Examen – Microservices & Spring Boot

1. Quelles affirmations sont vraies à propos des architectures Microservices ?
   a. Elles sont plus flexibles qu'une architecture monolithique
   b. Elles sont plus rapides à développer en jour/homme total que des architectures monolithiques
   c. Elles ont un « time to market » potentiel plus court que dans le cas des architectures monolithiques
   d. Le « time to market » sera égal à la somme des temps de développements de tous les services

2. Quelle affirmation est vraie à propos des architectures Microservices ?
   a. Faire évoluer le projet après le développement initial est complexe
   b. Le coût initial global de développement sera supérieur comparativement aux architectures monolithiques
   c. Le coût des tâches d'évolution sera supérieur comparativement aux architectures monolithiques
   d. Le coût d'hébergement sera supérieur comparativement aux architectures monolithiques

3. Quelles affirmations sont vraies à propos des architectures Microservices ?
   a. Elles supportent mieux les modifications en cours de projet que les architectures monolithiques
   b. Elles sont plus résilientes face à la complexité
   c. Elles sont faites pour être réalisées par une petite équipe de développeur
   d. Elles vont diminuer la compétitivité de l'entreprises qui la mette en œuvre

4. Quelles affirmations sont vraies à propos des architectures Microservices ?
   a. Elles sont liées à des thèmes abordés dans les domaines des API REST
   b. Echangent principalement via des échanges SOAP
   c. Chaque composant à une visibilité sur l'architecture complète
   d. Elles sont étroitement liées avec le Single Responsibility Principle

5. Quelles affirmations sont vraies à propos des architectures Microservices ?
   a. Elles doivent être développées sur un seul et même environnement technique (tout en Spring ou tout en Python par exemple)
   b. Elles peuvent avoir autant de contexte techniques différents de que « services »
   c. Elles doivent partager une même base de données pour tous les « services »
   d. Elles doivent être développées pour fonctionner sur des bases de données relationnelles
   e. Elles peuvent faire cohabiter tous types de base de données simultanément

6. Quelles affirmations sont vraies à propos de Spring Boot ?
   a. C'est l'équivalent de Spring avec d'autres langages
   b. C'est une surcouche du framework Spring
   c. Il s'agit uniquement d'une solution pour créer un projet Spring
   d. S'appuie sur un ensemble de « starters »

7. La gestion de la persistance dans Spring Boot via JPA implique plusieurs composants, quels sont-ils ?
   a. Les « Repository »
   b. Les « Controllers »
   c. Les « RestController »
   d. Les « Entity »
   e. Les « Service »

ABE

8. Quels sont les « Repositories » dont on peut « extends » ?
   a. EntityRepository
   b. CrudRepository
   c. JpaRepository
   d. RequestRepository
   e. Repository
   f. SimpleRepository

BC

9. Lors de la création d'un « Repository », on doit « extends » le « Repository » souhaité et préciser ces éléments …
   a. La cardinalité des relations
   b. Les contraintes à appliquer sur le modèle associé à l'entité
   c. Le type de l'identifiant de l'entité désignée
   d. La classe de l'entité désignée

D

10. Les « Repository » exposent un certain nombre de méthodes, quelles sont-elles ?
    a. save
    b. findAll
    c. findAllByName
    d. update

B

11. Quelles seront les conséquences de la déclaration dans un « RestController » d'une méthode précédé de l'annotation « @GetMapping(value = "**/products**") »
    a. La méthode sera exécutée lors d'un appel sur l'url « …/products » en GET
    b. La méthode sera exécutée lors d'un appel sur l'url « …/products » en POST
    c. La méthode renverra forcément une liste de produits
    d. La méthode doit renvoyer une réponse à l'utilisateur sans quoi le serveur retournera une erreur 500

C

12. L'injection de dépendances peut se faire de plusieurs manières, quelles sont-elles ?
    a. Injection sur les attributs
    b. Injection sur les méthodes
    c. Injection fonctionnelle
    d. Injection sur les constructeurs
    e. Injection prédictive
    f. Injection sur la classe
    g. Injection sur les getters / setters

