

### Exercice 3 : Hibernate et JPA

**Objectif :** Ce test a pour objectif de tester vos connaissances sur Hibernate, plusieurs réponses sont possibles.

1 – Hibernate est :

**a- Un framework open source gérant la persistance des objets en base de données relationnelle**

**b- Un ORM(Object-Relational Mapping)**

c- Un framework qui permet la sortie d'un programme

d- Je ne sais pas

---

2- JPA est :

**a- Une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java**

b- Une spécification d'Hibernate permettant le mapping des objets vers la base de données

**c- Une spécification d'ORM (Object Relational Mapping) pour Java**

d- Je ne sais pas

---

3- Quels sont les états possibles d'un objet en Hibernate ?

**a- Transcient, Persistent, detached**

b- Killed, Transcient, Persistent

c- Persistent, detached, Killed

d- Je ne sais pas

---

4- Le "mapping" d'une colonne avec un attribut de la classe POJO se fait :

**a- avec @Column("nom de colonne")**

b- avec @Entity("nom de colonne")

c- avec @Transient("nom de colonne")

d- Je ne sais pas

---

5- Comment crée-t-on une session en Hibernate ?

a- A l'aide d'un POJO

b- Avec SessionManager

**c- Avec Configuration.buildSessionFactory(...);**

d- Avec SessionCreator

e- Je ne sais pas

6- Comment peut-on réaliser un “mapping” en Hibernate ?

**a- Avec un fichier XML**

b- Avec un fichier JSON

**c- Avec des annotations @**

d- Je ne sais pas

---

7- Comment logger des requêtes SQL ?

a- Dans la configuration avec `hibernate.show_sql = true`

(je ne le mets pas car la syntaxe est plutôt celle-ci : `<property name="hibernate.show_sql">true</property>`)

**b- Par programmation, avec `configuration.configure().setProperty("hibernate.show_sql", "true")` ;**

c- Dans la configuration avec `show_sql_hibernate = true`

d- Je ne sais pas

---

8- Qu’est-ce qui permet de représenter une relation 1 à N en Hibernate ?

**a- @OneToMany**

b- <One-To-Many>

c- @ManyToMany

**d- @ManyToOne** (dans le cas d’une relation bidirectionnelle , nous en avons besoin)

e- Je ne sais pas

---

9- Quelle stratégie n’est pas une stratégie de “mapping d’héritage”?

**a- Table per hierarchy**

**b- Table per subclass**

c- Table per class

(c’est bien une stratégie de “mapping d’héritage”, mais sur notre code, on l’écrit `TABLE_PER_CLASS`)

d- Je ne sais pas

Remarque : En gras, ce sont les réponses du QCM !