

# TDA Ensemble

Sébastien Jean

IUT de Valence  
Département Informatique

v1.0, 7 décembre 2025

- 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\_vide : → Ensemble
  - **Transformateurs** :
    - ajouter : Ensemble × T → Ensemble
    - retirer : Ensemble × T → Ensemble
  - **Observateurs** :
    - taille : Ensemble → Entier
    - appartient : Ensemble × T → Booléen
- **Pré-conditions** : aucune

## ● Axiomes

- $\text{taille}(\text{ensemble_vide}()) = 0$
- $\text{taille}(\text{ajouter}(e, t)) = \text{taille}(e) + 1$   
si  $\text{appartient}(e, t) = \text{FAUX}$
- $\text{taille}(\text{ajouter}(e, t)) = \text{taille}(e)$   
si  $\text{appartient}(e, t) = \text{VRAI}$
- $\text{taille}(\text{retirer}(e, t)) = \text{taille}(e) - 1$   
si  $\text{appartient}(e, t) = \text{VRAI}$
- $\text{taille}(\text{retirer}(e, t)) = \text{taille}(e)$   
si  $\text{appartient}(e, t) = \text{FAUX}$

## • Axiomes (suite)

- $\text{appartient}(\text{ensemble_vide}(), t) = \text{FAUX}$
- $\text{appartient}(\text{ajouter}(e, t_1), t_2) = \text{VRAI}$   
si  $t_1 = t_2$
- $\text{appartient}(\text{ajouter}(e, t_1), t_2) = \text{appartient}(e, t_2)$   
si  $t_1 \neq t_2$
- $\text{appartient}(\text{retirer}(e, t_1), t_2) = \text{FAUX}$   
si  $t_1 = t_2$
- $\text{appartient}(\text{retirer}(e, t_1), t_2) = \text{appartient}(e, t_2)$   
si  $t_1 \neq t_2$

# 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 !