AB

Examen – SOLID Principles

1. Quelles affirmations sont vraies à propos des principes SOLID ?
   a. Ils vous aident à concevoir et produire des architectures flexibles, scalable et facile à maintenir
   b. L'ensemble de principes est voué à produire des architectures avec comme soucis principal la robustesse
   c. Les principes sont un ensemble de règles à respecter qui décrivent la manière d'implémenter vos architectures dans différents langages
   d. Les principes sont un ensemble des règles et bonnes pratiques de haut niveau sans lien avec une implémentation particulière

2. Quelles affirmations sont vraies à propos du **Single Responsibility Principle** ?
   a. Dit qu'une classe doit avoir une seule raison de changer
   b. Dit qu'une classe ne doit avoir qu'une seule interface
   c. Dit qu'une classe ne doit être parent que d'un seul héritage
   d. Fait un parallèle entre la « responsabilité » et les « raisons de changer »

3. La mise en œuvre du **Single Responsibility Principle** va aider à obtenir des architectures ...
   a. Robustes
   b. Véloces
   c. Avec des éléments réutilisables
   d. Résistantes aux intrusions

4. Dans la mise en œuvre du **Single Responsibility Principle** on peut segmenter plus ou moins une architecture, le niveau de segmentation dépendra de plusieurs critères, lesquels ?
   a. Le langage utilisé pour le développement de l'architecture
   b. La taille de l'équipe de développement
   c. Les compétences de l'équipe de développement
   d. Du type de gestion de projet utilisé
   e. Des contraintes de temps

5. Quelles affirmations sont vraies à propos de l'**Open / Closed Principle** ?
   a. La visibilité de vos attributs doit être précise et adaptée
   b. Vos classes doivent être ouvertes aux modifications
   c. Vos classes doivent être ouvertes aux extensions
   d. Vos classes doivent être fermées aux modifications
   e. Vos classes doivent être fermées aux extensions

6. Par quels moyens peut-on mettre en œuvre de ce principe ?
   a. L'héritage
   b. Les interfaces
   c. L'association
   d. La composition
   e. La relation

7. En Java, dans quels cas considère-t-on que S est un sous type de T ?
   a. Lorsque S est T
   b. Lorsque S est composé de T
   c. Lorsque S hérite de T

d. Lorsque S implémente T
e. Lorsque T est composé de S

8. Quelles affirmations sont vraies à propos du **Principe de Substitution de Liskov** ?

BC

a. Un sous-type doit pouvoir être remplaçable par son type de base
b. Un type de base doit pouvoir être remplaçable par son sous-type
c. Si S est un sous type déclaré de T, les objets de type S doivent se comporter comme les objets de type T sont censés se comporter s'ils sont traités comme des objets de type T
d. Une classe et son sous type doivent toujours hériter d'une superclasse

9. Quelles affirmations sont vraies à propos du **Principe de Substitution de Liskov** ?

C

a. Il doit y avoir contra variance des arguments
b. Il doit y avoir contra variance des types de retour
c. Il doit y avoir covariance des arguments
d. Il doit y avoir covariance des types de retour

10. Quelles affirmations sont vraies à propos du **Principe de Ségrégation des Interfaces** ?

D

a. Il vaut mieux créer des Interfaces génériques
b. Il vaut mieux créer des interfaces spécifiques
c. Il faut créer une interface par méthode « générique »
d. Il faut faire en sorte d'avoir des interfaces qui font sens

F

11. Quelles affirmations sont vraies à propos du **Principe d'Inversion des Dépendances** ?

a. Les modules de haut niveau doivent dépendre des modules de bas niveau
b. Les modules de bas niveau doivent dépendre des modules de haut niveau
c. Les modules de haut et bas niveau doivent dépendre d'abstractions
d. Les abstractions doivent dépendre des modules de haut niveau
e. Les abstractions doivent dépendre des modules de bas niveau
f. Est une conséquence naturelle de l'application du 2ème et 3ème principe

BC

12. Quels sont les objectifs de ces principes (Celle-là c'est pour moi ;)) ?

a. Être capable de briller en société
b. Être capable de concevoir des systèmes flexibles, tolérants aux pannes, scalable avec des composants hautement indépendants (Stateless) qui vous permettront facilement de mettre en œuvre des architectures type « Micro-services »
c. Donner un ensemble de guidelines qui vont vous mener vers un ensemble de bonnes pratiques qui vous permettront de vous adapter à n'importe quelle équipe de développement