

更多银行笔试面试经验请看:

<https://gitbook.cn/gitchat/activity/5d89b272ba2e817ead164ada>

- 1) 在三层结构中, 数据访问层承担的责任是 (B) [选一项]
 - a) 定义实体类
 - b) 数据的增删改查操作
 - c) 业务逻辑的描述
 - d) 页面展示和控制转发
- 2) 下面关于数据持久化概念的描述, 错误的是 (D [选择一项])
 - a) 保存在内存中数据的状态是瞬时状态
 - b) 持久状态的数据在关机后数据依然存在
 - c) 数据可以由持久状态转换为瞬时状态
 - d) 将数据转换为持久状态的机制称为数据持久化
- 3) 下面关于 Hibernate 的说法, 错误的是 (C) [选择一项]
 - a) Hibernate 是一个“对象-关系映射”的实现
 - b) Hibernate 是一种数据持久化技术
 - c) Hibernate 是 JDBC 的替代技术
 - d) 使用 Hibernate 可以简化持久化层的编码
- 4) Hibernate 配置文件中, 不包含下面的 (AB) [选择二项]
 - a) “对象-关系映射”信息
 - b) 实体间关联的配置
 - c) show_sql 等参数的配置
 - d) 数据库连接信息
- 5) 在 Hibernate 关系映射配置中, inverse 属性的含义是 (B)。[选择一项]
 - a) 定义在<one-to-many>节点上, 声明要负责关联的维护
 - b) 声明在<set>节点上, 声明要对方负责关联的维护
 - c) 定义在<one-to-many>节点上, 声明对方要负责关联的维护
 - d) 声明在<set>节点上, 声明要负责关联的维护
- 6) 在使用了 Hibernate 的系统中, 要想在删除某个客户数据的同时删除该客户对应的所有订单数据, 下面方法可行的是 (B)。[选择一项]

- a) 配置客户和订单关联的 cascade 属性为 save-update
 - b) 配置客户和订单关联的 cascade 属性为 all
 - c) 设置多对一关联的 inverse 属性为 true
 - d) 设置多对一关联的 inverse 属性为 false
- 7) 下面一对多关联的配置中, 存在哪些错误 (BD)。
- ```
<set name="orders" inverse="false" cascade="true">
 <one-to-many class="com.scce.pojo.Order"
 column="customer_id" />
</set>
```
- a) inverse 属性不可以设置为 false
  - b) cascade 属性不可设置为 true
  - c) class="com.scce.pojo.Order"应为 type="com.scce.pojo.Order"
  - d) column 属性不可再<one-to-many>节点内设置
- 8) 执行下面的 hql 查询, 关于返回值的说法, 正确的是 (B)。
- ```
select    u.username,u.userid  
from      com.scce.pojo.User u  
where     u is not null
```
- a) 语法错误, 不能执行
 - b) 返回一个类型为 List 的实例
 - c) 返回一个数组
 - d) 当返回结果只有一条时, 返回一个数组
- 9) 下面创建 Criteria 对象的语句中正确的是 (D)。
- a) Criteria c = query.createCriteria();
 - b) Criteria c = query.addCriteria();
 - c) Criteria c = session.createCriteria();
 - d) Criteria c = session.createCriteria(User.class);
- 10) 一般情况下, 关系数据模型与对象模型之间有哪些匹配关系 (ACD 多选)
- a) 表对应类
 - b) 记录对应对象
 - c) 表的字段对应类的属性
 - d) 表之间的参考关系对应类之间的依赖关系
- 11) 以下关于 SessionFactory 的说法哪些正确? BC
- a) 对于每个数据库事务, 应该创建一个 SessionFactory 对象
 - b) 一个 SessionFactory 对象对应一个数据库存储源。

更多银行笔试面试经验请看: <https://gitbook.cn/gitchat/activity/5d89b272ba2e817ead164ada>

- c) SessionFactory 是重量级的对象, 不应该随意创建。如果系统中只有一个数据库存储源, 只需要创建一个。
 - d) SessionFactory 的 load()方法用于加载持久化对象
- 12) Customer 类中有一个 Set 类型的 orders 属性, 用来存放 Order 订单对象, 在 Customer.hbm.xml 文件中, 用哪个元素映射 orders 属性? AB
- a) <set>
 - b) <one-to-many>
 - c) <many-to-one>
 - d) <property>
- 13) <set>元素有一个 cascade 属性, 如果希望 Hibernate 级联保存集合中的对象, cascade 属性应该取什么值? (D) [选择一项]
- a) none
 - b) save
 - c) delete
 - d) save-update
- 14) 以下哪些属于 Session 的方法? ABD
- a) load()
 - b) save()
 - c) open()
 - d) saveOrUpdate()
- 16) 以下程序的打印结果是什么? C
- ```
tx = session.beginTransaction();
Customer c1=(Customer)session.load(Customer.class,new Long(1));
Customer c2=(Customer)session.load(Customer.class,new Long(1));
System.out.println(c1==c2);
tx.commit();
session.close();
```
- a) 运行出错, 抛出异常
  - b) 打印 false
  - c) 打印 true
  - d) 编译出错

