

Question #1

Quelle technique de mise sous test devons-nous utiliser si nous voulons simuler le comportement d'une dépendance mais sans tester son utilisation?

- a) Fake object
- b) Mock object
- c) Injection de dépendance
- d) Design for testability
- e) Test de caractérisation
- f) Test-Driven Development
- g) Aucune de ces réponses

Question #2

Quelle technique de mise sous test devons-nous utiliser pour faciliter le développement de nouvelles fonctionnalités avec beaucoup de logique d'affaires?

- a) Fake object
- b) Mock object
- c) Injection de dépendance
- d) Design for testability
- e) Test de caractérisation
- f) Test-Driven Development
- g) Aucune de ces réponses

Question #3

Quelle technique de mise sous test devons-nous utiliser si nous voulons faire du refactoring sécuritaire dans une longue méthode complexe qui ne possède aucune couverture de tests?

- a) Fake object
- b) Mock object
- c) Injection de dépendance
- d) Design for testability
- e) Test de caractérisation
- f) Test-Driven Development
- g) Aucune de ces réponses

Question #4

Quelle affirmation suivante est vraie?

- a) L'injection de dépendance permet une meilleure encapsulation
- b) Le Design for testability est un ensemble de règles qui favorisent l'encapsulation
- c) Un Fake object est un Mock object plus évolué
- d) Un Mock object est habituellement utilisé avec l'injection de dépendance
- e) L'injection de dépendance permet d'ajouter des dépendances à une classe

Question #5

Quelle affirmation décrit le mieux le TDD?

- a) green, red, refactor
- b) Il faut écrire les tests le plus rapidement possible après le code
- c) Il faut un test pour chaque ligne de code dans une méthode
- d) Un peu de tests, un peu de code, un peu de nettoyage
- e) Toutes ces réponses

Question #6

Selon eXtreme Programming, une équipe de développement devrait travailler dans une aire ouverte. Quelle est la valeur véhiculée par cette pratique?

- a) Communication
- b) Simplicité
- c) Feedback
- d) Courage
- e) Respect

Question #7

Parmi les pratiques suivantes, laquelle est une pratique de contrôle de la qualité?

- a) Utilisation d'un gestionnaire de sources
- b) Intégration continue
- c) Construction automatisée
- d) Scrum
- e) Norme de codification

Question #8

À quelle vitesse doit-on adopter les pratiques d'eXtreme Programming?

- a) Instantanément, toutes à la fois
- b) Progressivement, une pratique à la fois
- c) Moyennement vite, deux ou trois pratiques à la fois
- d) Rapidement, quatre ou cinq pratiques à la fois
- e) XP ne traite que de gestion de projet

Question #9

Selon eXtreme Programming, quelle devrait être la longueur d'une itération?

- a) 1 jour
- b) 1 semaine
- c) 2 semaines
- d) 1 mois
- e) 3 mois

Question #10

Quelle affirmation suivante est fausse?

- a) XP recommande de faire de l'intégration continue
- b) XP recommande de faire de la révision de code
- c) XP recommande d'automatiser la construction du logiciel en moins de 10 minutes
- d) XP recommande de faire de la programmation en binôme
- e) Aucune de ces réponses

Question #11

Quelle pratique doit-on maîtriser avant de mettre en place la construction automatisée?

- a) Tests unitaires
- b) Intégration continue
- c) Utilisation d'un gestionnaire de sources
- d) Révision de code
- e) Toutes ces réponses

Question #12

Quelle est la caractéristique principale de Ant par rapport à ses prédécesseurs?

- a) La portabilité
- b) La vitesse d'exécution
- c) La simplicité de sa syntaxe
- d) Facile à installer et configurer
- e) La courbe d'apprentissage aisée

Question #13

Quel outil de construction permet de faire une gestion simplifiée des dépendances d'un projet?

- a) make
- b) Ant
- c) Maven
- d) Gradle
- e) Buildr
- f) Aucune de ces réponses

Question #14

Quel est l'outil de construction utilisé par défaut avec NetBeans?

- a) make
- b) Ant
- c) Maven
- d) Gradle
- e) Buildr
- f) Aucune de ces réponses

Question #15

Pourquoi voudrait-on suivre les conventions de Maven?

