

Compléments de programmation - Contrôle continu (déc. 2018)

1:1

Durée : 1h. **Document autorisé :** une feuille A4 recto-verso, tout autre document interdit.
Les réponses aux questions sont à donner exclusivement sur la feuille de réponses (au verso). Les réponses données sur les pages de questions ne sont pas prises en compte.
Il y a une unique bonne réponse par question.

GESTION DE VERSIONS

Question 1 Quelle commande git permet de mettre à jour un dépôt local à partir d'un dépôt distant ?

- ☐ A remote ☐ B push ☐ C pull

Question 2 L'index d'un dépôt git est utilisé

- ☐ A pour indexer l'ensemble des fichiers du projet
☐ B pour éditer les fichiers du projet
☐ C pour accumuler les changements pour le prochain commit

Question 3 Pour créer un nouveau projet git local, on utilise la commande

- ☐ A init ☐ B checkout ☐ C clone

TESTS

Question 7 Un test unitaire doit

- ☐ A tester une unique méthode
☐ B tester une classe complète
☐ C tester un module complet

Question 8 Un cas de test JUnit 4

- ☐ A doit être une méthode de classe (static)
☐ B est annoté par @Test
☐ C se trouve dans une classe héritant de TestCase

Question 9 Un cas de test JUnit 4

- ☐ A se trouve dans la classe à tester
☐ B est fortement dans une suite de tests
☐ C se trouve dans une classe de test

Question 10 Pour utiliser JUnit 4, il faut

- ☐ A importer le package junit.framework
☐ B importer le package org.junit.jupiter.api
☐ C importer le package org.junit

BUILD AVEC MAVEN

Question 14 Pour construire une archive du projet, on utilise la commande mvn ...

- ☐ A compile ☐ B package ☐ C install

Question 15 Pour utiliser une bibliothèque B dans le projet courant, on ajoute au POM

- ☐ A les coordonnées du projet courant dans dependency
☐ B les coordonnées de B dans properties
☐ C les coordonnées de B dans dependency

Question 16 Les fichiers source de l'application d'un projet maven/java doivent se trouver dans

- ☐ A src/main/java
☐ B src
☐ C src/test/java

Question 17 Pour fixer la version Java des fichiers du projet, on définit

- ☐ A maven.compiler.source dans properties
☐ B maven.compiler.source dans dependencies
☐ C maven.compiler.target dans properties

Question 18 Pour placer une archive du projet dans le référentiel local, on utilise la commande

- ☐ A compile ☐ B install ☐ C package

POO

Soit le code Java suivant:

```
abstract class A { String toString() { return "A:toString"; }}
class B extends A { String toString() { return "B:toString"; }}
class C extends A { String toString() { return super.toString() + "C:toString"; }}
class D extends C { String toString() { return "D:toString"; }}
Stream<Integer> aStream = Stream.iterate(1, i -> 2 * i);
```

Question 19 Le code A a = new B(); System.out.println((C)a); produira

- ☐ A B:toString ☐ B A:toString ☐ C une erreur de compilation ☐ D A:toStringC:toString
☐ E une erreur d'exécution

Question 20 Pour définir l'ordre naturel des instances d'une classe, il faut

- ☐ A implémenter l'interface Comparator
☐ B implémenter l'interface Comparable
☐ C redéfinir <
☐ D juste implémenter une méthode compareTo

Question 21 Le code A a = new A(); System.out.println(a); produira

- ☐ A B:toString ☐ B une erreur de compilation ☐ C A:toString ☐ D une erreur d'exécution

Question 22 Le code D a = new C(); System.out.println(a); produira

- ☐ A A:toString ☐ B A:toStringC:toString ☐ C C:toString ☐ D une erreur d'exécution
☐ E une erreur de compilation

Question 23 Le code int val = aStream.limit(10).filter(e -> e < 32).map(e -> e / 2).reduce(0, (c, e) -> c + e); produira la valeur

- ☐ A 15 ☐ B 16 ☐ C 31 ☐ D 7

Question 24 Le code B a = new B(); System.out.println(a); produira

- ☐ A B:toString ☐ B A:toString ☐ C une erreur de compilation ☐ D une erreur d'exécution

Question 25 Le code A a = new C(); System.out.println(a); produira

- ☐ A une erreur de compilation ☐ B C:toString ☐ C A:toString ☐ D une erreur d'exécution
☐ E A:toStringC:toString

1:2