

《使用 Java 实现面向对象编程》阶段测试-笔试试卷

一、选择题(共 25 题，每题 4 分，满分 100 分)

1) 下列选项中关于 Java 中 ArrayList、LinkedList、HashMap 三个类的描述错误的是(C)。

- A. ArrayList 和 LinkedList 均实现了 List 接口
- B. ArrayList 的访问速度比 LinkedList 快
- C. 添加和删除元素时，ArrayList 的表现更佳
- D. HashMap 实现 Map 接口，它允许任何类型的键和值对象，允许将 null 用作键或值

2) 关于 Java 中的集合实现类，下列说法描述错误的是 (D)。

- A. HashMap 是 Map 接口的实现类
- B. ArrayList 对象是长度可变的对象引用数组
- C. 集合框架都包含三大块内容：对外的接口、接口的实现和对集合运算的算法
- D. Set 中存储一组不允许重复、有序的对象

3) 对于 Java 中的集合类 java.util.ArrayList，以下说法正确的是 (A)。

- A. 这个集合中的元素是无序的
- B. 可以通过键来获取这个集合中的元素
- C. 在对这个集合的元素执行遍历操作时，效率比较高
- D. 可以通过 addFirst()方法，在列表的首部插入元素

4) 给定如下 Java 代码，其中第 (D) 行代码有错误。

```
import java.util.*;
public class MapTest {
    public static void main(String[] args) {
        List pList = new ArrayList();
        pList.add("A"); //第五行
        pList.add("B"); //第六行
        for (int i = 0; i < pList.size(); i++) { //第七行
            String str = pList.get(i); //第八行
            System.out.print(str);
        }
    }
}
```

- A. 五
- B. 六
- C. 七
- D. 八

5) 给定如下 Java 代码，可以填入横线处的代码是（ D ）。

```
import java.util.*;  
public class Test {  
    public static void main(String[] args) {  
  
        _____  
        list.addLast("001");  
    }  
}
```

- A. List list = new ArrayList();
- B. List list = new List();
- C. ArrayList list = new ArrayList();
- D. LinkedList list = new LinkedList();

6) 在 Java 中，（ A ）类的对象可被用作堆栈或者队列。

- A. LinkedList
- B. ArrayList
- C. Collection
- D. HashMap

7) 关于 Threads 哪些描述是**正确**的（ B ）

- A. 线程可以创建唯一的子类 java.lang.Thread。
- B. 调用 suspend() 方法可以使线程中止并且无法再启动它。
- C. 程序的执行完毕是以用户线程的结束来标志的，与超级线程无关。
- D. 不同线程对相同数据进行访问时，可能造成数据毁损。

8) 下面哪个选项可以正确地建立文件 file.txt?(C)

- A. File f = new File("D:\text", "file.txt");
- B. File f = new File("D:\\text", "file.txt");

C. File f = new File("file.txt", "D:/text"); D. File f = new File("D:/text", "file.txt");
--

9) 在 Java 中, JDBC API 定义了一组用于与数据库进行通信的接口和类, 它们包括在 (B) 包中。

- A. java.lang
- B. java.sql
- C. java.util
- D. java.math

10) 使用 JDBC 访问数据库时, 下列关于 JDBC 的说法正确的是 (B)。

- A. Connection 接口负责执行 SQL 语句
- B. 可以使用 Class.forName()方法把 JDBC 驱动类装载入 Java 虚拟机中
- C. Statement 接口由 Connection 产生负责保存返回的查询结果
- D. DriverManager 类负责连接数据库并传送数据

11) 下面哪个选项可以正确地建立文件 file.txt?(A)

E. File f = new File("D:\text", "file.txt"); F. File f = new File("D:\\text", "file.txt"); G. File f = new File("file.txt", "D:/text"); H. File f = new File("D:/text", "file.txt");

12) 在 Java 中开发 JDBC 应用程序时, 使用 DriverManager 类的 getConnection()方法建立与数据源的连接语句为:

```
Connection con =  
    DriverManager.getConnection("jdbc:microsoft:sqlserver://localhost:1433;  
    DatabaseName=news");
```

URL 连接中的 “news” 表示的是 (B)。

- A. 用户名
- B. 数据库的名称
- C. 数据库服务器的机器名

D. 数据库中表的名称

13) 在 Java 中，以下关于得到 ResultSet 对象、PreparedStatement 对象、Statement 对象的说法，正确的是（ AC ）。

- A. 可以使用 Statement 对象的 executeQuery 方法返回 ResultSet 对象
- B. 可以使用 Connection 对象的 createStatement 方法获得 PreparedStatement 对象
- C. 可以使用 Connection 对象的 createStatement 方法获得 Statement 对象
- D. 可以使用 Connection 对象的 executeQuery 方法返回 ResultSet 对象

14) 在使用 JDBC 访问数据库时，关于 Statement 和 PreparedStatement 的描述错误的是（ B ）。

- A. PreparedStatement 接口继承自 Statement 接口，因此 PreparedStatement 实例能够强制转换成 Statement 实例来使用
- B. 与 Statement 相比，使用 PreparedStatement 提高了 SQL 语句执行的性能，降低了安全性
- C. Statement 实例和 PreparedStatement 实例，均能对数据库进行增删改查操作
- D. Statement 实例和 PreparedStatement 实例，均能够通过“?”作为占位符，用于后面进行参数的设定

