

一、单项选择题

- 1、在基于 TCP 的 Client/Server 模型中，当 Server 端调用了 `accept()` 方法后，该进程将进入（ C ）状态。
A、运行 B、就绪
C、阻塞 D、死亡
- 2、有一个程序员在设计一个算法时需要用到某种集合类型，主要有两个要求，一是集合中的各个元素是有先后顺序的（ordered）；二是算法会经常执行 `add(0, object)` 操作，但很少进行集合元素的随机访问。请问这个程序员应该选用哪一种集合类型？（ B ）
A、ArrayList B、LinkedList
C、HashSet D、TreeSet

- 3、给定下列程序

```
public class Threads3 implements Runnable {  
    public void run() {  
        System.out.print("running");  
    }  
    public static void main(String[] args) {  
        Thread t = new Thread(new Threads3());  
        t.run();  
        t.run();  
        t.start();  
    }  
}
```

该程序的输出结果是（ E ）

- A. 编译错误 B、在运行时抛出一个异常 C. running
D、runningrunning E、runningrunningrunning
- 4、假设当前目录为空，即在当前目录下没有任何文件或子目录，且用户对当前目录具有读、写权限。给定以下程序

```
import java.io.*;  
public class Maker {  
    public static void main(String[] args) {  
        File dir = new File("dir"); // 注释：构造函数的一种写法：File(String pathname);  
        File f = new File(dir, "f"); // 注释：构造函数的另一种写法：File(File dir, String name);  
    }  
}
```

当上述程序运行完后，以下描述中正确的是（ B ）

- A、编译错误
B、文件系统没有添加任何新的内容
C、在文件系统中，只创建了一个新的文件
D、在文件系统中，只创建了一个新的目录
E、在文件系统中，创建了一个新的目录和一个新的文件

二、填空题

- 1、写出下列 Java 表达式的运算结果：

$5.0 / 4 - 4 / 5$ 1.25
"B" + 8 + 4 "B84"

- 2、String[] names = new String[5];

上述语句执行完后，将会在堆（heap）空间中创建 0 个 String 类的对象。

- 3、写出下列程序片断的输出结果：

```
class Atom {  
    Atom() { System.out.print("atom "); }  
}  
class Rock extends Atom {  
    Rock(String type) { System.out.print(type); }  
}  
public class Mountain extends Rock {  
    Mountain() {  
        super("granite ");  
        new Rock("granite ");  
    }  
}  
public class Test {  
    public static void main(String[] a) {  
        new Mountain();  
    }  
}
```

输出结果是： atom granite atom granite

三、代码分析

- 1、阅读下列程序：

```
import java.util.*;  
public class ReferenceMystery {  
    public static void main(String[] args) {  
        int x = 4;  
        int y = 8;  
        int[] data = {5, 10, 15};  
        x = mystery1(y, data);  
        // 注释： Arrays.toString(arr)的功能是对于数组 arr，返回一个 String，如： "[10, 30, 17]"  
        System.out.println(y + " " + Arrays.toString(data));  
        mystery2(x, y);  
        System.out.println(x + " " + y);  
    }  
    public static int mystery1(int n, int[] numbers) {  
        n = n + 100;  
    }
```

```

        numbers[2]--;
        System.out.println(n + " " + Arrays.toString(numbers));
        return numbers[1] * 2;
    }
    public static void mystery2(int x, int y) {
        x++;
        y = y * 3;
    }
}

```

上面这个程序的输出结果是：

108 [5, 10, 14]

8 [5, 10, 14]

20 8

2、假设定义了如下几个类：

```

public class Foo {
    public void method1() {
        System.out.println("foo 1");
    }
    public void method2() {
        System.out.println("foo 2");
    }
    public String toString() {
        return "foo";
    }
}
public class Bar extends Foo {
    public void method2() {
        System.out.println("bar 2");
    }
}

```

```

public class Baz extends Foo {
    public void method1() {
        System.out.println("baz 1");
    }
    public String toString() {
        return "baz";
    }
}
public class Mumble extends Baz {
    public void method2() {
        System.out.println("mumble 2");
    }
}

```

然后又有如下的代码片段：

```

Foo[] pity = {new Baz(), new Bar(), new Mumble(), new Foo()};
for (int i = 0; i < pity.length; i++) {
    System.out.println(pity[i]);
    pity[i].method1();
    pity[i].method2();
    System.out.println();
}

```

请问：上述代码片的输出结果是什么？

baz
baz 1
foo 2

foo
foo 1
bar 2

baz
baz 1
mumble 2

foo
foo 1
foo 2