

TYLA — Typologie des Langages

EPITA – Apprentis promo 2010
Tous documents (notes de cours, photocopiés, livres) autorisés

Février 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. Les multiméthodes permettent

Réponses possibles :

- a. aux méthodes de retourner plusieurs résultats.
- b. le polymorphisme dynamique sur plusieurs arguments de fonctions.
- c. aux classes d'avoir plusieurs méthodes de même nom.
- d. différents paradigmes de programmation.

2. Le patron de conception « Visitor » permet l'utilisation

Réponses possibles :

- a. d'itérateurs en profondeur d'abord.
- b. d'itérateurs en largeur d'abord.
- c. des multiméthodes dans un langage objet qui en est démuné.
- d. d'accesseurs sur des membres pourtant privés.

3. Les templates de classe du C++

Réponses possibles :

- a. sont des collections de templates de fonctions libres.
- b. sont des générateurs de classes.
- c. sont des classes dont toutes les méthodes sont virtuelles.
- d. sont des classes dont toutes les méthodes sont virtuelles pures.

4. Un design pattern est

Réponses possibles :

- a. un langage de conception universel.
- b. une bonne solution à un problème connu.
- c. une méthode de conception d'application orientée objet.
- d. un générateur de classes de conception.

5. La liaison dynamique en C++

Réponses possibles :

- a. fait référence aux bibliothèques dynamiques.
- b. a rapport avec la surcharge d'opérateurs.
- c. a rapport avec « virtual ».
- d. repose sur « template ».

6. La résolution des appels « virtual » nécessite

Réponses possibles :

- a. la connaissance du type des contenants.
- b. la connaissance du type des classes.
- c. la connaissance du type des opérateurs.
- d. la connaissance du type des contenus.

7. La résolution de la surcharge nécessite

Réponses possibles :

- a. la connaissance du type des contenants.
- b. la connaissance du type des classes.
- c. la connaissance du type des opérateurs.
- d. la connaissance du type des contenus.

8. Les visiteurs

Réponses possibles :

- a. permettent de parcourir de façon générique les conteneurs.
- b. sont des fonctions objets.
- c. permettent d'implémenter le « dispatching » une fois pour toute.
- d. remplacent les accesseurs.

9. Les multiméthodes sont disponibles dans

Réponses possibles :

- a. C++
- b. CLOS
- c. Tiger
- d. Haskell

10. Surcharge vs méthodes virtuelles: quelle est la bonne réponse ?

Réponses possibles :

- a. La surcharge et les méthodes virtuelles sont des mécanismes statiques.
- b. La surcharge et les méthodes virtuelles sont des mécanismes dynamiques.
- c. La surcharge est un mécanisme statique, les méthodes virtuelles un mécanisme dynamique.
- d. La surcharge est un mécanisme dynamique, les méthodes virtuelles un mécanisme statique.

11. Le support des fonctions récursives nécessite

Réponses possibles :

- a. une pile (*stack*).
- b. un tas (*heap*).
- c. que le langage propose des pré-déclarations (*forward declarations*).
- d. la liaison des fonctions dynamiques.

12. Généralement, lorsque l'on définit un constructeur non trivial en C++ dans une classe qui possède des attributs de type pointeurs, on écrit systématiquement aussi

Réponses possibles :

- a. un accesseur.
- b. un propulseur.
- c. un `operator=`.
- d. un pretty-printer.

13. En C++, l'encapsulation est réalisée grâce

Réponses possibles :

- a. aux mots-clef `public`, `private` et `protected`.
- b. à la séparation du code en fichiers d'interface (`foo.hh`) et d'implémentation (`foo.cc`).
- c. à `#include`.
- d. aux classes.

14. Dans l'absolu, les relations entre les modules d'un logiciel devraient former

Réponses possibles :

- a. un graphe quelconque.
- b. un graphe planaire.
- c. un graphe non orienté.
- d. un graphe orienté acyclique.

15. Lequel de ces éléments n'entre pas en compte lors de la résolution d'une méthode surchargée en C++?

Réponses possibles :

- a. le nom de la fonction.
- b. les arguments de la fonction.
- c. le type de retour.
- d. le qualificateur const de la méthode.

16. En C

Réponses possibles :

- a. on ne dispose pas de malloc et free.
- b. on dispose d'un garbage collector.
- c. on peut avoir des pointeurs non initialisés.
- d. on peut avoir des références non initialisées.

17. En C++

Réponses possibles :

- a. on ne dispose pas de malloc et free.
- b. on dispose d'un garbage collector.
- c. on peut avoir des pointeurs non initialisés.
- d. on peut avoir des références non initialisées.

18. En Java

Réponses possibles :

- a. on dispose de malloc et free.
- b. on ne dispose pas d'un garbage collector.
- c. on peut avoir des pointeurs non initialisés.
- d. on peut avoir des références non initialisées.

19. En C++, on appelle objet-fonction

Réponses possibles :

- a. un objet construit à l'intérieur d'une fonction.
- b. un objet disposant d'un `operator()`.
- c. une méthode.
- d. un fichier de code compilé (`foo.o`) ne contenant qu'une seule fonction (ex: `foo ()`).

