

# TOB - TDs

22 Janvier, 2024

Louis Thevenet

## Table des matières

1. TD1 .....	2
2. TD2 .....	2

1. TD1

Exercice 1.1:

- 1. Point cartésien (Abscisse, Ordonnée)  
Point polaire (Rayon, Angle)
- 2. tout
- 3. format textuel

Exercice 1.2:

1.

Point		
Requête		
x : double	y : double	mod : double
arg : double	distance (Point p) : double	
Commande		
translater	set_x(x : double)	set_y(y : double)
set_mod(mod : double)	set_arg(arg : double)	afficher

Exercice 1.3:

1.

Point		
Requête		
x : double	y : double	mod : double
arg : double	distance (Point p) : double	
Commande		
translater	set_x(x : double)	set_y(y : double)
set_mod(mod : double)	set_arg(arg : double)	afficher
Point(x : double, y : double)		
Point(mod : double, arg : double)		

Exercice 1.4:

```
1 class Point {
2     double mod;
3     double arg;
4
5     double mod;
6     double arg;
7
8     // ...
9
10    set_x(double x) {
11        this.x = x;
12        this.mod = Math.sqrt(this.x * this.x + this.y * this.y);
13        this.arg = Math.atan2(this.y, this.x);
14    }
15 }
```

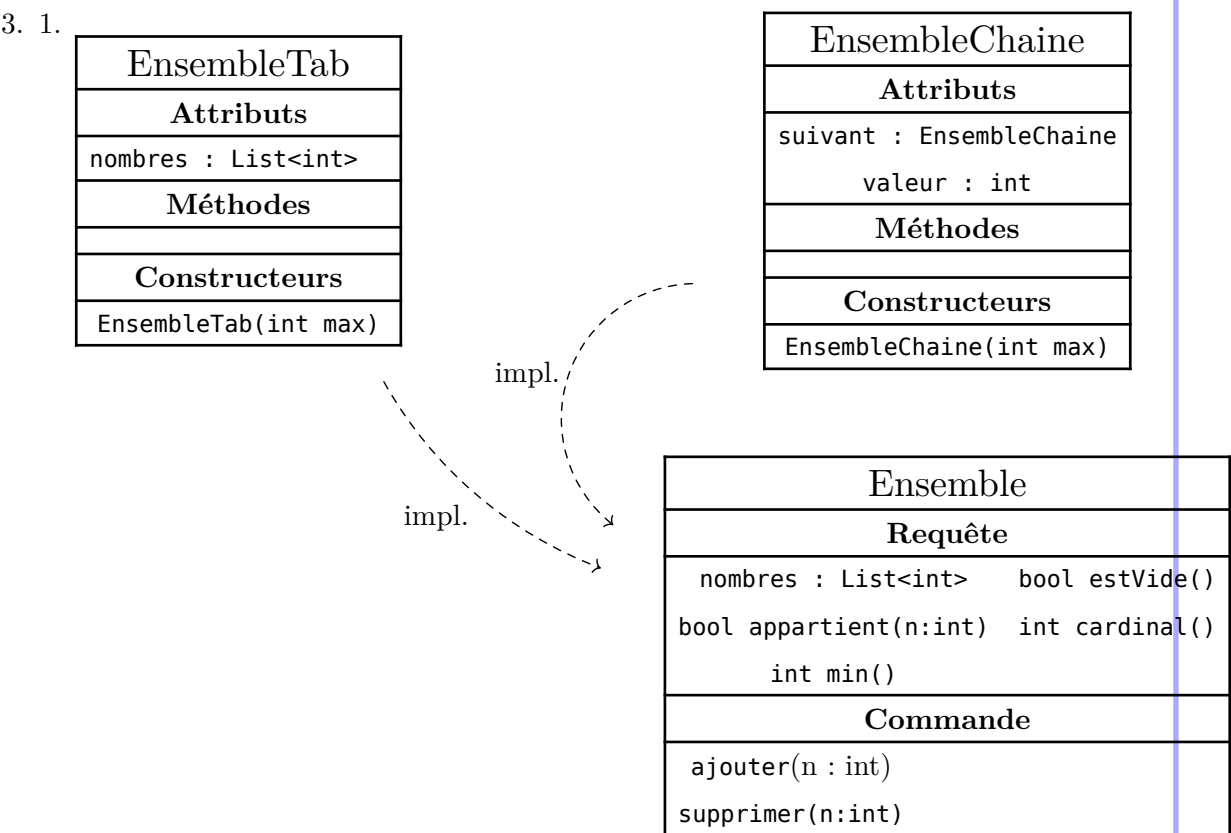
2. TD2

Exercice 2.1:

1.	<table><tr><th colspan="2">Ensemble</th></tr><tr><th colspan="2">Requête</th></tr><tr><td>nombres : List&lt;int&gt;</td><td>bool estVide()</td></tr><tr><td>bool appartient(n:int)</td><td>int cardinal()</td></tr><tr><td colspan="2">int min()</td></tr><tr><th colspan="2">Commande</th></tr><tr><td colspan="2">ajouter(n : int)</td></tr><tr><td colspan="2">supprimer(n:int)</td></tr></table>	Ensemble		Requête		nombres : List<int>	bool estVide()	bool appartient(n:int)	int cardinal()	int min()		Commande		ajouter(n : int)		supprimer(n:int)	
Ensemble																	
Requête																	
nombres : List<int>	bool estVide()																
bool appartient(n:int)	int cardinal()																
int min()																	
Commande																	
ajouter(n : int)																	
supprimer(n:int)																	

```
1 interface Ensemble {
2     boolean estVide();
3     boolean appartient(int n);
4     int cardinal();
5     int min();
6     void ajouter(int n);
7     void supprimer(int n);
8 }
```

```
1 Ensemble ensemble = new Ensemble(MAX);
2 Tantque !ensemble.estVide Faire
3     afficher(ensemble.min());
4     pour k de 1 à MAX Faire
5         ensemble.supprimer(ensemble.min() * k);
6     fin pour
7 Fin Tantque
```



- 2. On utilise implements
- 3. • Cas Tableau  
On ajoute au tableau
- Cas Liste Chainée  
On ajoute un maillon à la fin de la liste chaînée
- 4. // Cas Tableau  
int min() {  
 return this.nombres.get(0);  
}  
  
// Cas Liste Chainée  
int min() {  
 return this.valeur;  
}
- 5. Le cas List est plus efficace car le cas Liste Chainée est plus coûteux pour ajouter un élément.

Exercice 2.2:

```
public interface Ensemble<TypeDonnee> {
    boolean estVide();
    boolean appartient(TypeDonnee n);
    int cardinal();
    TypeDonnee min();
    void ajouter(TypeDonnee n);
    void supprimer(TypeDonnee n);
}
```