

## A3 : Langage C/C++

### TDO 8 : Polymorphisme

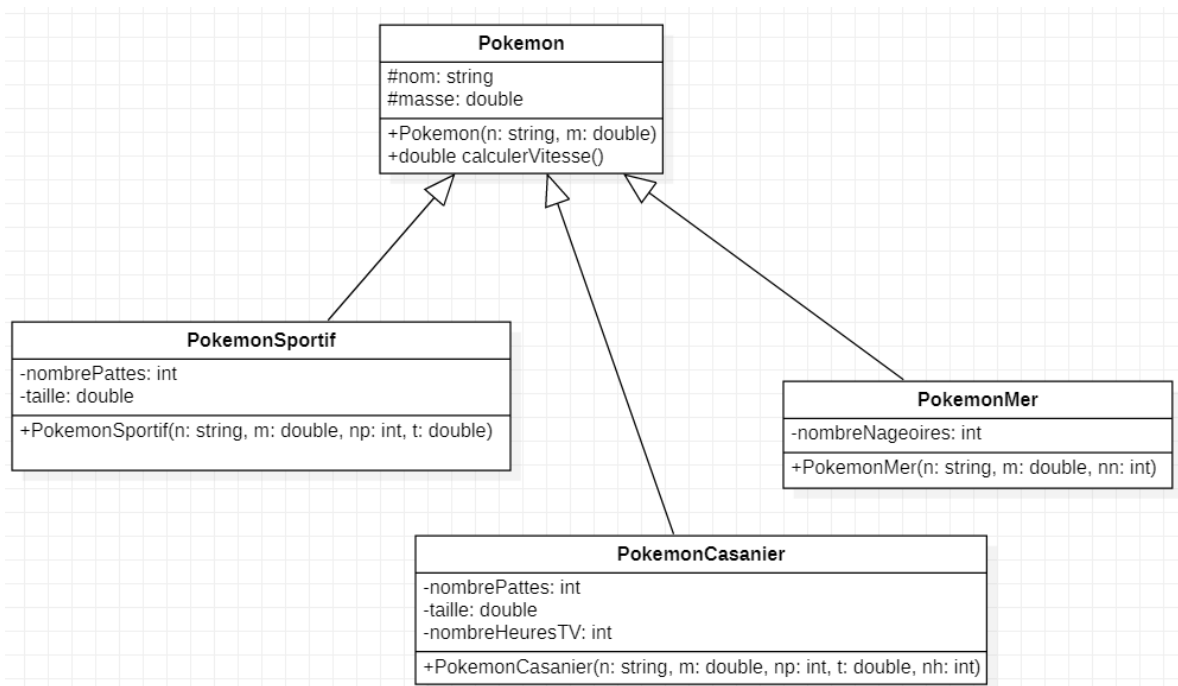
L'environnement de travail est Schooding pour la 1<sup>ère</sup> partie et VS Code pour la 2<sup>ème</sup> partie.

Extrait de <https://ufrsciencestech.u-bourgogne.fr/lpmi/UE5-POOetInternet/POO/Exercices/PokemonVehicules.pdf>

Les Pokémons sont des gentils animaux qui sont passionnés par la programmation objet en général et par le polymorphisme en particulier. Il existe 3 grandes catégories de Pokémons :

- 1) Les Pokémons sportifs : ces Pokémons sont caractérisés par un nom, une masse (en kg), un nombre de pattes, et une taille (en mètres). Ces Pokémons se déplacent sur la terre à une certaine vitesse que l'on peut calculer grâce à la formule suivante :  $\text{vitesse} = \text{nombre de pattes} * \text{taille} * 3$
- 2) Les Pokémons casaniers : ces Pokémons sont caractérisés par un nom, une masse (en kg), un nombre de pattes, une taille (en mètres) et le nombre d'heures par jour où ils regardent la télévision. Ces Pokémons se déplacent également sur la terre à une certaine vitesse que l'on peut calculer grâce à la formule suivante :  $\text{vitesse} = \text{nombre de pattes} * \text{taille} * 3$
- 3) Les Pokémons des mers : ces Pokémons sont caractérisés par un nom, une masse (en kg) et un nombre de nageoires. Ces Pokémons ne se déplacent que dans la mer à une vitesse que l'on peut calculer grâce à la formule suivante :  $\text{vitesse} = \text{masse} * \text{nombre de nageoires} / 25$ .

Il existe plusieurs solutions de modélisation pour décrire la relation d'héritage entre les Pokémons, et voici une :



## Partie 1 : Schooding

**Question 1 :** Complétez la classe Pokemon avec :

- ses attributs pour que les classes filles puissent les hériter;
- son constructeur;
- une fonction amie qui permet d'afficher les caractéristiques du Pokémon.

**Question 2 :** Complétez la classe PokemonSportif :

- qui est une sous classe de la classe Pokemon;
- avec ses propres attributs;
- avec son constructeur en appelant le constructeur de la classe mère;
- avec la surcharge de l'opérateur `operator<<` qui permet d'afficher les attributs d'un Pokémon sportif, en appelant d'abord la méthode d'affichage de la classe mère.

**Question 3 :** Complétez la méthode `calculerVitesse` de la classe Pokemon. Pourquoi elle pourrait être une méthode virtuelle pure ?

**Question 4 :** Redéfinissez la méthode `calculerVitesse` dans la classe fille PokemonSportif.

**Question 5 :** Déclarez un vecteur pouvant contenir une collection des Pokémon de catégories quelconques pour pouvoir ensuite insérer quelques Pokémon dans la collection.

**Question 6 :** Calculez la vitesse moyenne des Pokémon de la collection en utilisant le principe du polymorphisme, c'est-à-dire la capacité du système à choisir dynamiquement la méthode qui correspond au type de l'objet, au moment d'exécution du code.

## Partie 2 : VS Code

**Question 7 :** Implémentez tous ces classes, attributs et méthodes sur VS Code en organisant votre code de manière simple et optimale.

**Question 8 :** Testez votre code en créant plusieurs objets de différentes classes de Pokémon, avec et sans allocation dynamique, puis les ajoutez dans la collection `mesPokemons` pour calculez ensuite la vitesse moyenne.

**Question 9\* :** Calculez la vitesse moyenne des Pokémon **sportifs** de la collection `mesPokemons`.