- 17) 以下程序代码对 Customer 的 name 属性修改了两次: B

```
tx = session.beginTransaction();
Customer customer=(Customer)session.load(Customer.class,
 new Long(1));
customer.setName(\"Jack\");
customer.setName(\"Mike\");
tx.commit();
```

执行以上程序, Hibernate 需要向数据库提交几条 update 语句? (单选)

- a) 0
- b) 1
- c) 2
- d) 3

- 18) 在持久化层, 对象分为哪些状态? (ACD 多选)

- a) 瞬时(Transient)
- b) 新建(New)
- c) 游离(Detached)
- d) 持久(Persistent)

- 19) 对于以下程序, Customer 对象在第几行变为持久化状态? (B 单选)

```
Customer customer=new Customer();
customer.setName(\"Tom\");
Session session1=sessionFactory.openSession();
Transaction tx1 = session1.beginTransaction(); //line1
session1.save(customer); //line2
tx1.commit(); //line3
session1.close(); //line4
```

- a) line1
- b) line2
- c) line3
- d) line4

- 20) 对于以下程序, Customer 对象在第几行变为游离状态? (D 单选)

```
Customer customer=new Customer();
customer.setName(\"Tom\");
Session session1=sessionFactory.openSession();
Transaction tx1 = session1.beginTransaction(); //line1
session1.save(customer); //line2
tx1.commit(); //line3
session1.close(); //line4
```

- a) line1
- b) line2
- c) line3
- d) line4

21) 以下哪一种检索策略利用了外连结查询? (C 单选)

- a) 立即检索
- b) 延迟检索
- c) 迫切左外连结检索
- d) 迫切右外连结检索

22) 假设对 Customer 类的 orders 集合采用延迟检索策略, 编译或运行以下程序, 会出现什么情况 (D 单选)

```
Session session=sessionFactory.openSession();
tx = session.beginTransaction();
Customer customer=(Customer)session.get(Customer.class,new Long(1));
tx.commit();
session.close();
Iterator orderIterator=customer.getOrders().iterator();
```

- a) 编译出错
- b) 编译通过, 正常运行, 不显示任何信息
- c) 编译通过, 正常运行, 并显示出结果
- d) 编译通过, 但运行时抛出异常

23) POJO 是 A

- a) Plain Old Java Object
- b) Programming Object Java Object
- c) Page Old Java Object
- d) Plain Object Java Old

24) POJO 的作用是 AC

- a) 普通的 Java 对象映射数据库字段
- b) 数据访问
- c) 对象的赋值
- d) 保存客户端对象

25) 实现 equals 和 hashCode 最显而易见的作用是 A

- a) 比较两个对象标识符的值
- b) 比较数据库的两条数据

更多银行笔试面试经验请看: <https://gitbook.cn/gitchat/activity/5d89b272ba2e817ead164ada>

- c) 比较两个组建是否部署在同一个应用服务器上
  - d) 对象与字段的映射
- 26) 持久化状态前进行合法性检查, 需要实现的接口是 B
- a) ActionListener
  - b) Lifecycle
  - c) HttpServletRequest
  - d) 持久化状态合法检查不需要实现接口
- 27) 在源代码中嵌入映射信息需要 A
- a) XDoclet
  - b) EJB
  - c) Hibernate
  - d) Spring
- 28) 所有的 Hibernate, 除了\_\_\_\_\_了都支持空(null)语义。A
- a) Collection
  - b) Element
  - c) Entity
  - d) NodeList
- 30) 元素把子对象的一些元素与父类对应的表的一些字段映射起来 A
- a) Component
  - b) Element
  - c) Attribute
  - d) ActionForm
- 32) ORM 是 A
- a) 对象关系映射
  - b) 对象实体映射
  - c) 数据传输对象
  - d) 数据访问对象
- 33) JDBC 可以做哪三件事 ABC
- a) 与数据库建立连接
  - b) 通过 JDBC-API 向数据库发送 SQL 语句
  - c) 通过 JDBC-API 执行 SQL 语句
  - d) 进行实体关系的映射
- 34) ORM 有什么好处 A

