```
4GL1 - Spring - Evaluation Blanche
Total des points 7/20
- Début : Mercredi 20/05 à 09h00
- Deadline : Vendredi 22/05 à 09h00
- Pour chaque question, au moins une réponse est bonne.
L'adresse e-mail du répondant (medomar.walha@esprit.tn) a été enregistrée lors de l'envoi
de ce formulaire.
X Question 1: En ce qui concerne Spring AOP *
                                                                             0/1
     L'AOP est un Paradigme de Programmation.
 L'AOP permet d'ajouter des comportements à du code existant.
 L'AOP est un design Pattern.
                                                                             X
     L'AOP est l'abréviation de : Appellation d'Origine Protégée.
Bonne réponse
 L'AOP est un Paradigme de Programmation.
     L'AOP permet d'ajouter des comportements à du code existant.
X Question 2 : Quelle annotation est utilisée pour mapper une méthode
                                                                             0/1
    d'un Web Service REST pour une demande de mise à jour *
     @Put
     @PutMapping
     @PostMapping
     @Post
Bonne réponse
 @PutMapping
X Question 3 : Pour une entité classe Voiture(id, modele, annee), quelles
     méthodes sont automatiquement créées dans CrudRepository *
     findModele
 findByModele
     find By Modele And Id And Annee \\
     findVoitureByModele
Bonne réponse
 findByModeleAndIdAndAnnee
 findByModele
✓ Question 4 : Quelle propriété est utilisée pour changer le port à 8081
                                                                             1/1
     dans une application Spring Boot *
 server.port=9090
     server=9090
     il faut modifier dans les fichiers de configuration du serveur Tomcat
     port=9090
X Question 5 : Dans une application Web Spring, lequel est le Front
                                                                             0/1
     Controller *
     La classe Spring org.springframework.stereotype.Controller
 La classe Spring org.springframework.web.bind.annotation.RestController
                                                                            X
     La classe org.springframework.web.servlet.ViewResolver
     La classe Spring org.springframework.web.servlet.DispatcherServlet
Bonne réponse

    La classe Spring org.springframework.web.servlet.DispatcherServlet

✓ Question 6 : Si vous rencontrez l'erreur « Caused by: ... : Address already 1/1
    in use: bind », en lançant votre projet Spring Boot (Run As Spring Boot
     App), c'est quoi la/les cause(s) probable(s) de cette erreur *
     Le code contient une erreur de compilation.
 Le serveur Tomcat est déjà lancé.
     Le serveur de base de données est arrêté.
     Le serveur Tomcat est arrêté.
X Question 7 : Que signifie la première étoile dans l'expression ci-dessous : 0/1
@After("execution(* tn.esprit.spring.service.*.*(..))")
     N'importe quelle visibilité (public, private, ..).
     N'importe quel package, n'importe quel type de retour et n'importe quelle
                                                                            X
     visibilité.
     N'importe quel type de retour (void, String, ...).
     N'importe quel package.
Bonne réponse
 N'importe quel type de retour (void, String, ...).

★ Question 8 : En ce qui concerne l'advice de type @After et l'advice de 0/1

    type @AfterThrowing *
@After("execution(* tn.esprit.spring.service.*.*(..))")
public void logMethodExit(JoinPoint joinPoint) {
     String name = joinPoint.getSignature().getName();
     L.info("Out of method (After) : " + name );
@AfterThrowing("execution(* tn.esprit.spring.service.*.*(..))")
public void logMethodExitWithErrors(JoinPoint joinPoint) {
     String name = joinPoint.getSignature().getName();
     L.error("Out of method with errors (After Throwing) : " + name );
     Si un Join Point (méthode d'une classe du package tn.esprit.spring.service) est
     exécutée, alors les deux méthodes logMethodExit et logMethodExitWithErrors
     seront toutes les deux exécutées. .
     Si un Join Point (méthode d'une classe du package tn.esprit.spring.service) est
     exécutée, alors aucune des deux méthodes logMethodExit et
     logMethodExitWithErrors ne seront exécutées. .
     Si un Join Point (méthode d'une classe du package tn.esprit.spring.service) est
     exécutée, alors la méthode logMethodExitWithErrors peut être exécutée.
     Si un Join Point (méthode d'une classe du package tn.esprit.spring.service) est
     exécutée, alors la méthode logMethodExit sera exécutée.
Bonne réponse
     Si un Join Point (méthode d'une classe du package tn.esprit.spring.service) est
     exécutée, alors la méthode logMethodExit sera exécutée.
     Si un Join Point (méthode d'une classe du package tn.esprit.spring.service) est exécutée, alors la méthode logMethodExitWithErrors peut être exécutée.
