

## Java 试题

### 一. 填空题 (每个 1 分, 共 20 分)

1. Java 中可使用 throw 关键字来引发用户异常, 使用 try-catch-finally 结构捕获产生的异常。
2. 由于资源不足而阻塞线程方法是: wait(), 当资源释放唤醒阻塞线程的方法是 notify() 或 notifyAll()。
3. java 语言使用的 Unicode 字符集含 2<sup>16</sup> 个字符。
4. 给方法加上关键字 synchronized 表示 方法同步。
5. Java 数据类型中, long 是 8 (64 位) 字节, short 是 2 (16 位) 字节。
6. import 语句的作用是: 加载已定义好的类或包/简化类名, 除了 java.lang 包外, 其它包都需要显式引入。
7. Java.net 包中, 支持 http 协议的用户网页数据提交的类是 URLConnection。
8. java 类中的方法按属性可分为 类 方法和 实例 方法两种。
9. AWT 所有的组件和容器都从 Component 类派生。Applet 类的父类是: Panel。
10. 类变量除了可用对象引用访问外, 还可以通过 类名 来访问。
11. Java 中, 用 final 修饰的类则表示该类 不能被继承。
12. 可支持文件随机读写的流类是 RandomAccessFile, 支持套接口的客户端数据传递的类是 Socket, 服务端的类是 ServerSocket。

### 二. 选择题 (每个 1 分, 共 10 分)

1. 以下关于 abstract 的说法, 正确的是 **D**
    - A. abstract 只能修饰类
    - B. abstract 只能修饰方法
    - C. abstract 类中必须有 abstract 方法
    - D. abstract 方法所在的类必须用 abstract 修饰
  2. 以下关于类继承的说法, 正确的是 **C**
    - A. 子类中可见父类中的同名方法
    - B. 子类中可以直接使用父类的同名方法
    - C. 子类中使用 super 才能使用父类同名方法
    - D. 以上都不对
  3. 关于类继承的说法, 正确的是 **B**
    - A. Java 类允许多继承
    - B. Java 接口允许多继承
    - C. 接口和类都允许多继承
    - D. 接口和类都不允许多继承
  4. 关于接口的说法, 正确的是 **A**
    - A. 接口中的方法只能在接口的实现类中实现
    - B. 接口中可定义变量成员
    - C. 接口中不能定义常量
    - D. 以上都不对
-

5. 接口 A 中有 2 个方法，接口 B 中有两个方法，以下正确的是 **D**
- A. 若接口 C 继承 A,B 两个接口，则 C 一定继承了四个方法
- B. 若类 C 实现接口 A,B，则 C 中必须至少实现 A、B 中各一个方法
- C. 若类 C 要继承接口 A,B，则可以定义接口 D 来继承接口 A、B，由类 C 实现接口 D, 才能达到目的。

D. 以上都不对

6. AWT 中容器跟组件的关系是 **A**

**Component Container**

- A. 容器类也是一种特殊的组件类
- B. 组件类也是一种特殊的容器类
- C. 容器类与组件类没有关系
- D. 以上都不对

7. 以下类中属于 Container 的是 **D**

- A. List
- B. Canvas
- C. TextField
- D. Panel

8. 关于 Applet 类，正确的说法是 **B**

java.lang.Object

java.awt.Component

java.awt.Container

java.awt.Panel

**java.applet.Applet**

- A. Applet 类是 java.awt 中的类
-

---

B. Applet 类是 Container 的间接子类

C. Applet 类不是容器类

D. 以上都不对

9. 关于以下的组件的接口，正确的是 D

A. TextArea 组件对应的接口是 ActionListener

B. List 组件对应的接口是 ActionListener

C. Choice 组件对应的接口是 ActionListener

D. 以上都不对

10. 以下那种布局是 Applet 的默认布局 B

A. BorderLayout

B. FlowLayout

C. CardLayout

D. GridLayout

三. 判断说明(每题 2 分，共 20 分，用 T 和 F 表示正确和错误，并说明理由)

