

P00

Part 3 - Modélisation



Objectifs

Objectifs

- Expliquer l'intérêt d'UML
- Représenter un modèle simple



UML

UML

Représenter un système de manière standardisée et compréhensible par tous quelque soit leur :

- Langue
- Domaine métier
- Language utilisé

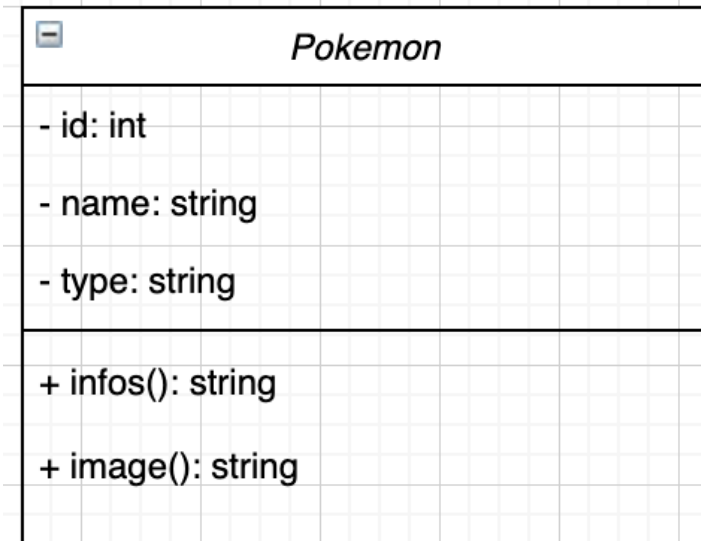
UML

Un pokemon en UML

Code

```
1 const IMG_BASE_URL: String =  
2   "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon";  
3  
4 class Pokemon {  
5   id: number;  
6   name: String;  
7   type: String;  
8  
9   constructor(id: number, name: String, type: String) {  
10     this.id = id  
11     this.name = name  
12     this.type = type  
13   }  
14  
15   infos() {  
16     return `${this.name} de type ${this.type}`;  
17   }  
18  
19   image() {  
20     return `${IMG_BASE_URL}/${this.id}.png`;  
21   }  
22 }
```

UML



UML

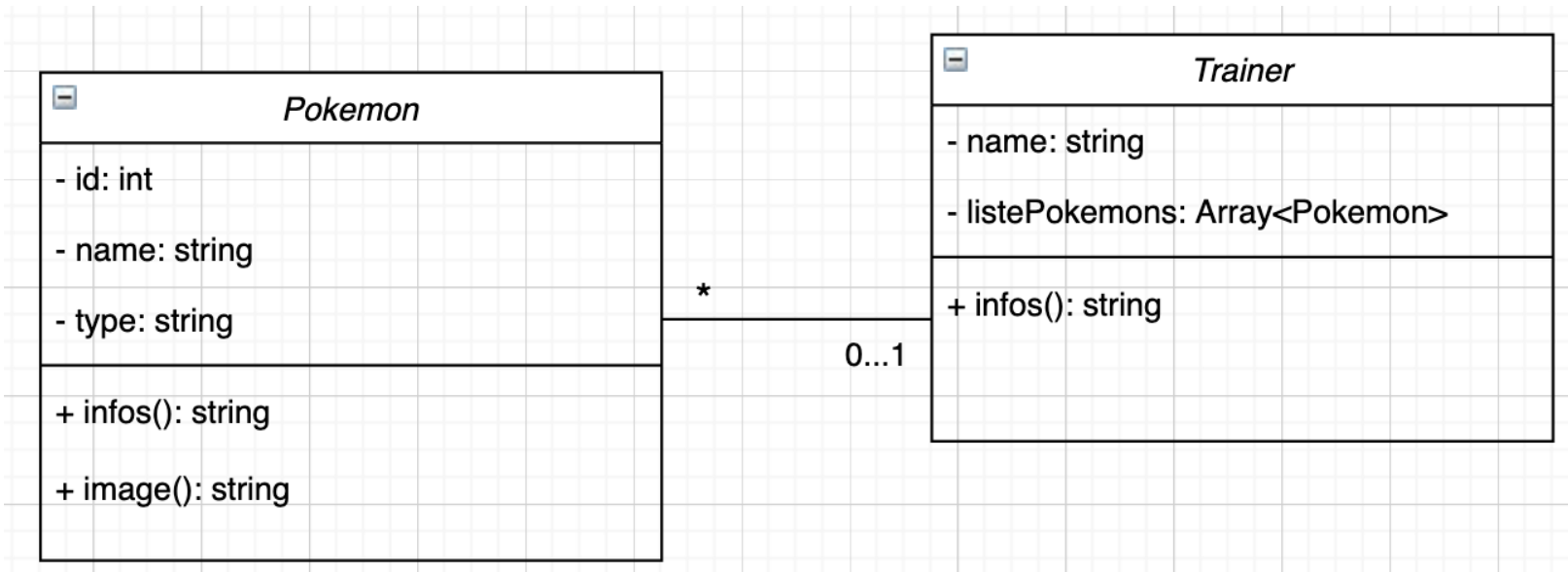
Amélioration

Créons une classe pour représenter un dresseur de pokémons

Qu'est ce qui caractérise un dresseur ?

