

学号	姓名	提交时间	总分	成绩
20225102050116	夏浩然	-	100	-

1. 【单选题】 下列哪种方法用于创建PreparedStatement对象 得分：0/10
- A. 通过Connection对象的prepareStatement()方法
- B. 通过Statement对象的createPreparedStatement()方法
- C. 通过PreparedStatement类的新Instance()方法
- D. 通过ResultSet对象的prepareStatement()方法
- 学生答案：未作答
参考答案：A
2. 【单选题】 在使用PreparedStatement创建对象时，下列哪种描述是正确的 得分：0/10
- A. PreparedStatement对象创建后可以直接执行SQL语句
- B. PreparedStatement对象创建后需要手动指定SQL语句
- C. PreparedStatement对象创建时需要传入SQL语句和参数占位符
- D. PreparedStatement对象创建时不需要连接数据库
- 学生答案：未作答
参考答案：C
3. 【单选题】 使用PreparedStatement进行动态参数传递时，下列哪种方法是正确的？ 得分：0/10
- A. 使用字符串拼接将参数值直接嵌入SQL语句中
- B. 使用getString()、getInt()等方法设置参数值
- C. 在SQL语句中使用问号(?)作为占位符，然后使用setXXX()方法设置参数值
- D. 在SQL语句中使用花括号({})将参数值括起来，然后使用setParameter()方法设置参数值
- 学生答案：未作答
参考答案：C
4. 【单选题】 下列哪个说法正确描述了PreparedStatement和Statement的区别？ 得分：0/10
- A. PreparedStatement适用于静态SQL语句，而Statement适用于动态SQL语句。
- B. PreparedStatement在编译阶段预编译SQL语句，而Statement在执行阶段编译SQL语句。
- C. PreparedStatement可以防止SQL注入攻击，而Statement容易受到SQL注入攻击。
- D. PreparedStatement可以同时执行多条SQL语句，而Statement只能执行单条SQL语句。
- 学生答案：未作答
参考答案：C
5. 【单选题】 PreparedStatement与Statement相比，以下哪一项是它的优点？ 得分：0/10
- A. 更安全，能够防止SQL注入攻击
- B. 执行速度比较慢
- C. 可以执行任意SQL语句
- D. 更简单易用
- 学生答案：未作答
参考答案：A
6. 【单选题】 JDBC 是 Java Database Connectivity 的缩写，它是用来做什么的？ 得分：0/10
- A. 连接 Java 应用程序和操作系统
- B. 连接 Java 应用程序和文件系统
- C. 连接 Java 应用程序和网络
- D. 连接 Java 应用程序和数据库
- 学生答案：未作答
参考答案：D
7. 【单选题】 在 JDBC 中，通过哪个接口可以执行 SQL 语句并返回结果集？ 得分：0/10
- A. Connection
- B. DriverManager
- C. ResultSet
- D. Statement/PreparedStatement

学生答案：未作答

参考答案：D

8. 【单选题】 JDBC 中的 ResultSet 对象可以用来做什么？

得分：0/10

- A. 连接到数据库并执行事务
- B. 执行数据库的备份和恢复操作
- C. 执行数据库的元数据操作
- D. 执行 SQL 查询并返回结果

学生答案：未作答

参考答案：D

9. 【单选题】 JDBC 中的 SQL 注入攻击是指什么？

得分：0/10

- A. 攻击者通过修改数据库的元数据来获取数据
- B. 攻击者通过发送恶意的 SQL 语句来获取或修改数据库的数据
- C. 攻击者通过修改 JDBC 配置文件来篡改数据
- D. 攻击者通过关闭数据库连接来阻止正常访问

学生答案：未作答

参考答案：B

10. 【单选题】 在 JDBC 中，如何连接到数据库？

得分：0/10

- A. 使用 DriverManager 类的 registerDriver() 方法
- B. 使用 Statement 对象的 execute() 方法
- C. 使用 Connection 对象的 connect() 方法
- D. 使用 ResultSet 对象的 getMetaData() 方法

学生答案：未作答

参考答案：C