20. On dit d'un langage qu'il est fonctionnel

Réponses possibles :

- a. s'il supporte le concept de fonction, éventuellement récursive.
- b. lorsqu'il permet de manipuler des fonctions comme n'importe quel autre entité/objet.
- c. lorsqu'il dispose d'un compilateur implémenté et en état de marche.
- d. s'il n'effectue aucun effet de bord.

21. Qui est l'auteur du langage C?

Réponses possibles :

- a. Brian Kernighan
- b. Dennis Ritchie
- c. Bjarne Stroustrup
- d. Ken Thompson

22. De nos jours, les programmes C++

Réponses possibles :

- a. sont interprétés.
- b. sont compilés vers un équivalent en C, puis vers le langage d'assemblage de la machine.
- c. sont compilés nativement vers le langage d'assemblage de la machine.
- d. sont compilés vers un code-octet (*byte-code*).

23. Le type dynamique d'un objet

Réponses possibles :

- a. est un sous-type de son type statique.
- b. est un sur-type de son type statique.
- c. est connu à la compilation .
- d. est utilisé pour résoudre les appels de fonctions/méthodes. surchargées

24. Lequel de ces opérateur ne peut être surchargé en C++?

Réponses possibles :

- a. ,
- b. %
- c. @
- d. |

25. BNF

Réponses possibles :

- a. est un « langage de langages ».
- b. est une « grammaire de grammaires ».
- c. est un « langage de grammaires ».
- d. est une « grammaire de langages ».

26. Lequel de ces langages n'est pas normalisé ?

Réponses possibles :

- a. C
- b. C++
- c. C#
- d. D

27. On dit qu'un langage de programmation dispose d'un « typage statique fort » lorsque

Réponses possibles :

- a. la liaison des noms est effectuée à la compilation.
- b. tous les types sont vérifiés à la compilation.
- c. tous les types sont vérifiés à l'exécution.
- d. ce langage est dépourvu du polymorphisme d'inclusion.

28. Parmi les assertions suivantes, laquelle est fausse ?

Réponses possibles :

- a. Il est possible d'écrire des macros récursives en C.
- b. Il est possible d'écrire quelques conteneurs génériques en C grâce aux macros.
- c. Les constantes numériques définies à l'aide de macros C sont portables.
- d. Les macros utilisées comme fonctions peuvent produire des effets de bord.

29. Une fonction C++ ne peut être mise en ligne (*inlined*) si

Réponses possibles :

- a. elle utilise des variables globales.
- b. elle est récursive.
- c. elle fait usage de `new`.
- d. elle renvoie une valeur (son type de retour est différent de `void`).

30. Laquelle de ces déclarations Tiger est invalide ?

Réponses possibles :

- a. `var a : int`
- b. `var a := 42`
- c. `var a : int := 42`
- d. aucune

31. Le programme Tiger (`print_int ((4;2) + (5;1)); print ("\n")`)

Réponses possibles :

- a. ne compile pas.
- b. affiche 3 à l'exécution.
- c. affiche 9 à l'exécution.
- d. affiche 9;3 à l'exécution.

32. Quelle structure de données ne peut-on construire en Tiger ?

Réponses possibles :

- a. un tableau dynamique de chaînes de caractères.
- b. un tableau statique de pointeurs.
- c. une liste chaînée.
- d. un dictionnaire (*map*).

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

Réponses possibles :

- a. C'est un conteneur à accès aléatoire.
- b. La recherche d'un élément a une complexité temporelle en $O(1)$.
- c. Un élément `y` est présent au plus une fois.
- d. C'est le seul conteneur de la bibliothèque standard du C++ capable de contenir d'autres conteneurs.

34. Quel polymorphisme est supporté par le langage C?

Réponses possibles :

- a. le polymorphisme d'inclusion
- b. le polymorphisme de coercition
- c. le polymorphisme paramétrique
- d. aucun

35. En C++, `std::list` est

Réponses possibles :

- a. un type.
- b. un paramètre.
- c. un identifiant.
- d. un concept.

36. Quel est la signature du constructeur par copie d'une classe C++ `Foo` ?

Réponses possibles :

- a. `Foo::Foo (Foo&) const`
- b. `Foo::Foo (const Foo)`
- c. `Foo::Foo (const Foo&)`
- d. `Foo::Foo (const Foo&) const`

37. En C++, un paramètre effectif d'une classe (paramétrée, donc) ne peut pas valoir

Réponses possibles :

- a. `const`.
- b. `unsigned`.
- c. une constante entière.
- d. un type de classe défini par l'utilisateur.

38. Tiger supporte

Réponses possibles :

- a. les fonctions imbriquées.
- b. les fonctions d'ordre supérieur (manipulables comme n'importe quelle autre valeur).
- c. les fonctions paramétrées.
- d. les classes paramétrés.

39. Les concepts du C++ ISO 2003

Réponses possibles :

- a. se définissent grâce au mot clef `concept`.
- b. sont vérifiés implicitement par le compilateur.
- c. sont compilés automatiquement aux sites d'utilisations.
- d. sont des types qui supportent la redéfinition.

40. En C++, `int main () {}`

Réponses possibles :

- a. ne compile pas.
- b. produit une erreur à l'exécution.
- c. produit un programme dont le code de retour est 0.
- d. produit un programme dont le code de retour est dépendant de l'implémentation du compilateur utilisé.