UML

Représentation de la relation Pokemon / Dresseur



"*" Signifie de 0 à N Pokemons. Un Dresseur peut avoir 50 comme 0 Pokemon à entrainer

"0...1" Signifie qu'un Pokemon a 0 ou 1 dresseur.

UML

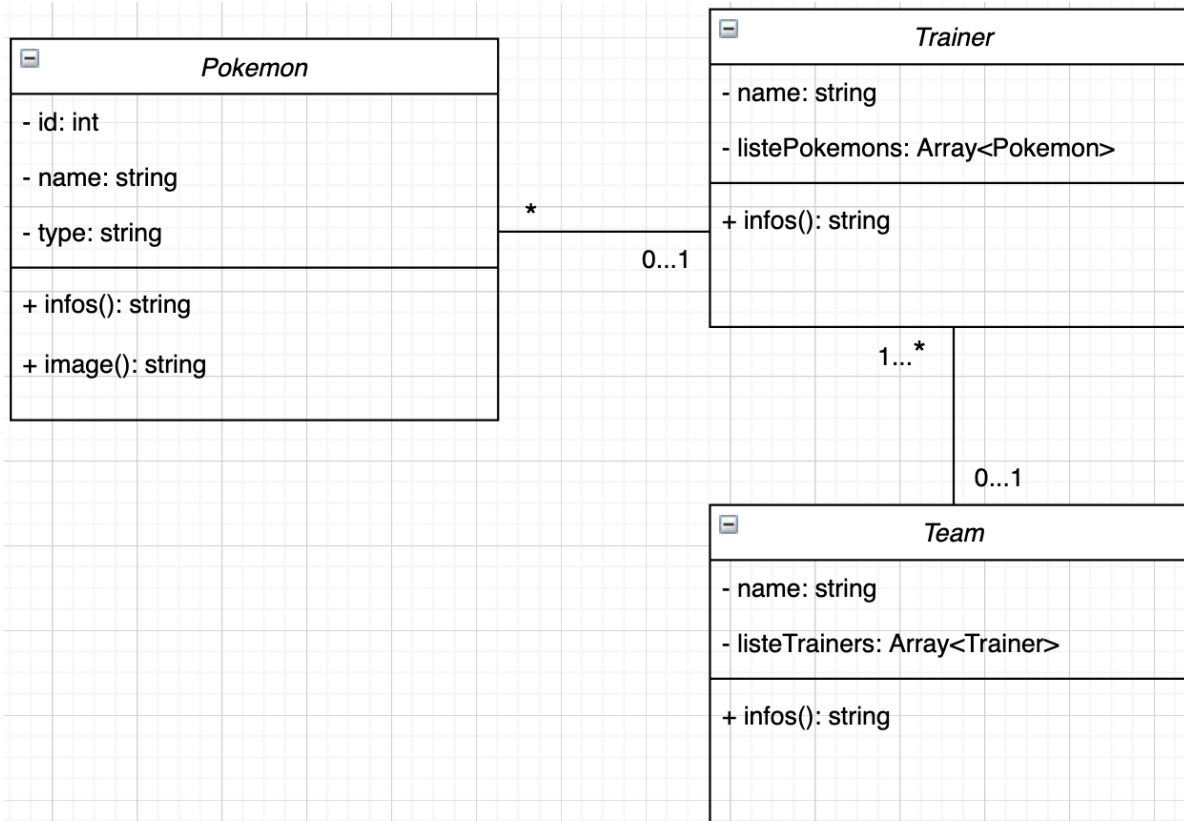
À vous de jouer

À partir du modèle UML existant, rajoutez la notion d'équipe.

- Une **Team** est composée de plusieurs **Trainers**
- Une **Team** a un nom
- Une **Team** a une méthode **info()** qui affiche la liste des **Trainers** et des **Pokemons**

UML

Correction



UML

Codons

- Faire ces trois classes
- Vous êtes un•e dresseu•r•se Pokémon avec au moins trois Pokémon de votre choix !
- La méthode **info()** dans **Trainer** retourne :

*Salut je suis **Nico** et j'ai **8** Pokemons*

UML

Et dans un vrai site ?



- Ajouter / retirer un produit
- Calculer le coût total
- Afficher le nombres de produits dans le panier