

TYLA — Typologie des Langages

EPITA_APPING1_2013_S2_TYLA

EPITA – Apprentis promo 2013 – **Sans documents ni machine**

Juin 2011 (1h00)

Répondre sur les formulaires de QCM ; aucune réponse manuscrite ne sera corrigée. Renseigner les champs d'identité. Bien lire les questions, chaque mot est important. Il y a exactement une et une seule réponse juste pour ces questions. Si plusieurs réponses sont valides, sélectionner la plus restrictive. Par exemple s'il est demandé si 0 est *nul*, *non nul*, *positif*, ou *négatif*, cocher *nul* qui est plus restrictif que *positif* et *négatif*, tous deux vrais.

1 Programmation orientée objet

Q.1 Le type dynamique d'un objet

- 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 distinguer des fonctions/méthodes surchargées.

Q.2 Dans quel langage les appels de méthodes ne sont pas vérifiés statiquement ?

- a. C++
- b. Java
- c. Simula
- d. Smalltalk

Q.3 Qu'appelle-t-on une métaclasse en Smalltalk ?

- a. Une classe abstraite.
- b. Une classe qui hérite d'elle-même.
- c. Une classe ayant des méta-méthodes.
- d. Une classe dont les instances sont des classes.

Q.4 Les multiméthodes permettent

- a. aux méthodes de retourner plusieurs résultats.
- b. le polymorphisme dynamique sur plusieurs arguments de fonctions.
- c. à une classe d'avoir des méthodes portant le même nom.
- d. d'avoir des méthodes polymorphes (virtuelles) dans une hiérarchie de classe utilisant l'héritage multiple.

2 Programmation générique

- Q.5 Les templates de classes du C++ sont
- des collections de templates de fonctions libres.
 - des générateurs de classes.
 - des classes dont toutes les méthodes sont virtuelles.
 - des classes dont toutes les méthodes sont virtuelles pures.
- Q.6 Parmi les termes suivants, lequel ne peut pas être utilisé comme paramètre effectif d'une classe paramétrée ?
- const.
 - Une constante entière.
 - unsigned.
 - Un type classe défini par l'utilisateur.
- Q.7 Parmi les lignes C++ suivantes, laquelle est invalide ?
- `std::pair p1 (42, 51);`
 - `std::pair<float, int> p2 (42, 3.14f);`
 - `std::pair<int, float> p3 = std::make_pair (42, 3.14f);`
 - `std::pair<int, float> p4 = std::make_pair<int, float> (2.72f, 51);`
- Q.8 Les concepts du C++ ISO 2003
- se définissent grâce au mot clef `concept`.
 - sont compilés automatiquement aux sites d'utilisations.
 - sont vérifiés explicitement par le compilateur.
 - expriment des contraintes sur les paramètres de templates.

3 Programmation fonctionnelle

- Q.9 On dit d'un langage qu'il est fonctionnel
- s'il n'effectue aucun effet de bord.
 - lorsqu'il permet de manipuler des fonctions comme n'importe quel autre entité/objet.
 - s'il supporte le concept de fonction récursive.
 - lorsqu'il dispose d'un compilateur implémenté et en état de marche.
- Q.10 Un langage fonctionnel est dit pur lorsque
- il proscrit tout effet de bord.
 - il ne contient aucune construction orientée objet.
 - ses fonctions ont au plus un argument.
 - ses expressions sont évaluées paresseusement.
- Q.11 En C++, on appelle objet-fonction
- un objet construit à l'intérieur d'une fonction.
 - un objet disposant d'un `operator()`.
 - une méthode.
 - un fichier de code compilé ('foo.o') ne contenant qu'une seule fonction (`foo()`).

Q.12 On appelle fermeture

- a. une fonction qui n'est pas récursive.
- b. une fonction qui capture des références à des variables libres dans l'environnement lexical.
- c. une fonction qui a été mise en ligne (*inlined*).
- d. une fonction passée en argument à une autre fonction.

4 C++

Q.13 La liaison dynamique en C++

- a. a un rapport avec `virtual`.
- b. est liée à la surcharge des opérateurs.
- c. repose sur `template`.
- d. s'appuie sur `dynamic_cast`.

Q.14 Surcharge vs méthodes virtuelles : quelle est la bonne réponse ?

- a. La surcharge et les méthodes virtuelles sont des mécanismes dynamiques.
- b. La surcharge et les méthodes virtuelles sont des mécanismes statiques.
- 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.

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

- a. les arguments de l'appel.
- b. le nom de la fonction.
- c. le qualificatif `const` de la méthode.
- d. le type de retour.

Q.16 Une fonction C++ ne peut être mise en ligne (*inlined*) si

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

Q.17 En C++, `std::list` est

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

5 Langages de programmation

Q.18 Qui est l'auteur du langage C ?

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

- Q.19 Lequel de ces langages n'a pas été influencé par Simula ?
- a. Algol
 - b. C++
 - c. Objective C
 - d. Smalltalk
- Q.20 Qui est l'inventeur de la souris ?
- a. Douglas Engelbart
 - b. Donald Knuth
 - c. Gordon Moore
 - d. Konrad Zuse
- Q.21 Que signifie « BNF » ?
- a. Backus-Naur Form
 - b. BASIC Numbering Formalism
 - c. Bison Normal Format
 - d. Bound Non-Finite (automaton)
- Q.22 Classez les langages suivants par date d'apparition croissante.
- a. Ada, Algol, FORTRAN, PL/I
 - b. FORTRAN, Algol, PL/I, Ada
 - c. FORTRAN, PL/I, Algol, Ada
 - d. PL/I, FORTRAN, Ada, Algol
- Q.23 Lequel de ces langages n'est pas normalisé ?
- a. Ada
 - b. C++
 - c. Fortran
 - d. Java

6 Fonctions

- Q.24 Le support des fonctions récursives nécessite
- a. un tas (*heap*).
 - b. une pile (*stack*).
 - c. la liaison des fonctions dynamiques.
 - d. que le langage dispose de pré-déclarations (*forward declarations*).
- Q.25 Parmi les expressions C++ ci-dessous, laquelle effectue une allocation sur le tas ?
- a. `int p[42];`
 - b. `int p(42);`
 - c. `int* p = new int[42];`
 - d. `int* p = (int*) alloca(42);`