15) 假设 SQL Server 数据库中有表 book，包含两个字段：id 整型，title 字符型。向此表插入数据的 Java 代码片段如下所示，其中 con 为已正确获得的连接数据库的 Connection 对象，则以下描述正确的是（ C ）。

```
PreparedStatement pstmt=con.prepareStatement("insert into book(id, title)
values(?,?)");
pstmt.setInt(0,10);
pstmt.setString(1,"java");
pstmt.executeUpdate();
```

- A. 创建 PreparedStatement 对象的语句有错误
- B. 向 book 表插入数据的 sql 语句有错误
- C. setInt、setString 方法有错误
- D. PreparedStatement 没有 executeUpdate()方法

16) 给定一个 Java 程序的代码片段如下所示，其中 `conn` 为连接数据库的 `Connection` 对象，此处假定已经正确获得连接。在下划线处加入下列（ B ）代码，可以在控制台循环打印出 `student` 表中 `name` 数据列的值。

```
Statement stmt = conn.createStatement();
String query = "select * from student";
java.sql.ResultSet rs = stmt.executeQuery(query);
_____
```

- A. `while (rs.hasNext()) {`
 `System.out.println(rs.getString("name"));`
}
- B. `while (rs.next()) {`
 `System.out.println(rs.getString("name"));`
}
- C. `if(rs.next()) {`
 `System.out.println(rs.getString("name"));`
}
- D. `System.out.println(rs.getString("name"));`

17) 以下选项中关于 Oracle 中数据库的说法错误的是（ B ）。

- A. 数据库在逻辑上表现为数据文件、日志文件和控制文件等
- B. 必须首先创建数据库，然后才能使用 Oracle
- C. 可在安装 Oracle 软件时同时创建数据库，也可以在安装后再单独创建数据库
- D. 每个启动的数据库都对应一个数据库实例，然后由这个实例来访问和控制数据库

18) 使用数据库表 `student` 存放学生信息，其中 `birthday` 字段存放学生生日，查询该表中 1980 年 9 月 15 日及以后的学生信息，以下 SQL 语句正确的是（ C ）。

- A. `SELECT * FROM student WHERE birthday>='1980-09-15';`
- B. `SELECT * FROM student WHERE birthday >='15-9 月-1980';`
- C. `SELECT * FROM student WHERE TO_DATE(birthday,'YYYY-MM-DD')>='2009-10-1';`
- D. `SELECT * FROM student WHERE birthday>=TO_CHAR('1980-09-15','YYYY-MM-DD');`

19) 一个数据库表包含性别字段，在该字段上建立（ A ）最合适。

- A. 唯一索引

- B. 位图索引
- C. 基于函数的索引
- D. B 树索引

20) 以下选项中关于 DAO 模式的说法错误的是 (C)。

- A. DAO 是 “Data Access Object” 的含义，实现对数据库资源的访问
- B. DAO 模式中要定义 DAO 接口和实现类，隔离了不同数据库的实现
- C. DAO 负责执行业务逻辑操作，将业务逻辑和数据访问隔离开来
- D. 使用 DAO 模式提高了数据访问代码的复用性

21) 以下选项中关于 Java 分层开发技术中实体类的说法错误的是 (D)。

- A. 在分层开发中，不同层之间可以通过实体类来传输数据
- B. 实体类一般对应一个数据库表，其属性和数据库表的字段对应
- C. 实体类最好实现 `java.io.Serializable`，支持序列化机制
- D. 实体类的属性和方法一般均使用 `public` 修饰

22) 以下选项中关于 XML 文档的说法正确的是 (A)。

- A. XML 文档中使用用户自定义标签，标签名不区分大小写
- B. 可以使用 DTD 或者 XSD 定义 XML 文档结构
- C. XML 文档的作用是数据存储、数据交换、数据显示
- D. XML 文档总是以一个 XML 声明开始

23) 以下选项中关于 DOM 和 SAX 的说法错误的是 (C)。

- A. DOM 和 SAX 都是目前常用的 XML 解析技术
- B. DOM 把 XML 文档映射成一个倒挂的树状结构
- C. DOM 模型内存消耗小，DOM 解析器能提供更好的性能优势
- D. SAX 读取文档时会激活一系列事件，推给事件处理器，由事件处理器来访问文档

24) 以下选项中关于 Java 中文件输入输出的说法正确的是 (A)。

- A. 在 Java 中，文件的输入输出功能是通过流来实现的

- B. 如果要把数据写入到一个文件中，需要创建一个输入流对象
- C. 字符流在某些方面比如汉字的处理，比字节流更高效
- D. 可以通过 `Reader r=new Reader("c:\\pet.txt")`来创建一个输入流对象

25) 给定如下 Java 代码，实现文件复制功能，其中第（ C ）行代码有错误。

```
public class TestIO {  
    public static void main(String[] args) throws Exception {  
        Reader fr = new FileReader("c:\\source.txt"); //第三行  
        Writer fw = new FileWriter("c:\\target.txt");  
        char ch[] = new char[1024];  
        int length = fr.read(ch); //第六行  
        while (length != -1) {  
            fw.write(ch); //第八行  
            length = fr.read(ch);  
        }  
        if (null != fr)  
            fr.close(); //第十三行  
        if (null != fw)  
            fw.close();  
    }  
}
```

- A. 三
- B. 六
- C. 八
- D. 十三

