

Rattrapage TYLA — Typologie des Langages

EPITA – Apprentis promo 2010

Tous documents (notes de cours, photocopiés, livres) autorisés

Septembre 2008 (1h00)

Attention, dans ces questions il y a toujours une et une seule réponse valable. En particulier, lorsque plusieurs réponses sont possibles, prendre la plus restrictive. Par exemple s'il est demandé si 0 est *nul*, *non nul*, *positif*, ou *négatif*, sélectionner *nul* qui est plus restrictif que *positif* et *négatif*, tous deux vrais.

1. Le sucre syntaxique

Réponses possibles :

- a. permet d'ajouter la surcharge de fonctions dans un langage.
- b. est indispensable dans certains cas.
- c. est une notion qui n'existe que dans les langages à objet.
- d. est une notion qui relève purement de la syntaxe : on peut le toujours traiter au stade du parser dans un compilateur.

2. Dans un (des) compilateur(s), la présence d'une représentation intermédiaire du programme

Réponses possibles :

- a. est indispensable.
- b. est un langage inutile, car non exécutable.
- c. permet la factorisation de code dans certains cas.
- d. est dépendante du langage cible.

3. Dans un langage de programmation, les types

Réponses possibles :

- a. sont indispensables.
- b. sont forcément évalués à la compilation.
- c. sont forcément évalués à l'exécution.
- d. peuvent participer à l'optimisation des performances.

4. Citez une caractéristique essentielle de la programmation orientée objet.

Réponses possibles :

- a. Le masquage de données (en C++: `public`, `private`, `protected`).
- b. L'encapsulation (regroupement des données et des traitements).
- c. La surcharge d'opérateurs.
- d. L'absence de fonctions indépendantes (n'appartenant pas à une classe).

5. Qu'est-ce qui distingue fondamentalement une méthode (fonction membre) d'une fonction en C++?

Réponses possibles :

- a. Le fait que sa déclaration soit située dans celle de sa classe.
- b. La convention d'appel utilisant l'opérateur `.`.
- c. Son "premier" argument.
- d. On peut surcharger des méthodes, pas des fonctions.

6. Le type dynamique d'un objet

Réponses possibles :

- a. est un sur-type de son type statique.
- b. intervient lors de la résolution d'une fonction virtuelle.
- c. intervient lors de la résolution d'une fonction surchargée.
- d. est connu à la compilation.

7. Les multiméthodes sont disponibles dans

Réponses possibles :

- a. C++
- b. C#
- c. Common Lisp (CLOS)
- d. Java

8. Quel *design pattern* permet de garantir une unique instance pour un type donné ?

Réponses possibles :

- a. FABRIQUE ABSTRAITE
- b. POIDS PLUME
- c. PROTOTYPE
- d. SINGLETON

9. Lequel de ces design patterns n'est pas mis en œuvre dans l'implémentation des Abstract Syntax Trees (ASTs) de TC et de leur traitement ?

Réponses possibles :

- a. COMPOSITE
- b. PATRON DE MÉTHODE
- c. SINGLETON
- d. VISITEUR

10. En C

Réponses possibles :

- a. on peut avoir des pointeurs non initialisés.
- b. on peut avoir des références non initialisées.
- c. on dispose de new et delete.
- d. on dispose d'un garbage collector.

11. En C++, `std::list<int>` est

Réponses possibles :

- a. un type.
- b. un paramètre.
- c. une classe.
- d. un concept.

12. En C, quel est le type d'un pointeur constant vers un entier mutable ?

Réponses possibles :

- a. `const int *`
- b. `int const *`
- c. `int * const`
- d. `const int * const`

13. Parmi les propositions suivantes, laquelle est une propriété essentielle de `std::set` (en C++) ?

Réponses possibles :

- a. Ses éléments sont triés.
- b. C'est un conteneur à accès aléatoire.
- c. Ce conteneur ne dispose pas de `const_iterator`.
- d. La recherche d'un élément a une complexité temporelle en $O(n^2)$, où n est le nombre d'éléments.

14. La classe C++ `std::list<T>`

Réponses possibles :

- a. modèle (incarne) le concept Container.
- b. modèle le concept Random Access Container.
- c. modèle le concept Associative Container.
- d. ne modèle aucun de ces concepts.