- a) 能进行关系对象的映射
  - b) 具有控制器功能
  - c) 能在服务器端保存客户端状态
  - d) 向 DTO 一项能传递数据
- 35) 常见的实体域对象是 AB
- a) 主动域对象(EJB)
  - b) JavaBean 对象(也称为 POJO, Plain Old Java Object)
  - c) 控制器对象
  - d) ActionForm
- 37) 常见的三层分别是 ABC
- a) 应用层
  - b) 业务逻辑层
  - c) 数据访问层
  - d) 通讯层
- 38) 对象(实体的实例)对一个特定的 Session 来说,要么是一个\_\_\_对象,要么是\_\_\_对象 AB
- a) 瞬时
  - b) 持久化
  - c) 控制器
  - d) 数据传输对象
- 39) 如果你已知某个实例的持久化标识,你可以使用 Session 的\_\_\_方法来获取它 A
- a) load()
  - b) loadAll()
  - c) getAllData()
  - d) init()
- 40) 如果你不能确定你要寻找的对象的持久化标识符,那么你需要使用查询,使用 Session 的\_\_\_方法 A
- a) createQuery()
  - b) query()
  - c) queryAll()
  - d) queryObject()
- 41) 使用 Session 的\_\_\_会把对象的状态从数据库中移除 A
- a) delete()
  - b) remove()
  - c) clear()
  - d) removeAll()

43) HQL 不支持聚合函数有 D

- a) count(\*)
- b) avg(...)
- c) count(...)
- d) count(?)

44) \_\_\_\_和\_\_\_\_子句都不支持算数表达式。AB

- a) group by
- b) order by
- c) compute by
- d) insert by

45) 除了\_\_\_\_和\_\_\_\_外, 查询语句都是大小写不敏感得到。AD

- a) Java 类
- b) POJO 类
- c) 映射文件
- d) 属性名称

46) from Cat as cat,不仅仅返回\_\_\_\_的实例, 而且还返回\_\_\_\_的实例 AC

- a) Cat
- b) Cat 父类
- c) Cat 子类
- d) Cat 实现接口

47) \_\_\_\_代表升序, \_\_\_\_代表降序 AC

- a) asc
- b) ansi
- c) desc
- d) description

48) 对原生 SQL 查询的控制是通过\_\_\_\_接口进行的。A

- a) SQLQuery
- b) Session
- c) List
- d) Query



- 51) 假设实体映射时有一个\_\_\_\_的关联指向另外一个实体,在查询时必须也返回那个实体,否则会导致发生一个"column not found"的数据库错误。D
- a) many-to-one
  - b) many-to-many
  - c) one-to-many
  - d) one-to-one
- 54) 在 Hibernate 中,关于数据库的方言,下面哪个说法是正确的 D
- a) 可以不写数据库方言
  - b) 数据库方言没有作用,可以随便写
  - c) 有通用的数据库方言可以使用
  - d) 数据方言对应于每个数据库,主要用于产生 SQL 语句
- 55) 下面不属于持久化的是 A
- a) 把对象转换成字符串的形式通过网络传输,在另一端接收到这个字符串后能把对象还原出来
  - b) 把程序数据从数据库中读出来
  - c) 从 XML 配置文件中读取程序的配置信息
  - d) 把程序数据保存为文件
- 56) 下面不是 Hibernate 映射文件中包含的内容。(AB 选两项)
- a) 数据库连接信息
  - b) Hibernate 属性参数
  - c) 主键生成策略
  - d) 属性数据类型

- 57) 下面的程序执行后没有报错, 但数据总保存不到数据库, 最可能的原因是 E

```
public static void imain(String[] args){
 SessionFactory sf=new Configuration().configure().buildSessionFactory();
 Session session=sf.openSession();
 Medal medal=new Medal();
 medal.setOwner("totong");
 medal.setType("Gold medal");
 session.save(user);
 session.close();
}
```

- a) 配置文件配置有误
- b) 没有在配置文件中包含对映射文件的声明
- c) 配置文件配置有误
- d) 没有对持久化操作捕捉异常
- e) 没有开启事务

- 58) 下列属于多对一关系的是\_D

- a) 书和作者
- b) 手机和生产厂家
- c) 用户和发布的出租信息
- d) 士兵和班长

- 59) 已知 `wage.getItem().size()` 的值是 1。要想使执行 `session.update(wage)` 时自动执行 `session.update(wage.getItems().get(0))`, 必须配置 D

- a) `<set name="items" inverse="true" cascade="none">...</set>`
- b) `<set name="items" inverse="false" cascade="none">...</set>`
- c) `<set name="items" inverse="true" cascade="all">...</set>`
- d) `<set name="items" inverse="false" cascade="save-update">...</set>`

- 60) 已知系统中的 `tbl_user` 表对应的实体类是 `User`, 下列 HQL 语句中正确的是(AD 选两项)

- a) `from User`
- b) `select * from tbluser`
- c) `select User from User`
- d) `from User u where u.uname="15"`