|  |  |  |
| --- | --- | --- |
| Cours : | **Labo Java base 2021-2022** | Rôle : |
| Sujet : | **Grille d’évaluation** | Formateur : | Yannick Boogaerts |

Pouvez-vous me montrer dans votre code un exemple d’utilisation des principes suivant ?

*Classificateur*

* Classe = Classe Main (qui permet de lancer le programme)
* Classe descendante = DataStore extends Serializable
* Classe abstraite (= pas dans le code)
* Interface = JavaNode
* Enum (= pas dans le code)
* Classe interne (= pas dans le code)
* Classe anonyme interne (= pas dans le code)
* Expression Lambda (= pas dans le code)

*Variables*

* Locales
* à une méthode = ListActivityType, l. 15
* à un bloc de code = HoraireAdd, l. 24-27
* Paramètres = ProjetFactory, l. 13 (String et Callable)
* Attribut d’instance = idem attributs de classe ?
* Attribut de classe = Stage, l. 13-16

*Opérations*

* Opération d’instance (= pas dans le code)
* Opération de classe = ProjetFactory, l. 35-41
* Constructeur = Stage, l. 20-25
* Overloading (= pas dans le code)
* Overridding = ListActivity, l. 27-33 / Item, l. 18-21 & l. 27-28
* Méthode virtuelle (= pas dans le code)
* Passage de paramètre = ProjetFactory, l. 13 (String et Callable) / l. 22 (JavaModel)
* Retour d’une valeur = ProjetFactory, l. 17

*Collection*

* List = JavaModel, l. 12
* Set (= pas dans le code)
* Map = Horaire, l. 17
* Iterator = Horaire, l. 17 (map) / JavaModel, l. 12 (list)
* Comparator = ActivityReplace, l. 30 et 32 (à moitié fait, equals)

*Exception*

* Try/catch/finally = Main, l. 46-53
* Throw = Main, l. 56-58
* Throws = Main, l. 12

*Conception Objet*

* Modèle = JavaModel, Stage, Activity, Horaire,
* Vue = JavaVue, ActivityVue, HoraireVue
* Controleur = JavaControler, HoraireAdd, HoraireRemove
* Factory = JavaFactory
* Injection de dépendance = ListActivityType, l. 25 / HoraireAdd, l. 19-21

Lister et situer dans votre code le plus d’exemples possibles d’utilisation de classes différentes de l’API Java.

Montrer un exemple de code dont vous êtes fier.