15. La classe C++ `std::vector<T>`

Réponses possibles :

- a. modèle le concept Container.
- b. modèle le concept Random Access Container.
- c. modèle le concept Associative Container.
- d. ne modèle aucun de ces concepts.

16. La classe C++ `std::set<T>`

Réponses possibles :

- a. modèle le concept Container.
- b. modèle le concept Random Access Container.
- c. modèle le concept Associative Container.
- d. ne modèle aucun de ces concepts.

17. La classe C++ `std::map<T>`

Réponses possibles :

- a. modèle le concept Container.
- b. modèle le concept Random Access Container.
- c. modèle le concept Associative Container.
- d. ne modèle aucun de ces concepts.

18. Qu'est ce qui rend les unions difficiles à utiliser en C et en C++?

Réponses possibles :

- a. On ne peut pas les placer sur la pile.
- b. On ne peut y mettre aucune struct.
- c. On ne sait jamais exactement quelle place elles occuperont en mémoire.
- d. L'identification du type de données effectivement stocké est à la charge de l'utilisateur.

19. Pour laquelle de ces tâches le préprocesseur du C++ est-il réellement utile ?

Réponses possibles :

- a. La compilation séparée.
- b. La définition de constantes.
- c. La mise en ligne de définitions de fonctions.
- d. La création de code générique à l'aide de macros.

20. Le programme Tiger 42 + 51

Réponses possibles :

- a. provoque une erreur à la compilation.
- b. provoque une erreur à l'exécution.
- c. renvoie un code de sortie (exit status) valant 0 à l'exécution.
- d. renvoie un code de sortie (exit status) valant 93 à l'exécution.

21. En Tiger, les tableaux sont construits

Réponses possibles :

- a. sans être initialisés.
- b. sur la pile.
- c. sur le tas.
- d. dans le segment de données.

22. Dans TC, un environnement ou table de symboles

Réponses possibles :

- a. est présent dans la pile Tiger.
- b. est relatif à une portée donnée.
- c. est une structure de données nécessaire à l'exécution d'un programme.
- d. contient une copie des registres.

23. En Tiger, comment la mémoire alloué dynamiquement est-elle libérée ?

Réponses possibles :

- a. À l'aide du mot clef delete.
- b. À l'aide de la routine de la bibliothèque standard dispose.
- c. Cette tâche est dévolue à l'environnement d'exécution du programme.
- d. Il n'y a jamais d'allocation dynamique en Tiger.

24. Quel type à l'expression `foo := 42` en Tiger ?

Réponses possibles :

- a. Bool
- b. Int
- c. Nil
- d. Void

25. Dans tous les langages fonctionnels

Réponses possibles :

- a. l'affectation est proscrite.
- b. il n'y a pas de méthode (fonctions membre).
- c. les expressions sont évaluées paresseusement.
- d. on peut manipuler une fonction comme n'importe quel autre variable.

26. Quelle fonctionnalité de Java est assez dépendante du mode d'exécution à base de machine virtuelle ?

Réponses possibles :

- a. L'introspection.
- b. Les types génériques.
- c. L'invocation de méthodes à distance ou Remote Method Invocation (RMI).
- d. L'absence de fonctions indépendantes (n'appartenant pas à une classe).

27. Qu'est-ce qui distingue les références Java de celles de C++?

Réponses possibles :

- a. Les références Java sont toujours valides.
- b. Les références Java peuvent être invalides.
- c. Les références C++ peuvent être réutilisées pour référencer plusieurs objets au cours de leur durée de vie.
- d. Les références C++ ne permettent pas de référencer des objets sur la pile.

28. L'inventeur de la technique de parsing LR est

Réponses possibles :

- a. John Backus.
- b. Noam Chomsky.
- c. Donald Knuth.
- d. Lars Rasmussen.

29. Quel est le premier "vrai" langage de programmation haut niveau ayant eu du succès ?

Réponses possibles :

- a. COBOL
- b. FORTRAN
- c. Lisp
- d. PL/I

30. Quel fameux langage de programmation nécessitait un clavier particulier pour en écrire les symboles ?

Réponses possibles :

- a. ALGOL
- b. APL
- c. FP
- d. Simula