

Nomenclature à respecter

Table des matières

Nomenclature à respecter	1
I) Structure générale du projet	2
II) Convention de nommage	2
III) Utilisation de .this	2
IV) Javadoc	2
V) Organisation des membres dans une classe	3
VI) MVC	3
a. Structure des fichiers	3
b. Nom des packages	4
c. Convention de nommage	4

I) Structure générale du projet

- Une **classe** par fichier
- Le fichier doit porter le même nom que la classe publique qu'il contient
- Organisation en packages
- TOUS LES NOMS EN ANGLAIS (plus facile à comprendre et plus simple pour les outils IA).

II) Convention de nommage

ELEMENT	STYLE	EXEMPLE
CLASSE/INTERFACE	UpperCamelCase	BankAccount, Book
METHODE	lowerCamelcase	calculateAccount()
VARIABLE /ATTRIBUT	lowerCamelCase	clientName
CONSTANTE (FINAL STATIC)	MAJUSCULE_UNDERSCORE	PI_VALUE
PACKAGE	Tout en miniscules	com.william.finance
PARAMETRE METHODE	lowerCamelCase	bookName

III) Utilisation de .this

Lorsqu'un nom de paramètre est identique à l'attribut, on utilise .this :

```
public void setName(String name) {
    this.name = name ; // nécessaire ici
}
```

IV) Javadoc

- Toujours mettre une Javadoc pour :
 - Les classes
 - Les méthodes publiques
 - Les interfaces
- Indiquer clairement ce que fait la méthode à l'aide des attributs @.....

V) Organisation des membres dans une classe

Ordre conseillé :

1. Constante (public static final)
2. Attributs (privés)
3. Constructeurs
4. Méthodes publiques
5. Méthodes privées (utilisaires)
6. Redéfinition (toString(),equals())

VI) MVC

a. Structure des fichiers

```
/MyProject
|
|— model/
|   └─ Account.java
|
|— view/
|   └─ AccountView.java
|
|— controller/
|   └─ AccountController.java
|
└─ Main.java
```

On n'oublie pas de mettre dans des packages, et on respecte les conditions vues précédemment.

b. Nom des packages

```
package myproject.model;  
package myproject.view;  
package myproject.controller;
```

Normalement , IntelliJ s'en charge tout seul.

c. Convention de nommage

Type	Convention	Exemple
<i>Model</i>	UpperCamel	Account, User
<i>View</i>	Finit par View	AccountView
<i>Controller</i>	Finit par Controller	AccountController