题目：java.lang： 包含执行与网络有关的类，如 URL，SCOKET，SEVERSOCKET
解析：java.awt 提供了绘图和图像类，主要用于别写 gui 程序，包括按钮、标签等常用的组件及相应的事件类；

java.lang 是 java 的语言包是核心包，默认导入到用户程序，包括了 object 类，数据类型包装类，数学类，字符串类，运行时类操作类，线程类，错误和异常处理类，过程类；
javaio 包含了多种输入输出功能；
javanet 包含了执行网络有关的类，如 url，socket 和 serversocket
javaapplet 包含了 java 小应用程序的类；
javautil 包含了集合框架，遗留的 collection，事件模型等使用的工具类
javasql 包含了 javatm 编程语言并处理存储在数据源中的数据 api

题目：JVM 内存不包含如下哪个部分 heap frame
原因：java 中只有栈帧没有堆帧，另外 Pc 寄存器也可以称作是程序计数器

题目：下列语句中，正确的是 float x=0.0 和 boolean b=3>5
原因：java 中的 float 类型中的数据类型初始化时必须使用后缀 f，因为 java 默认的浮点型是 double；boolean 可以是一个表达式的值，但必须是一个 true 或者是 false 值；

题目：容器 panel 缺省使用的布局编辑策略是 Flow Layout
原因：GUI PANEL 容器以及布局管理器 Panel 是 AWT 中的另一个典型的容器，它代表不能独立存在、必须放在其他容器中使用。 1、可作为容器来盛装其他组件，为放置组件提供空间。 2、不能单独存在，必须放置到其他容器当中。 3、默认使用 FlowLayout 作为布局管理器。

题目：在 Java 图形用户界面编程中，如果需要显示信息，一般是使用_____类的对象来实现。JLabel
原因：jlable 显示标签信息；jbutton 按键显示；jtextarea 多行文本；jtextfield 表示显示单行文本；

题目：
```
class Base{
    public Base(String s){
        System.out.print("B");
    }
}
public class Derived extends Base{
    public Derived (String s) {
        System.out.print("D");
    }
    public static void main(String[] args){
        new Derived("C");
    }
}
```
原因：会报编译错误：原因：子类构造方法在调用时必须先调用父类的，由于父类没有无参构造，必须在子类中显示调用，修改子类的构造方法可以使用下面方法：

```
public Derived(String s){
        super("s");
        System.out.print("D");
    }
```

题目：LinkedList 继承自 List　和　HashSet 继承自 AbstractSet
原因：LinkedList 是继承自 AbstractSequentialList（抽象类，实现了 List 接口）的，并且实现了 List 接口;　　.HashSet 是继承自 AbstractSet，实现了 Set 接口。所以 C 正确。

题目：下列流当中，属于处理流的是：
原因：按照流是否直接与特定的地方（如磁盘、内存、设备等）相连，分为节点流和处理流两类；节点流：可以从一个特定的地方读写数据；处理流：对一个已经存在的流的连接和封装，通过所封装的流的功能来实现数据的读写；
常用的节点流有：文件流 fileinputstream；字符串 stringreader；数组 bytearrayinputstream；管道 pipeinputstream
创建的处理流：转换流 bufferedinputstream；转换流 inputstreamreader；数据流 datainputstream

题目：在方法中定义的局部变量在该方法被执行时创建　错误
原因：在方法中的局部变量在方法被调用的时候入栈并穿件，方法入栈创建栈帧包括局部变量表操作数栈，局部变量存放局部变量。并在执行该方法的时候被创建；

题目：已知有下列 Test 类的说明，在该类的 main 方法内，则下列哪个语句是正确的？（）

```
public class Test
{
private float f = 1.0f;
int m = 12;
static int n = 1;
public static void main (String args[])
{
Test t = new Test();
}
}
```

原因：正确：t.f;因为 f 虽然是 test 类中的私有成员属性，但是因为 main 方法就在 test 类中，因此可以通过对象名.属性名的方式进行调用；
错误：this.nstatic 静态成员属性不能使用 this 关键字进行调用；

题目：下面有关 jsp 中静态 include 和动态 include 的区别，说法错误的是？
正确：静态 include 和动态 include 都可以允许变量同名的冲突.页面设置也可以借用主文件的
原因：静态的 include：是 jsp 的指令来实现的，<% @ include file="xx.html"%> 特点是 共享 request 请求域，先包含再编译，不检查包含页面的变化。
动态的 include：是 jsp 动作来实现的，<jsp:include page="xx.jsp" flush="true"/>　这个是不共享 request 请求域，先编译在包含，是要检查包含页面的变化的。

题目：public class MyClass {

```
    long var;
    public void MyClass(long param) { var = param; }//(1)
    public static void main(String[] args) {
        MyClass a, b;
        a =new MyClass();//(2)
        b =new MyClass(5);//(3)
    }
}
```

