

TDA Ensemble

Sébastien Jean

IUT de Valence
Département Informatique

v1.0, 7 décembre 2025

TDA : *Ensemble*

- Type de données **linéaire** et **dynamique**, collection de valeurs (de type T) **sans doublon**
- Possibilité d'**ajouter** et de **retirer** un élément
- Possibilité d'**obtenir la taille** (nombre d'éléments), et de **savoir si un élément appartient à l'ensemble**

TDA : *Ensemble*

- Nom : **Ensemble (de T)**
- Dépendances : Entier, T (type des éléments), Booléen
- Opérations :
 - Constructeurs :
 - **ensemble_vider** : \rightarrow Ensemble
 - Transformateurs :
 - **ajouter** : $\text{Ensemble} \times T \rightarrow \text{Ensemble}$
 - **retirer** : $\text{Ensemble} \times T \rightarrow \text{Ensemble}$
 - Observateurs :
 - **taille** : $\text{Ensemble} \rightarrow \text{Entier}$
 - **appartient** : $\text{Ensemble} \times T \rightarrow \text{Booléen}$
- Pré-conditions : aucune

- **Axiomes**

- `taille(ensemble_vide()) = 0`
- `taille(ajouter(e, t)) = taille(e) + 1`
si `appartient(e, t) = FAUX`
- `taille(ajouter(e, t)) = taille(e)`
si `appartient(e, t) = VRAI`
- `taille(retirer(e, t)) = taille(e) - 1`
si `appartient(e, t) = VRAI`
- `taille(retirer(e, t)) = taille(e)`
si `appartient(e, t) = FAUX`

- Axiomes (suite)

- $\text{appartient}(\text{ensemble_vide}(), t) = \text{FAUX}$
- $\text{appartient}(\text{ajouter}(e, t1), t2) = \text{VRAI}$
si $t1 = t2$
- $\text{appartient}(\text{ajouter}(e, t1), t2) = \text{appartient}(e, t2)$
si $t1 \neq t2$
- $\text{appartient}(\text{retirer}(e, t1), t2) = \text{FAUX}$
si $t1 = t2$
- $\text{appartient}(\text{retirer}(e, t1), t2) = \text{appartient}(e, t2)$
si $t1 \neq t2$

Ensemble et pseudo code

- Dans notre *pseudo code*, on suppose que
 - Le **constructeur** est remplacé par la **déclaration de la variable**
 - Les **transformateurs** sont des **mutateurs**
 - Les **paramètres** sont **passés par copie**

```
VARIABLE e : Ensemble d'entiers
```

```
VARIABLE b : booléen
```

```
ajouter(e, 3)
```

```
ajouter(e, 6)
```

```
i ← appartient(e, 6)
```

```
retirer(2)
```

```
AFFICHER(taille(e))
```

Fin !

