

Corrigé type de l'examen : Paradigmes de Langages de Prog. MI-GLSD-M1-UEM213

Niveau : master 1 math. & .info option : GLSD

Durée : 1 heure

Date : 26 mai 2022

Les réponses sur les questions de cours (04 pts)

- **La spécification** est une définition purement mathématique pour un problème donné. par exemple : pour calculer $n!$ la spécification est comme suit :

$$0! = 1$$

$$n! = n \times (n-1)! \quad n > 0$$

(0.5 pt)

- **L'invariant** est une formule logique qui est vrai chaque appel récursif pour les arguments de cet appel, par exemple : pour calculer $n!$ la spécification par invariant est comme suit :

$$n! = i! * a$$

(0.5 pt)

- **Un Objet** est une manière d'organiser l'abstraction de données, il contient en lui-même la valeur et le jeu d'opérations.

(0.5 pt)

- **L'état** est une séquence de valeurs calculées progressivement, qui contiennent les résultats intermédiaires d'un calcul on peut dire aussi

Un état = un temps abstrait = une séquence de valeurs = une cellule



Création d'une cellule
avec contenu initiale 5



Changement du contenu
qui devient 6



FIGURE 1 – schéma de l'état

(0.5 pt)

Année universitaire : 2021/2022

- **L'abstraction de données** est une technique pour organiser les choses, elle a un intérieur, un extérieur et une interface entre les deux soutenu par le principe d'encapsulation.
 L'abstraction de données comprend : les objets et les Types Abstrait de Données.

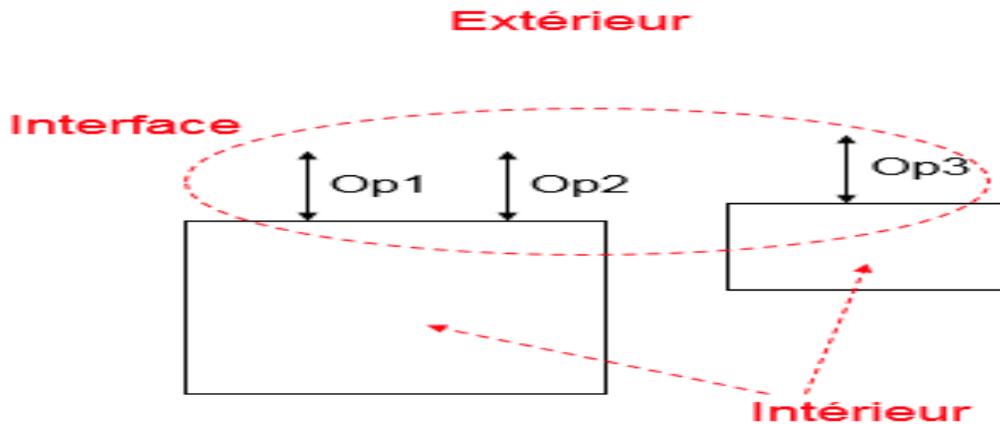


FIGURE 2 – schéma de l'abstraction de données

(0.5 pt)

- **La sémantique** : un modèle précis des opérations du langage de programmation pour la raison de vérifier l'exactitude d'un programme en utilisant une exécution de la machine abstraite
- (0.5 pt)
- **Héritage** : L'héritage est une manière de définir des abstractions de façon incrémentale mais on doit utiliser l'héritage le moins possible, parce que l'héritage de A peut être vue comme une autre interface à A ⇒ Une source supplémentaire d'erreurs.



FIGURE 3 – schéma de l'héritage

Année universitaire : 2021/2022

Pour résoudre ce problème on utilise le **principe de substitution** basé sur les liens statiques (**self**) et les liens dynamiques (**SuperClass**).

(0.5 pt)

— une opération est **polymorphe** si elle peut prendre des arguments de types différents, **Le polymorphisme** permet d'isoler des responsabilités dans les parties du programme qui les concernent, un exemple Le programme "guérir d'une maladie" est polymorphe :

- ◇ il marche avec toutes sortes de médecins
- ◇ Le médecin est un argument du programme
- ◇ Tous les mdceins comprennent le message "guérissez-moi"
- ◇ Le médecin fait ce qu'il faut selon sa spécialité

(0.5 pt)

Solution d'exercice 01 (06 pts)

Choisir la (les) bonne(s) réponse(s) dans ceux qui suivent :

- L'invariant utilise le principe des **vases communicants** qui vient de : (0.5 pt)

- (a) vider une variable, tout comme les vases. **augmenter une autre, c'est comme dans deux vases** (0.5 pt) (c) remplir des variables, tout comme des vases.
(b) **diminuer une variable et** **pt).**

- Toute utilisation d'une **fonction récursive** doit revenir dans un appel précédent pour exécuter quelque chose : (0.5 pt)

- (a) Vrai (b) **Faux (Tail recursion : voir la récursion terminale)** (0.5 pt).

