

M2103 – Structures de Données

Semestre 2 2015-2016

Cours 1

Introduction – Types Abstraits/Classes

Préambule

- Le langage C, étudié au S1, permet une programmation structurée, selon l'équation de Wirth :
Programme = Algorithmes + Structures de Données
- Cette approche a permis d'augmenter *exactitude* et *robustesse*
 - Exactitude : fonctionnement correspondant aux spécifications
 - Robustesse : comportement vis-à-vis des conditions 'anormales'
- ... Mais n'a pas permis d'atteindre les objectifs souhaités en termes d'évolutivité et de réutilisabilité
 - Evolutivité : facilité à modifier/ajouter des fonctionnalités
 - Réutilisabilité : possibilité de réutiliser des modules 'ailleurs'
- Origine du problème : le découplage entre structures de données et traitements qui oblige à « casser » un module dès que l'on remet en cause une structure de données.

Programme

- Approche «Type Abstrait »
 - Introduction et mise en oeuvre
 - Implémentation
 - Types Structurés
 - Classes
- Allocation Dynamique et Listes Chaînées
- Piles
- Files
- Dictionnaires

Enseignant

- Dr Mohammed Belkhatir
- Bureau 2^{ème} étage
- E-mail : mohammed.belkhatir@univ-lyon1.fr

Modalités

- Cours : notes + transparents (à connaître sur le bout des doigts)
- Exercices : sur place ou à emporter
- Evaluation
 - Interrogation(s) surprise
 - Examen

Plan du Cours

- **Importance des SDD**
- Type Abstrait : Présentation et Exemples
- Structures de Données Abstraites
- Illustration : Type Abstrait pour la Recherche d'Information
- Implémentation des Types Abstraits
 - Types Structurés
 - Programmation Orientée Objet

Importance des SDD

- Dans ce cours, on s'intéresse à :
 - La description de structures de données réutilisables dans différents contextes de résolution de problèmes
 - Aux traitements élémentaires permettant leur manipulation
- Essentiels pour la résolution de problèmes
 - ▣ Abstraction du problème
 - ▣ Algorithme(s)
 - ▣ **Structures de données**
 - ▣ Implémentation

Abstraction du Problème

- Identification du problème abstrait sans les détails non nécessaires
- Décision réalisée sur les attributs essentiels
- Exemple
 - Pour calculer la factorielle d'un entier positif n , je dois calculer...

Algorithmes

- Suite d'instructions pour résoudre le problème
- Chaque étape doit être *implémentable*.
- Correction
- Terminaison
- Exemple

factorielle (n) :

= 1, si $n = 0$

= $n * \text{factorielle}(n-1)$, sinon

Structures de Données

- Structures dont le but est de stocker des données manipulables par les algorithmes
- Indépendantes des langages de programmation
- Le but 'ultime' est de spécifier la nature des opérations sur les structures, pas la manière dont elles fonctionnent

Exemples de Structures de Données

- Pile
 - LIFO, récursivité, peut être implémentée par un tableau
- File
 - FIFO, peut aussi être implémentée par un tableau
- Liste chaînée
 - Allocation dynamique, insertion facilitée, accès séquentiel lent
- Arbre etc.

Plan du Cours

- Importance des SDD
- **Type Abstrait : Présentation et Exemples**
- Structures de Données Abstraites
- Illustration : Type Abstrait pour la Recherche d'Information
- Implémentation des Types Abstraits
 - Types Structurés
 - Programmation Orientée Objets

Type Abstrait

- Séparation entre “interface” (visible pour le monde extérieur) et implémentation (visible pour le programmeur)
- Consistent en
 - Domaine : ensemble de valeurs ou d’éléments permettant de mettre en œuvre le type abstrait
 - Opérations
 - Syntaxe (noms, valeurs qu’elles manipulent et retournent)
 - Fonctionnalité (implémentation)
- Données accessibles seulement au travers de l’interface définie (ou des opérations proposées)

Application de Géométrie

- **Objectif: fournir une boîte à outils de fonctions et procédures pour la manipulation de droites :**
 - Représenter une droite passant par 2 points distincts
 - Déterminer si 2 droites sont séquentes, parallèles ou confondues
 - Déterminer le point d'intersection de deux droites séquentes...

