Types Abstraits de Données et Collections

Algo – Chapitre 4

I. Définition d'un TAD

Un Type Abstrait de Données (TAD) est la **définition mathématique d'un type de données**. C'est un ensemble d'éléments et d'opérations associées (Ici, Ice, …) Il est défini par les éléments suivants :

Nom : *Nom du TAD*

Paramètre : *Pour les TAD collections, le nom générique du type des éléments stockés*

Utilise : *Types utilisés*

Opérations : $\text{nomOp}_1 : \quad Type_{e1} \times Type_{e2} \times \ldots \quad \rightarrow \quad Type_{s1} \times Type_{s2} \times \ldots$

⋮

Contient des opérations d'accès à l'information (constantes), sans entrée.

Axiomes/Sémantique : *Axiome : décrit logiquement une composition d'opérations*
Sémantique : explique ce que fait chaque opération

Préconditions : nomOp1 : *précondition de l'opération*

⋮

II. Etapes de création d'un TAD

- **Analyse :** Identifier et définir les TAD du programme
  - Complétude (pas d'op° manquantes)
  - Consistance (non contradictoire)
  - Identifiants significatifs
  - Préconditions
- **Conception préliminaire :** Déclarer les fonctions et procédures des opérations du TAD

Conception détaillée : Choisir une manière de représenter les entités du TAD, écrire algos

Exemple :

Nom : Réel

Utilise : Booléen

Opérations :

[0-9]⁺.[0-9]⁺ :		→	Réel
+ :	Réel × Réel	→	Réel
- :	Réel × Réel	→	Réel
* :	Réel × Réel	→	Réel
/ :	Réel × Réel	→	Réel
< :	Réel × Réel	→	Booléen
	⋮		

Axiomes : …

Préconditions : a / b : b ≠ 0