1. (T) Java 程序中没有全程函数，必须全部由类组成。

理由: Java 具有面向对象的特点，不兼容过程式计算机语言，面向对象性比 C++ 更彻底

2. (F) Java 程序在不同类型的 cpu 上执行，必须重新编译。

理由: Java 具有平台无关性的特点，编译一次就可以运行在任何操作系统上运行

3. (F) 一个类的方法，如果没有明确定义，则其访问限制缺省为 protected。

---

理由: 缺省为 Default

4. (F) Java 中的类和接口都不允许多继承。

理由: Java 的类不允许多继承，但接口允许多继承

5. (T) char 类型可以直接当成整数使用。

理由: char 数据类型比 int 数据类型弱，可以直接转换为 int 类型

6. (T) 当一个对象停止使用后，程序员可以不显式撤销这个对象。

理由: Java 的垃圾回收机制会自行处理无用的对象，释放内存

7. (T) Applet 本身是个容器，不允许再产生窗口对象。

理由: Applet 是顶层容器而且没有 main 方法

---

顶层容器:Frame Applet Dialog 以及 JFrame JApplet JDialog

8. (F)start 方法可以使线程立刻在物理机上执行。

理由: 只是把当前线程加入到线程队列中, 由 Java 虚拟机进行调度

1. 观察下面鼠标绘图代码解答下列问题

//程序代码

```
import java.applet.*
import java.awt.*;
import java.awt.event.*;
public class Example extends Applet implements MouseMotionListener
{ int x=-1,y=-1;
  public void init()
  { setBackground(Color.green) ;
    addMouseMotionListener(this);
  }
  public void mouseMoved(MouseEvent e) {}
  public void update(Graphics g) { paint(g); } //第 11 行
  public void paint(Graphics g)
  { if(x!=-1&&y!=-1)
    {g.setColor(Color.red);
     g.drawLine(x,y,x,y);}
  }
  public void mouseDragged(MouseEvent e)
  {x=(int)e.getX();
   y=(int)e.getY();
   repaint();
  }
}
```

//代码结束

1) MouseMotionListener 接口中处理的鼠标事件有几种, 分别是什么 (2 分)

答: 主要有鼠标拖动事件和移动鼠标事件。

mouseDragged(MouseEvent e) 和 mouseMove(MouseEvent e)

2)MouseListener 接口处理什么鼠标事件? 与本程序中实现的 mouseMotionListener 接口有什么区别? 是否可以在同一个类中同时实现这两个接口? (2 分)

点击, 移入, 移出, 按下, 释放

void	<a href="#"><code>mouseClicked(MouseEvent e)</code></a> Invoked when the mouse button has been clicked (pressed and released) on a component.
void	<a href="#"><code>mouseEntered(MouseEvent e)</code></a> Invoked when the mouse enters a component.
void	<a href="#"><code>mouseExited(MouseEvent e)</code></a> Invoked when the mouse exits a component.
void	<a href="#"><code>mousePressed(MouseEvent e)</code></a> Invoked when a mouse button has been pressed on a component.
void	<a href="#"><code>mouseReleased(MouseEvent e)</code></a> Invoked when a mouse button has been released on a component.

可以

3) 如果 11 行不重写 update 方法, 这个程序的运行结果有改变吗, 会有什么现象出现? (2 分)

答: 有, 当鼠标拖动时只绘画一小段线, 然后很快地消失了, 无法进行正常的绘画操作。

---

4) 本例中实际画出图形的语句是哪行? 它的功能有什么缺陷 (2 分)

答: 进行画图的语句是: `g.drawLine(x, y, x, y);`

该函数只根据鼠标所在坐标绘画单点, 当鼠标快速移动的时候, 绘制出的图形是不连续

5) 在鼠标快速移动的时候, 绘制出的图形还是不连续, 应该怎么改进? 请写出进一步代码 (7 分)

