



Qualité & Génie Logiciel - QCM 2 - 03.05.2018

0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	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.

Q 1 La notion de cohésion est définie comme :

- ☒ Le degré d'interdépendance entre les éléments dans un même module
- ☐ Le nombre de classes dans le système
- ☐ Le nombre d'attribut d'une classe
- ☐ Le degré d'interdépendance entre les éléments de différents modules

Q 2 Complétez la définition de la règle de trois fournie par Martin Fowler: "The first time you do something, you just do it. The second time you do something similar, you wince at the duplication, but you do the duplicate thing anyway. The third time you do something similar, you ..."

- ☒ ... refactor.
- ☐ ... test.
- ☐ ... run the compiler.
- ☐ ... run quality analysis.

Q 3 Dans une approche de "test par mutation", une mutation est une transformation de code qui :

- ☐ modifie un cas de test en changeant ses valeurs
- ☐ modifie le code sources des tests
- ☒ modifie le code source du programme
- ☐ modifie la manière dont le projet est compilé

Q 4 PiTest est un outil :

- ☐ de "property based testing"
- ☐ de test unitaire
- ☐ de test d'acceptation
- ☒ de test par mutation

Q 5 Quand un conflit arrive dans Git, cela veut dire :

- ☐ que le développeur perd toutes ses modifications locales
- ☐ que le dépôt Git distant n'est plus accessible
- ☒ que d'autres développeurs ont travaillé au même endroit dans le code
- ☐ cette situation ne peut arriver avec Git

Q 6 Le refactoring est une opération qui consiste à :

- ☐ ajouter de nouvelles fonctionnalités
- ☐ modifier le comportement d'un programme en modifiant le moins possible sa structure
- ☒ modifier la structure d'un programme sans modifier son comportement
- ☐ ajouter de nouveaux tests dans une suite de tests

RIEN COCHÉ



Q 7 Une classe difficile à tester unitairement est souvent un symptôme :

- ☐ des faiblesses d'expressivité de JUnit
- ☐ d'une mauvaise séparation en package des tests
- ☒ d'une mauvaise conception objet du programme
- ☐ de la non adéquation des tests unitaires avec les principes de la programmation objet

Q 8 Pour explorer l'espace des valeurs possible, une approche de "property based testing" repose sur:

- ☒ des fonctions appelées "générateurs"
- ☐ de fonctions appelées "mutations"
- ☐ de cas prédéfinis qui servent de valeurs étalon
- ☐ de cas prédéfinis fournis par le développeur

Q 9 Dans un système de gestion de version distribué comme Git, l'opération "pull" permet :

- ☒ de récupérer les modifications faites sur un dépôt distant
- ☐ 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

Q 10 La notion de couplage est définie comme :

- ☒ Le degré d'interdépendance entre les éléments de différents modules
- ☐ Le nombre de classes dans le système
- ☐ Le nombre d'attribut d'une classe
- ☐ Le degré d'interdépendance entre les éléments dans un même module

Q 11 Dans un test, l'oracle est :

- ☒ en charge de connaître la valeur attendue
- ☐ un personnage de Matrix pliant des cuillères
- ☐ une méthode statique de JUnit
- ☐ en charge de lancer le test et de stocker son résultat

Q 12 Dans un langage objet réflexif, la notion d'intercession permet :

- ☐ d'observer les classes comme des objets
- ☐ au compilateur de produire le code machine
- ☐ à Git de résoudre les conflits de gestion de version
- ☒ d'ajouter de nouveaux attributs dans une classe pendant l'exécution

Q 13 Un langage orienté objet contient obligatoirement la notion :

- ☐ de visibilité (public, privé, ...)
- ☐ d'héritage entre classes
- ☐ de classe
- ☒ d'envoi de message à un objet

Q 14 Pour que le résultat d'une approche de test par mutation soit satisfaisante pour un programme donné, il faut chercher à :

- ☐ maximiser le nombre de mutants créés
- ☒ maximiser le nombre de mutants tués
- ☐ minimiser le nombre de mutants tués
- ☐ minimiser le nombre de mutants tués

Q 15 Les tests par mutations servent à :

- ☒ tester la qualité d'un banc de test existant
- ☐ tester le code source d'un programme
- ☐ valider des propriétés de haut niveau
- ☐ générer automatiquement des nouveaux tests JUnit

1/1

1/1

1/1

1/1

1/1