原因：错误：编译错误将发生在（1），因为构造函数不能指定返回值；因为构造方法没有返回类型，不需要 void 进行修饰

正确：编译错误将在（3）处发生，因为该类没有构造函数，该构造函数接受一个 int 类型的参数

题目：ArrayList list = new ArrayList(20);中的 list 扩充几次

原因：正解 0 次，错解：2 次：因为源码里面规定的是，默认数组大小是 10，扩容机制的扩 1.5 倍，但是创建对象时如果给了初始值就是默认传入的大小，不会进行扩容；

题目：public class Test {

```
public static void main(String args[]) {
int i = -5;
i =    ++(i++);
System.out.println(i);
}
}
```

原因：错解-3；正解：编译错误；因为++()括号里必须是一个变量，而 i++是一个字面量；

tips：序列化和反序列化：把看得懂的转换为看不懂的就是序列化；把看不懂的转换为看得懂的就是反序列化；

tips：java 提供的事件处理模型是一种人机交互模型的基本三要素是：事件源、时间对象和事件监听器。

题目：错误：局部内部类前面可以修饰 public,protected 和 private

原因：局部内部类是定义在一个方法或者是一个作用域里面的类，他和成员内部类的却别在于局部内部类的访问权限于方法内或者是该作用域内；2.局部内部类就像是方法里面的局部不连梁一样是不能有 public，protected，private 以及 static 修饰符的；

补充：成员内部类：成员内部类依附于外部类而存在的，也就是说如果创建成员内部类的对象，前提是必须存在一个外部类的对象；内部类可以拥有 private 访问权限，protected 访问权限，public 访问权限以及报访问权限；如果成员内部类用 private 来修饰则只能扎起外部类的内部访问，如果是用 public 来修饰则可以在任何地方都能够访问，如果使用的 peotected

来修饰，则只能在同一个包下面进行访问；如果采用的是默认的访问权限，则只能在同一个包线面进行访问，恩爱不累只能被 public 和默认两种形式来修饰；匿名内部类：一般使用匿名内部类的方法来编写时间监听代码；2.匿名内部类是不能有访问修饰符合 static 修饰符的；3.匿名内部类是唯一一种没有构造器的类；匿名内部类用于继承其他类或者是实现接口并不需要增加额外的方法，只是对继承方法的实现或者是重写；匿名内部类用于继承其他类或者是实现接口，并不需要增加额外的方法，只是对继承方法的实现或者是重写；静态内部类是不需要依赖于外部类的，这点和类的静态成员属性有点类似；不能使用外部类的非 static 成员变量或者是方法；

题目：
```
public class HelloA {
    public HelloA() {
        System.out.println("A 的构造函数");
    }
{
    System.out.println("A 的构造代码块");
}
static {
    System.out.println("A 的静态代码块");
}
public static void main(String[] args) {
    HelloA a = new HelloA();
}
```
}打印顺序 A 的构造代码块> A 的构造函数

原因：需要牢记“先静态后非静态，先父类后子类”加载顺序是：父类静态代码块--->子类静态代码块--->父类构造代码块--->父类的构造函数--->子类构造代码块--->子类构造方法；

题目：在 Jdk1.7 中，下述说法中抽象类与接口的区别与联系正确的有哪些？

正确：抽象类中可以有普通成员变量，接口中没有普通成员变量。抽象类和接口中都可以包含静态成员常量。

原因： 接口和抽象类之间的区别：1.抽象类可以有构造方法，接口中不能有构造方法。2.抽象类中可以包含非抽象的普通方法，接口中的所有方法都必须是抽象的，不能有非抽象的普通方法。3.抽象类中可以有普通的成员变量；4.抽象类中的抽象方法的访问类型可以是 public，protected 和默认的类型；5.抽象类中可以包含静态方法，接口中不能包含静态方法；6，抽象类和接口中都可以包含静态成员变量，抽象类中的静态成员变量的访问类型可以任意，但接口中定义的变量只能是 public static final 类型，并且默认即 public staticfinal 类型的；7.一个类可以实现多个接口，但只能继承一个抽象类，二者在应用方面有一定的区别：接口风度哦的是在系统架构设计方法发挥作用，主要用于定义一个模板之间的通信契约，而抽象类在代码实现方面发挥作用，可以实现代码的重用，例如，模板方法设计模式是抽象类中的一个典型应用，假设某个项目所用的 wervlet 类都要用相同的方式进行权限判断，记录访问日志和处理以上那么就可以定义一个抽象的积累，让所有的 servlet 都继承这个抽象积累，在抽象积累的 service 方法中完成权限判断，记录访问日志和处理异常的代码，在各个子类中只是完成各自的业务逻辑代码。