答: 应保存当前左标和新左标, 利用 `g.drawLine(x, y, old_x, old_y)` 进行绘画。

关键代码:

```
public class Example extends Applet implements MouseMotionListener
{
    int x=-1, y=-1;
    int m_x=-1, m_y=-1;
    public void init()
    {
        setBackground(Color.green);
        addMouseMotionListener(this);
    }
    public void mouseMoved(MouseEvent e) { x= y= m_x= m_y= -1; }
    public void update(Graphics g) { paint(g); }
    public void paint(Graphics g)
    {
        if(x!=-1&&y!=-1&&m_x!=-1&&m_y!=-1)
        {
            g.setColor(Color.red);
            g.drawLine(x, y, m_x, m_y);
        }
    }
    public void mouseDragged(MouseEvent e)
```

```

{   m_x=x;
    m_y=y;
    x=(int)e.getX();
    y=(int)e.getY();
    repaint();
}
}

```

#### 四. 综合题

4. (15 分)什么是 JDBC (1 分), 它有几中连接数据库的方式(2 分)请写出使用数据库的步骤及其代码(5 分)。如何实现模糊查询 (2 分), JDBC 在 internet 上使用有何缺陷, 如何克服, 举例说明 (5 分)。

答: P351~352, P358~363

JDBC 就是 Java Database Connectivity JAVA 数据库连接

有 4 种连接方式: 1、JDBC-ODBC 桥

2、由本地化方法实现 JDBC 的 API

3、通过中间件实现 JDBC 的 API

4、直接采用 java 语言实现与特定数据库交互的协议, 直接访问数据库

步骤: 1、创建数据源, 如果有需要的话要设置数据源的各项属性

2、创建命名服务上下文, 然后设置命名服务上下文的各项属性

3、在新创建的命名服务上下文中绑定新创建数据源和指定的名称

## P558 有代码

### SQL 语句实现模糊查询

文章分类:数据库

**SQL** 语句实现模糊查询, 有些东西老是既不清楚, 现在做个笔记吧。

我们可以在 **where** 子句中使用 **like** 来达到模糊查询的效果; 在 **Where** 子句中, 可以对 **datetime**、**char**、**varchar** 字段类型的列用 **Like** 子句配合通配符选取那些“很像...”的数据记录, 以下是可使用的通配符:

**%** 零或者多个字符

**\_** 单一任何字符 (下划线)

**\** 特殊字符

**[]** 在某一范围内的字符, 如[0-9]或者[aeth]

**[^]** 不在某范围内的字符, 如[^0-9]或者[^aeth]

后两种, 需要 **Oracle 10g** 以上使用支持 **like** 的正则 **regexp\_like**。

---

## 其中关于条件，Oracle 中 SQL 语句提供了四种匹配模式：

1, %: 表示任意 0 个或多个字符。可匹配任意类型和长度的字符，有些情况下若是中文，请使用两个百分号（%%）表示。

比如 `SELECT * FROM [user] WHERE u_name LIKE '%三%'`

将会把 `u_name` 为“张三”，“张猫三”、“三脚猫”，“唐三藏”等等有“三”的记录全找出来。

另外，如果需要找出 `u_name` 中既有“三”又有“猫”的记录，请使用 `and` 条件

`SELECT * FROM [user] WHERE u_name LIKE '%三%' AND u_name LIKE '%猫%'`

若使用 `SELECT * FROM [user] WHERE u_name LIKE '%三%猫%'`

虽然能搜索出“三脚猫”，但不能搜索出符合条件的“张猫三”。

2, \_: 表示任意单个字符。匹配单个任意字符，它常用来限制表达式的字符长度语句：

比如 `SELECT * FROM [user] WHERE u_name LIKE ' _三 _'`

只找出“唐三藏”这样 `u_name` 为三个字且中间一个是“三”的；

再比如 `SELECT * FROM [user] WHERE u_name LIKE '三 _ _'`；

只找出“三脚猫”这样 `name` 为三个字且第一个字是“三”的；

3, []: 表示括号内所列字符中的一个（类似正则表达式）。指定一个字符、字符串或范围，要求所匹配对象为它们中的任一个。

比如 `SELECT * FROM [user] WHERE u_name LIKE '[张李王]三'`

---

将找出“张三”、“李三”、“王三”（而不是“张李王三”）；

如 [] 内有一系列字符（01234、abcde 之类的）则可略写为“0-4”、“a-e”

```
SELECT * FROM [user] WHERE u_name LIKE '老[1-9]'
```

