Notations UML

# **Classes :**

# Nom de la classe :

* Commence par une majuscule et chaque mot suivant commence également par une majuscule (Upper Camel Case) ;

# Attribut :

* Commence par une minuscule, chaque mot suivant commence par une majuscule (lower Camel Case) ;
* Visibilité généralement privée ;
* Le nom de l’attribut et son type sont séparés par « : » ;
* On peut définir un attribut en tant que constante avec <<const>>.

# Opération :

* Suivant le langage utilisé, Upper Camel Case pour C#, lower Camel Case pour Java ;
* Si besoin de paramètres, les mettre avec leur type entre parathèse
* Ne pas mettre de paramètres internes à la classe ;
* Le nom de l’opération et le type de ce qu’elle renvoi séparé par « : » ;
* Suivant le format suivant :

+ NomOperation(type parametre1, type parametre2, type parametre3) : type

* L’opération ToString() renvois une concaténation de tous les attributs ;
* Visibilité généralement publique.

# Constructeur :

* Initialise l’état de l’objet ;
* Visibilité généralement publique ;
* Le constructeur par défaut, donne des états par défaut ;
* Le constructeur classique, les valeurs des attributs et leurs types doivent être spécifiés ;
* Le constructeur par clonage, clone un objet déjà crée.

# **Objet :**

# Nom de l’objet :

* On nomme l’objet suivant ce format : « nomObjet : NomDeClasse » ;
* nomObjet en lower Camel Case, NomDeClasse en Upper Camel Case ;
* Le nom de l’objet et le nom de la classe sont séparés par « : » ;
* Le tout doit être souligné.

# Etat :

* Les noms d’états sont en lower Camel Case ;
* On sépare les états et leurs valeurs avec le symbole « = » ;
* On ne mentionne pas les types des états ;
* Les états suivent les règles de visibilités (généralement privés)

# Comportement :

* Il n’est pas d’usage d’indiquer les comportements, ils correspondent aux opérations de la classe ;
* Si indiqués ils suivent les mêmes règles que les opérations.

# **Généralités :**

* **Jamais d’espaces ni de caractères spéciaux dans les noms de classes, d’objets, attributs, d’états, d’opération ou de comportement.**
* Visibilité :
  + Publique, signalé par le symbole « + » en début de ligne, permet la visibilité, l’écriture ou l’utilisation par les autres classes ou objets ;
  + Protégé, signalé par le symbole « # » en début de ligne, permet la visibilité, l’écriture ou l’utilisation par uniquement les dérivés de la même classe ;
  + Privé, signalé par le symbole « - » en début de ligne, les attributs et opérations ne sont accessible que par leur propre classe ou objet.