X Question 9 : Pour créer un bean Spring dans l'loC Container, quelle(s)
                                                                             0/1
    annotion(s), nous pouvons utiliser *
     @Service
     @Autowired
 @Resource
                                                                             X
     @Bean
Bonne réponse
     @Service
     @Bean
✓ Question 10 : Soit le contenu suivant d'un pom.xml. Où se trouvera le jar 1/1
    spring-boot-devtools dans un dépôt Maven local (local repository)?*
    <parent>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-parent</artifactId>
          <version>2.1.4.RELEASE
          <relativePath />
    </parent>
    <dependencies>
          <dependency>
               <groupId>org.springframework.boot</groupId>
               <artifactId>spring-boot-devtools</artifactId>
          </dependency>
    </dependencies>
    C:\.....\.m2\repository\org\spring\framework\boot\spring-boot-devtools\2.1.4\
 C:\.....\.m2\repository\org\springframework\boot\spring-boot-devtools\
     C:\.....\.m2\repository\org\springframework\boot\spring-boot-devtools\2.1.4\
     \label{lem:c:lim2} C:\\ \label{lem:c:lim2} C:\\ \label{lem:c:lim2} devtools\\ \end{cases}. 1.4. RELEASE\\
    Question 11 : Quelle annotation JUnit 4 permet de déclarer une méthode 1/1
     de test *
     @OnTest
     @JUnitTest
 @Test
 @RunWith
✓ Question 12 : Quels 'Join Point' correspondent à l'expression ci-desous : * 1/1
execution(* org.User.*(..))
     Toutes les méthodes commencent par User
     Toutes les méthodes contenues dans toutes les classes se trouvant dans le package
     org.User
     Toutes les méthodes contenues dans des classes se trouvant dans le package org
 Toutes les méthodes contenues dans la classe User
X Question 13 : Que signifie l'expression ci-dessous : *
                                                                             0/1
execution(* tn.esprit.spring.*.*.*(..))
     Toutes les méthodes contenues dans les classes se trouvant dans les packages 🗸
     immédiatement sous le package spring
     Toutes les méthodes contenues dans les classes se trouvant dans tous les sous-
     packages de spring
     Toutes les méthodes contenues dans les classes se trouvant dans le package spring
Bonne réponse
     Toutes les méthodes contenues dans les classes se trouvant dans les packages
     immédiatement sous le package spring
✓ Question 14 : Par convention, pour une classe Toto, c'est quoi le nom de 1/1
     la classe de test unitaire correspondante *
     TotoImp
 TotoTest
     TestToto
 ImpToto
X Question 15 : Que signifie la dernière étoile de l'expression ci-dessous : * 0/1
 @After("execution(* tn..*.*(..))")
 N'importe quelle classe.
                                                                            X
     N'importe quel jar.
 N'importe quelle méthode.
     N'importe quel package.
Bonne réponse
 N'importe quelle méthode.
X Question 16: Que signifie les deux points « .. » dans l'expression suivante 0/1
 execution(* tn..*.*())
     N'importe quel package.
     Le package tn et tous les sous packages de tn.
     N'importe quelle classe.
 Tous les sous packages de tn.
                                                                            X
Bonne réponse
 Le package tn et tous les sous packages de tn.
✓ Question 17 : L'expression suivante signifie : *
                                                                             1/1
 execution(* tn.*.*(..))
     Toutes les méthodes publiques se trouvant dans des classes dans le package tn.
 Toutes les méthodes publiques de tout notre projet.
     Toutes les méthodes de tout notre projet.
 Toutes les méthodes se trouvant dans des classes du package tn.
                                                                            /
X Question 18 : Quels sont les avantages de Spring Boot ? *
                                                                             0/1
     Centralise les configurations utilisées dans notre application.
     Facilite l'utilisation des systèmes de contrôle de version.
                                                                             X
 Facilite la gestion des dépendances Maven.
     Centralise les tests unitaires.
Bonne réponse
     Facilite la gestion des dépendances Maven.
 Centralise les configurations utilisées dans notre application.
X Question 19 : Cette expression signifie : *
                                                                             0/1
 execution(* tn.*.tn*(..))
     Toutes les méthodes qui commencent par tn et qui finissent par tn
     Toutes les méthodes qui commencent par tn
     Aucune de ses réponses
 Toutes les méthodes se trouvant dans des classes du package tn
Bonne réponse
 Aucune de ses réponses
X Question 20 : En en ce qui concerne STS *
                                                                             0/1
     Eclipse et STS sont deux IDE.
     Pour faire du Spring, il est obligatoire d'utiliser STS.
     Eclipse ne permet pas de développer des applications se basant sur Spring.
 STS est un Eclipse.
Bonne réponse
 STS est un Eclipse.
 Eclipse et STS sont deux IDE.
                           Ce formulaire a été créé dans esprit.
                              Google Forms
```