

## TP : Les objets métiers

CESi : RIL/DevOps



### Objectifs :

Mettre en œuvre la création de plusieurs objets métiers dans une architecture en couches.

En respectant le diagramme de classe, vous réaliserez les manipulations suivantes :

Pokemon	Dresseur	Attaque
f ◦ id int	f ◦ nom String	f ◦ nom String
f ◦ nom String	f ◦ prenom String	f ◦ force int
f ◦ poids float	m ↗ Dresseur(String, String)	m ↗ Attaque(String, int)
f ◦ taille float	m ↗ getNom() String	m ↗ getNom() String
f ◦ design String	m ↗ setNom(String) void	m ↗ setNom(String) void
f ◦ attaque01 Attaque	m ↗ getPrenom() String	m ↗ getForce() int
f ◦ attaque02 Attaque	m ↗ setPrenom(String) void	m ↗ setForce(int) void
f ◦ dresseur Dresseur		
m ↗ Pokemon(int, String, float, float, Attaque, Attaque, Dresseur)		
m ↗ getId() int		
m ↗ setId(int) void		
m ↗ getNom() String		
m ↗ setNom(String) void		
m ↗ getPoids() float		
m ↗ setPoids(int) void		
m ↗ getTaille() float		
m ↗ setTaille(int) void		
m ↗ getDesign() String		
m ↗ setDesign(String) void		
m ↗ getAttaque01() Attaque		
m ↗ setAttaque01(Attaque) void		
m ↗ getAttaque02() Attaque		
m ↗ setAttaque02(Attaque) void		
m ↗ getDresseur() Dresseur		
m ↗ setDresseur(Dresseur) void		

Powered by yFiles

- Créer une classe Pokémon
- Créer une classe Attaque
- Créer une classe Dresseur
- Lancer la classe boTest.java fournie par le formateur