



+61/1/60+

Qualité & Génie Logiciel - QCM 1 - 29.02.2018

<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0
<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1	<input type="checkbox"/> 1
<input checked="" type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input type="checkbox"/> 2	<input checked="" type="checkbox"/> 2	<input checked="" type="checkbox"/> 2
<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3	<input type="checkbox"/> 3
<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input checked="" type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4	<input type="checkbox"/> 4
<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5	<input type="checkbox"/> 5
<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6	<input type="checkbox"/> 6
<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input checked="" type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7	<input type="checkbox"/> 7
<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8	<input type="checkbox"/> 8
<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input checked="" type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9	<input type="checkbox"/> 9

← Noircissez les cases pour encoder votre numéro étudiant.

Nom et prénom :

.....SALORD.....
.....FLORIAN.....

Veuillez répondre aux questions ci-dessous du mieux que vous pouvez en noircissant les cases (pas en les cochant). Chaque question à une et une seule bonne réponse. Une bonne réponse rapporte un point, une mauvaise réponse fait perdre un point. Tout autre situations (plusieurs cases noircies, absence de réponse) ne fait ni perdre ni gagner de point.

Question 1 Dans l'acronyme SOLID, que signifie le 'S' ?

- ☐ Small (size)
- ☐ Safe (usage)
- ☒ Single (responsibility)
- ☐ Simple (behavior)

Question 2 Selon le principe Ouvert / Fermé, une classe est Ouverte :

- ☒ à l'extension
- ☐ au chargement dynamique
- ☐ à la modification
- ☐ à l'appel de méthode

Question 3 Dans un test, un bouchon (mock) est utilisé pour :

- ☐ observer comment les méthodes des objets bouchonnés sont appelées
- ☐ tester des classes impossible a tester autrement
- ☐ produire un rapport de consommation mémoire du programme
- ☒ remplacer le comportement des méthodes bouchonnés

Question 4 Du point de vue du génie logiciel, dans le code source d'un logiciel, on cherche en premier lieu :

- ☒ La lisibilité
- ☐ la présence systématique de commentaires
- ☐ la performance
- ☐ l'utilisation de structures itératives plutôt que récursives

Question 5 Dans un système de gestion de version distribué comme Git, l'opération 'add' permet :

- ☒ d'enregistrer une modification dans le dépôt local
- ☐ de partager les modifications du dépôt local avec un dépôt distant
- ☒ d'enregistrer un fichier dont les modifications seront versionnées
- ☐ de récupérer les modifications faites sur un dépôt distant

Question 6 Un programme orienté-objet recherche entre ses classes :

- ☐ Un couplage fort ET une forte cohésion
- ☐ Un couplage fort ET une faible cohésion
- ☒ Un couplage faible ET une forte cohésion
- ☐ Un couplage faible ET une faible cohésion

1/1

1/1

1/1

-1/1

1/1

1/1



Question 7 Dans un gestionnaire de tickets comme Jira, une tâche est un ticket :

- ☒ métier, à destination du client
- ☒ technique, à destination de l'équipe de développement
- ☐ technique, décrivant un bug dans le programme
- ☐ aucun des trois

Question 8 Dans l'acronyme BICEP (bonnes pratiques de test logiciel), le B signifie :

- ☒ Boundaries
- ☐ Backend
- ☒ Binary
- ☐ Byte

Question 9 Dans les principes de V&V, la Vérification permet de répondre à la question:

- ☐ Faisons-nous un programme utilisable ?
- ☒ Faisons-nous le produit correctement ?
- ☐ Faisons-nous le bon produit ?
- ☐ Faisons-nous un programme maintenable ?

Question 10 Maven est un outil permettant :

- ☐ l'implémentation d'un programme en Java
- ☒ la gestion des dépendances entre modules
- ☐ le chargement d'un projet par un IDE
- ☐ la définition de tests unitaires

Question 11 Dans un langage objet réflexif, la notion d'introspection permet :

- ☐ à l'IDE de fournir des opérations de refactoring
- ☐ au compilateur de produire le code machine
- ☒ d'observer les classes comme des objets
- ☐ d'ajouter de nouveaux attributs dans une classe pendant l'exécution

Question 12 Un test qui réussit :

- ☐ est nécessaire pour compiler le programme testé
- ☒ démontre un comportement attendu dans une situation donnée
- ☐ est nécessaire pour exécuter le programme testé
- ☐ démontre l'absence d'erreurs

Question 13 Les tests logiciels s'inscrivent, dans une démarche de V&V :

- ☐ dans aucun des deux principes
- ☐ dans la Validation
- ☒ dans la Vérification
- ☒ dans la Validation ET la vérification

Question 14 Pour partager ses modifications avec un dépôt distant, Erik doit enchaîner les 4 opérations 'add', 'commit', 'push' et 'pull' dans l'ordre suivant :

- ☐ add / push / commit / pull
- ☐ push / add / commit / pull
- ☐ commit / add / pull / push
- ☒ pull / add / commit / push

Question 15 Alice vient de faire un 'pull' du dépôt distant 'origin'. La commande Git se termine bien (pas de conflits détecté), mais son code ne compile plus. Pourquoi ?

- ☐ Cette situation ne peut jamais arriver
- ☐ Les autres développeurs ont "cassé" le code, c'est à Alice de réparer
- ☒ Ses modifications sont incompatibles avec celles effectuées sur 'origin'
- ☐ C'est normal, Git ne sait pas détecter automatiquement un conflit