- a) Parce que Maven ne fonctionne pas autrement
- b) Pour augmenter la vitesse de la construction
- c) Pour augmenter la portabilité du projet
- d) Pour minimiser la configuration du projet
- e) Pour améliorer l'intégration à NetBeans
- f) Toutes ces réponses

Question #16

Quelle pratique améliore grandement l'utilité de l'intégration continue?

- a) La révision de code
- b) L'utilisation d'un gestionnaire de sources
- c) Les tests automatisés
- d) Le développement itératif
- e) Le déploiement quotidien
- f) Toutes ces réponses

Question #17

Qu'est-ce qu'on intègre dans l'intégration continue?

- a) Les outils de l'équipe
- b) Le travail de l'équipe
- c) Les test automatisés
- d) Les pratiques émergentes en développement de logiciels
- e) Des nouveaux membres dans l'équipe

Question #18

Quelle affirmation est vraie lorsqu'on fait de l'intégration continue?

- a) Les branches sont interdites
- b) Les commits sont plus fréquents
- c) Le serveur d'intégration continue est optionnel
- d) La construction n'est que lancée avant la livraison
- e) Uniquement le responsable est avisé des défauts

Question #19

Quelle affirmation suivante est vraie?

- a) Le daily scrum est principalement pour le product owner
- b) Le daily scrum est principalement pour l'équipe de développement
- c) Le daily scrum est principalement pour l'assurance-qualité
- d) Le daily scrum est principalement pour le ScrumMaster
- e) Toutes ces réponses

Question #20

Quel est l'objectif principal de Scrum?

- a) Résolution de problèmes complexes
- b) Amélioration de la profession
- c) Vendre des certifications
- d) Offrir une solution Agile pour les développeurs conservateurs
- e) Aucune de ces réponses

Question #21

Quel élément n'est pas une valeur importante de Scrum?

- a) Transparence
- b) Adaptation
- c) Courage
- d) Inspection
- e) Aucune de ces réponses

Question #22

Qui a la responsabilité d'encadrer l'équipe Scrum dans le but de la rendre plus autonome?

- a) Gestionnaire de projet
- b) Product owner
- c) ScrumMaster
- d) Équipe de développement
- e) Aucune de ces réponses

Question #23

Qui a la responsabilité d'affecter les tâches aux développeurs d'une équipe Scrum?

- a) Gestionnaire de projet
- b) Product owner
- c) ScrumMaster
- d) Équipe de développement
- e) Aucune de ces réponses

Question #24

Qui a la responsabilité d'entretenir le carnet de sprint dans une équipe Scrum?

- a) Gestionnaire de projet
- b) Product owner
- c) ScrumMaster
- d) Équipe de développement
- e) Ce livrable n'existe pas

Question #25

Qui a la responsabilité d'entretenir le carnet de livraison dans une équipe Scrum?

- a) Gestionnaire de projet
- b) Product owner
- c) ScrumMaster
- d) Équipe de développement
- e) Ce livrable n'existe pas

Question #26

Quel est l'objectif de la revue de sprint?

- a) Présenter l'avancement du projet aux parties prenantes
- b) Planifier les tâches du sprint
- c) Point d'inspection quotidien sur l'avancement du sprint
- d) Améliorer les pratiques de l'équipe
- e) Affecter les tâches aux membres de l'équipe

Question #27

Quel est le langage le moins expressif parmi les suivants?

- a) Assemblage PEP/8
- b) Java
- c) Groovy
- d) Python
- e) Langage C
- f) Ruby

Question #28

Pourquoi est-il possible de mélanger des fichiers Groovy et Java dans un même projet?

- a) Parce que Java peut lire le Groovy
- b) Parce que Groovy peut produire du bytecode
- c) Parce que la machine virtuelle Java interprète le Groovy
- d) Parce que Groovy est rapide et puissant
- e) Aucune de ces réponses

Question #29

Parmi les éléments suivants, lequel n'est pas un objectif d'une révision de code?

- a) Détection des erreurs
- b) Communication dans l'équipe
- c) Formation
- d) Productivité
- e) Transfert de connaissances

Question #30

Quelle affirmation suivante est fausse?

- a) L'inspection formelle nécessite plus de temps
- b) L'inspection formelle permet de détecter plus d'erreurs
- c) La révision informelle nécessite une réunion avec beaucoup de participants
- d) Le pair-programming est une forme de révision
- e) Le pair-programming est recommandé par eXtreme Programming