将找出“老 1”、“老 2”、.....、“老 9”；

4. [^]：表示不在括号所列之内的单个字符。其取值和 [] 相同，但它要求所匹配对象为指定字符以外的任一个字符。

比如 SQL Server: SELECT \* FROM [user] WHERE u\_name LIKE '[^张李王]三' 将找出不姓“张”、“李”、“王”的“赵三”、“孙三”等；

Oracle 10g 以上的用法为：

```
select * from table where regexp_like(name, '[^张李王]三');
```

```
SELECT * FROM [user] WHERE u_name LIKE '老[^1-4]';
```

将排除“老 1”到“老 4”，寻找“老 5”、“老 6”、.....

注：oracle like 不支持正则，你可以使用支持 like 的正则 regexp\_like

## 5. 查询内容包含通配符时

由于通配符的缘故，导致我们查询特殊字符“%”、“\_”、“[]”的语句无法正常实现，而把特殊字符用“[]”括起便可正常查询。据此我们写出以下

函数：

```
function sqlencode(str)
```

```
str=replace(str,"[","[]") '此句一定要在最前
```

```
str=replace(str,"_","_[]")
```

```
str=replace(str,"%","[%]")
```

```
sqlencode=str
```

```
end function
```



1、什么是类方法（静态方法）？什么是实例方法？它们的主要区别是什么？

答：方法声明时，方法前面不加 `static` 关键字的是实例方法，加 `static` 关键字的是类方法（静态方法）。

当类的字节码文件被加载到内存时，实例方法不会被分配入口地址，当该类创建对象后，实例方法能分配入口地址，对于类中的类方法，在该类加载到内存时，即分配入口地址。

实例方法不能用类名调用，类方法可以用类名调用。

实例方法可以操作实例变量也可以操作类变量，类方法只能操作类变量。

2、Java 从 JDK1.1 开始引入了基于授权的事件模型，简述其事件处理过程。

以文本框上的 `ActionEvent` 为例说明

1) 激发事件源，系统自动创建一个 `ActionEvent` 对象

2) 发生 `ActionEvent` 的事件源获得监听器

3) 处理发生 `ActionEvent` 事件的接口 `ActionListener`，创建文本框的监听器类实现 `ActionListener` 接口，系统将 `ActionEvent` 类自动创建的对象传递给方法 `actionPerformed (ActionEvent e)` 中的参数 `e`，监听器将自动调用方法 `actionPerformed (ActionEvent e)` 对发生的事件作出处理。

3、类的封装性、继承性和多态性各自的内涵是什么？

面向对象编程的核心思想之一就是将数据和对数据的操作封装在一起，通过抽象，即从具体的实例中抽取共同的性质形成一般的概念。

继承体现了一种先进的编程模式。子类可以继承父类的属性和功能，即子类继承了父类所具有的数据和数据上的操作，同时又可以增添子类独有的数据和数据上的操作。

多态是面向对象的又一重要特征。有两种意义的多态，一种是操作名称的多态，即有多个操作具有相同的名字，但这些操作所接收的消息类型必须不同。另一种多态是和继承有关的多态，是指同一操作被不同类型对象调用时可能产生不同的行为。

4、布局管理器的作用是什么？在 JDK 中哪些常用布局管理器？

布局管理器的作用是控制添加到容器中的组件在容器的位置。

常用的布局管理器有：`FlowLayout` 布局，`BorderLayout` 布局，`CardLayout` 布局，`GridLayout` 布局和 `BoxLayout` 布局。

（具体各个布局的特点见教程）

5、简述建立数据库连接的步骤。