Exemple : TDroite

TypeAbs TDroite

Domaine

// coordonnées d'un point par lequel passe la droite

réel x,y

// l'angle que fait la droite avec l'axe des x

réel angle

Opérations

sontParallèles : TDroite x TDroite \rightarrow bool

sontConfondues : TDroite x TDroite \rightarrow bool

sontSéquentes : TDroite x TDroite \rightarrow bool

intersectionDroites : TDroite x TDroite \rightarrow réel x réel

(...)

FTypeAbs TDroite

Approche Type Abstrait

- Spécification des algorithmes fournis par la “boîte à outils” VS implémentation
- Le type *TDroite* est ensuite accessible pour les utilisateurs de la boîte à outils
 - L'accès se fait par l'intermédiaire des opérations
 - Pas de manipulation directe de la partie domaine

Changement Structure (1/2)

TypeAbs TDroite

Domaine

// la droite est le lieux des points (x,y)

// tels que $ax + by + c = 0$, avec a ou $b \neq 0$

réel a, b, c

(...)

FTypeAbs TDroite

Changement Structure (2/2)

- Réécriture des algorithmes
 - Implémentation plus lisible et efficace
 - Spécification inchangée
- Attention!
 - Ceci ne change rien pour l'utilisateur de l'application qui se contente de manipuler les TDroite avec les opérations de la boîte à outils.
 - Si son code accédait directement aux variables x, y et angle, il est évident qu'il ne fonctionnerait plus et devrait donc être modifié pour s'adapter aux modifications de la boîte à outils.

Principe de l'Approche «Type Abstrait »

- Identifier les opérations nécessaires pour manipuler ce type
 - spécifiées comme des procédures et fonctions dont un (ou plusieurs) paramètres a pour type le type abstrait.
 - L'ensemble de ces spécifications d'opération constitue la spécification du type abstrait.
- Le type abstrait est ensuite implémenté
 - Mise en oeuvre du type abstrait différente selon le langage de programmation choisi.
 - Les procédures et fonctions sont implémentées.
- L'utilisation du type abstrait devra se faire exclusivement en ayant recours aux opérations spécifiées

Autre Exemple: TVecteur

- Mettre en oeuvre une base
- Déterminer si un vecteur est nul
- Déterminer si deux vecteurs sont égaux, colinéaires
- Faire l'addition de deux vecteurs
- Multiplication d'un vecteur par un scalaire

Autre Exemple : TVecteur

TypeAbs TVecteur

Domaine

...

Opérations

base : TVecteur x TVecteur \rightarrow TVecteur x TVecteur

sontEgaux : TVecteur x TVecteur \rightarrow bool

estNul : TVecteur \rightarrow bool

sontColinéaires : TVecteur x TVecteur \rightarrow bool

addition : TVecteur x TVecteur \rightarrow TVecteur

multiplication : TVecteur x réel \rightarrow TVecteur

FTypeAbs TVecteur

Plan du Cours

- Importance des SDD
- Type Abstrait : Présentation et Exemples
- **Structures de Données Abstraites**
- Illustration : Type Abstrait pour la Recherche d'Information
- Implémentation des Types Abstraits
 - Types Structurés
 - Programmation Orientée Objets

Structures de Données Abstraites : une classification

- Organisation par position
 - Piles
 - Files
- Organisation par valeur
 - Dictionnaires

Opérations des structures de données abstraites

- Par position
 - Insertion de données à la position i de la structure
 - Effacement de données à la position i de la structure
 - Questions concernant les données à la position i de la structure
- Par valeur
 - Insertion de données avec valeur x dans la structure
 - Effacement de données avec valeur x dans la structure
 - Questions concernant les données avec valeur x

Plan du Cours

- Importance des SDD
- Type Abstrait : Présentation et Exemples
- Structures de Données Abstraites
- **Illustration : Type Abstrait pour la Recherche d'Information**
- Implémentation des Types Abstraits
 - Types Structurés
 - Programmation Orientée Objets

Type Abstrait pour la Recherche d'Information (1/3)

Annuaire

- Trouver le numéro de téléphone de Fred
- Supprimer l'adresse et le numéro de téléphone d'Alice
- Ajouter l'adresse et le numéro de téléphone d'Adam

Utilisation du nom comme une *clé de recherche*

Nom	Adresse	Numéro
Ahmed	Villeurbanne	0472221537
Fred	Part Dieu	0472123478
Nora	Perrache	0472139820
Jean	Vaulx	0472093456
Alice	Grenoble	0476290763
Adam	Croix Rousse	0472089562