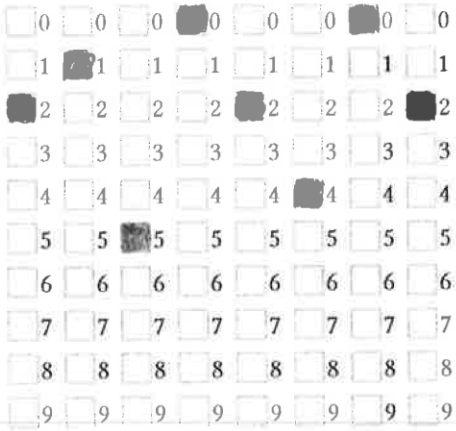
STROBBE
Mark: 16/20 (total score: 12/15)

+23/1/16+

# Qualité & Génie Logiciel - QCM 2 - 03.05.2018



← Noircissez les cases pour encoder votre numéro étudiant.

Nom et prénom :
STROBBE
Nathan G2

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** Le refactoring est une opération qui consiste à :

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

1/1

**Q 2** 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 crées
- [x] maximiser le nombre de mutants tués
- [ ] minimiser le nombre de mutants tués
- [ ] minimiser le nombre de mutants tués

1/1

**Q 3** PiTest est un outil :

- [ ] de test d'acceptation
- [ ] de test unitaire
- [x] de test par mutation
- [ ] de "*property based testing*"

1/1

**Q 4** 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 la manière dont le projet est compilé
- [x] modifie le code source du programme
- [ ] modifie le code sources des tests

1/1

**Q 5** Une classe difficile à tester unitairement est souvent un symptôme :

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

0/1

**Q 6** Dans un langage objet réflexif, la notion d'intercession permet :

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

1/1

Q 7 Un langage orienté objet contient obligatoirement la notion :

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

0/1

Q 8 Dans un système de gestion de version distribué comme Git, l'opération "*pull*" permet :

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

1/1

Q 9 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 ...*"

- [x] ... refactor.
- [ ] ... run quality analysis.
- [ ] ... run the compiler.
- [ ] ... test.

1/1

Q 10 La notion de couplage est définie comme :

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

1/1

Q 11 Quand un conflit arrive dans Git, cela veut dire :

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

1/1

Q 12 La notion de cohésion est définie comme :

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

1/1

Q 13 Dans un test, l'oracle est :

- [x] en charge de connaitre 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

0/1

Q 14 Les tests par mutations servent à :

- [ ] tester le code source d'un programme
- [ ] valider des propriétés de haut niveau
- [x] tester la qualité d'un banc de test existant
- [ ] générer automatiquement des nouveaux tests Junit

1/1

Q 15 Pour explorer l'espace des valeurs possible, une approche de "property based testing" repose sur:

- [ ] de cas prédéfinis qui servent de valeurs étalon
- [x] des fonctions appelées "générateurs"
- [ ] de cas prédéfinis fournis par le développeur
- [ ] de fonctions appelées "mutations"

1/1