- Parmi les concepts suivants, lesquels ne font pas partie de **la machine abstraite** d'un **paradigme fonctionnel**? (1.5 pt)

- (a) **Spécification** (0.5 pt). (c) Mémoire d'affectation unique (f) État d'exécution
(b) **Induction mathématique** (0.5 pt). (d) Environnement (g) **Sémantique axiomatique** (0.5 pt).
(e) Instruction sémantique

- Concernant **les objets** et **les objets fonctionnels**, choisissez quatre affirmations : (02 pts)

- (a) **Les objets sont mutables** (0.5 pt). (c) **Les objets combinent des valeurs et des opérations ensemble** (0.5 pt). pt).
(b) Les objets fonctionnels sont mutables (d) **Les objets fonctionnels combinent des valeurs et des opérations ensemble** (0.5 pt). (e) **Les objets sont populaires** (0.5 pt).
(f) Les objets fonctionnels sont populaires

Année universitaire : 2021/2022

- Choisissez les trois affirmations qui sont vraies concernant le paradigme orienté objets : (1.5 pt)

- | | |
|--|---|
| (a) La programmation orientée objet est toujours le bon paradigme. | (e) Le polymorphisme est un principe fondamental de la POO pour structurer les programmes (0.5 pt). |
| (b) C++ fournit correctement l'encapsulation | (f) La fonction est un principe fondamental de la POO pour structurer les programmes |
| (c) Java fournit correctement l'encapsulation (0.5 pt). | |
| (d) L'héritage est un principe de base de la POO | |

Solution d'exercice 02(06 pts)

Choisir la (les) bonne(s) réponse(s) dans ceux qui suivent :

- Si on donne la définition intuitive de Fibonacci par la suite : $fib(n) = fib(n-1) + fib(n-2)$, combien d'accumulateurs peut-on utiliser? (0.5 pt)

- | | |
|-------|---|
| (a) 0 | (c) 2 (Acc1 pour n-1 et Acc2 pour n-2)(0.5 pt). |
| (b) 1 | (d) 3 |

- Considérez le code suivant :

```

1 proc {TestPattern L}
2   case L
3   of M then {Browse 1}
4   [] H|T then {Browse 2}
5   [] nil then {Browse 3}
6   end
7 end
    
```

Qu'est-ce qui est imprimé si nous appelons $\{TestPattern\ nil\}$ et $\{TestPattern[ab]\}$ (01 pt) -

- | | | | |
|--|------------|------------|------------|
| (a) 1 et 1 (le <i>patern matching</i> dans ce programme est liée à M (0.5 pt)) | (b) 3 et 2 | (c) 2 et 2 | (d) 3 et 3 |
|--|------------|------------|------------|

- Considérez la règle de grammaire suivante qui définit une représentation d'un arbre binaire :

$$\langle tree\ T \rangle ::= leaf \mid tree(\ T\ left\ :< tree\ T \rangle\ right\ :< tree\ T \rangle)$$

Lesquels de ces arbres sont les cinq arbres binaires respectant la règle précédente? (4.5 pts)

- | | |
|---|---|
| (a) tree(42 leaf leaf leaf) | (f) tree(1 left :tree(2 left :leaf right :leaf right :tree(3 left :leaf right :leaf))(01 pt). |
| (b) tree(42 left :leaf right :leaf) (01 pt). | (g) tree(1 left :tree(2 left :tree(3 left :leaf right :leaf) right :leaf) right :leaf) (01 pt). |
| (c) tree(42 right :leaf left :leaf) (01 pt). | (h) leaf (0.5 pt). |
| (d) tree(1 tree(2 leaf) tree(3 leaf leaf leaf)) | |
| (e) tree(1 tree(2 leaf leaf) tree(3 leaf leaf)) | |

Solution d'exercice 03(04 pts)

Si on veut exprimer les critères de la divisibilité pour les nombres entiers 7 et 9.

- Pour la spécification de la divisibilité pour le nombre 9 : (0.5 pt)

9234 est divisible par 9 car

$$(9+2+3+4)=18=9 \times 2.$$

la spécification est donc :

$$s(n) = s(d_{k-1}d_{k-2} \dots d_i) + (d_{i-1} + d_{i-2} + \dots + d_0)$$

$$s(n) \text{ modulo } 9 = 0.$$

Année universitaire : 2021/2022

— 17 381 est divisible par 7 car

17381

$1738 + 5 \times 1 = 1743$,

$174 + 5 \times 3 = 189$,

$18 + 5 \times 9 = 63$ et

$6 + 5 \times 3 = 21 = 7 \times 3$.

On peut maintenant donner la spécification de la divisibilité pour le nombre 7 : **(0.5 pt)**

$$sF(n) = sF(d_{k-1}d_{k-2}\cdots d_i) + (d_{i-1} \times 5 + d_{i-2} \times 5 + \cdots + d_0 \times 5)$$

$$sF(n) \text{ modulo } 7 = 0.$$

— Les concepts à utiliser par un paradigme fonctionnel sont : **fonction et la récursion terminale (0.5 pt)**.

l'avantage : **la fonction ne jamais sera changé son comportement à cause de l'invariant (0.5 pt)**.

et l'inconvénient : **une changement inattendue vaut dire plus de changement sur les interfaces et le programme.(0.5 pt)**

— Les concepts à utiliser par un paradigme orienté objets sont : **les états explicite (temps) et les objets (ou exactement l'abstraction de données (0.5 pt)**

l'avantage : **les composants sont facilement changé en terme de programmation (0.5 pt)** et l'inconvénient : **le comportement se change par le temps.(0.5 pt)**