent. Le nom de la variable ne doit pas être un verbe

Utiliser le **lower camel case** : la première lettre de la chaîne en minuscule et la première lettre de chaque mot de la chaîne en majuscule

Exemple : prixViande pour indiquer le prix de la viande

Les constantes

Un nom de constante est un nom principal. La notation **Screaming snake case** est donc à utiliser pour nommer les constantes.

Cela consiste à mettre tous les mots de la constante en majuscule séparés par les tirets du bas (_).

Le nom de la constante ne doit pas commencer par un tiret (-).

Exemple de nommage : PRIX_VIANDE .

Utiliser les constantes en lieu et place des valeurs en dur dans le code.

Pas de *print(10)* mais *CONSTANTE = 10, print(CONSTANTE)*

Les fonctions et les méthodes

Le nom de la fonction est construit à l'aide d'un verbe et éventuellement un complément d'objet ou un adjectif qualificatif.

Le nom de la fonction ne doit pas être un nom principal.

On utilisera la notation camel case.

Exemple : estNull.

Une fonction doit exécuter une seule action logique et celle correspondant à son nom.

La taille de la fonction doit être limitée de préférence à 10 ou 15 lignes.

Les classes

Le nom de la classe est un nom principal qui représente un concept. Il doit être suffisamment éloquent.

On utilisera le pascal case.

Exemple : AnimalSauvage

Les fichiers

Les noms des fichiers sont des noms principaux suffisamment éloquent et qui synthétise leurs contenus.

On utilisera la notation camel case ou pascal case selon les circonstances.

Commentaire

Les commentaires sont écrits pour permettre aux autres développeurs de facilement comprendre le code.

Les variables, les constantes, les méthodes et les classes doivent être commentés dès lors qu'on ne peut pas comprendre intuitivement leur rôle ou leur fonctionnement.

Une attention particulière doit être accordée à la propreté du code.

Formatage

- L'indentation est obligatoire. Elle doit se faire avec l'aide de 4 espaces notamment après une accolade ouvrante.
- Les identifiants d'une expression sont séparés un espace *nombreVélos = 1*
- Les traitements sont séparés par une seule ligne

