《JAVA面向对象程序设计》试题与解析库

1) 在JAVA编程中，关于Font下列（）是正确的。（选择两项）
   a) 在我们的程序中可以使用Font类中定义的字体常量。
   b) 我们可以使用Tookit类中的函数来获取我们的机器中java所支持的字体的列表
   c) 我们可以自己创建字体实例对象
   d) 以上说明都正确

【解析】参考答案：A B

Font属于java.awt包中的一个类，专门用来设置程序中的字体实例对象。

创建Font类的对象时使用的方法：getFont(int face,int style,int size);

例如：Font font = Font.getFont(Font.FACE_SYSTEM,Font.STYLE_BOLD,Font.SIZE_MEDIUM);

无论哪一个参数，都只能使用系统设置的数值，这些数值具体的大小在不同的手机上可能不同。下面对于其中的三个参数的取值做详细的介绍：

face参数指字体的外观；style参数指字体的样式；size参数指字体的大小 。这三个参数都是系统中定义好的一些常量。

获得系统的默认字体：Font font = Font.getDefaultFont();

ToolKit类是 Abstract Window Toolkit 的所有实际实现的抽象父类。Toolkit 用于把各种组件绑定到特定的本地工具箱实现上。

该类有个方法getFontList()

用于返回该工具箱中可用字体名。 也就是可以获取机器中java所支持的字体的列表。

2) 在Java语言中，在包Package1中包含包Package2,类Class_A直接隶属于包Package1,类Class_B直接隶属于包Package2。在类Class_C要应用Class_A的方法A和Class_B的方法B,你需要（）语句。（选择两项）
   a) import Package1.*;
   b) import Package1.Package2.*;
   c) import Package2.*
   d) import Package2.Package1.*;

【解析】A B

Java语言中的包机制就像Windows中的文件夹，用来归类 类文件的位置。要使用包中的类，有如下二种方法：

- 直接导入该包中要用到的类。例如要用到Color类，可以
  import java.awt.Color
- 干脆将包中所有的类直接导入。但此时如果包有嵌套的包，不能将嵌套包中的类倒入。
  Import java.awt.*

本题包的关系如下：

Package1包嵌套包Package2

Class_A　　Class_B

掌握了这二条，要在类Class_C要应用Class_A的方法A和Class_B的方法B。也就是要在Class_C中导入Class_A和Class_B。方法如下：

```
import Package1.*     //可以导入Class_A
import Package1.Package2.* //可以导入Class_B
```

3) import java.awt.*;
import java.applet.*;
public class ButtonDemo extends Applet {
Public void init()
{
Button pushBotton=new Button(“ok”);
Button downButton=new Button(“Yes”);
add(pushBotton);
add(downBotton);}
}
根据以上代码，下列解释正确的是（）。（选择两项）
a) 该代码画了一个按钮
b) Button(“ok”)创建一个有显示“ok”的按钮
c) Button()是构造函数
d) 按钮属于容器

【解析】B C

Button是一个GUI组件，并不是容器。因为它里面不能再放其他组件。要创建按钮，可以使用Button类创建。

Button pushBotton=new Button(“ok”)

其中pushBotton是按钮实例对象名，Button（）是构造方法，初始化该按钮，此处带有参数ok，表示该按钮所显示的文本为OK。等价于setText方法的作用。此代码中的add方法是将按钮添加到容器。但此时该按钮并没有显示出来，因为还没有实现画按钮。

【解析】

4) 在JAVA编程中，关于Graphics,下面（）是正确的。（选择一项）
a) 在这个类中定义了一些基本的绘图方法
b) 这个类还存在一些不足，因此出现了Graphics2D类，弥补了这个类的某些不足
c) 这个类是一个抽象类，我们不能创建这个类的实例
d) 以上说法都正确

【解析】D

类 java.awt.Graphics，

定义：public abstract class Graphics extends Object

Graphics 类是所有图形上下文的抽象基类，该类包含了一些基本的绘图方法。这个上下文允许应用将图形绘制到由不同设备实现的组件上，以及绘制到空闲屏幕的映像中。

由于 Graphics 是一个抽象类，应用不能直接调用该构造函数。图形上下文是从其他图形上下文获得的或是通过在一个组件上调用 getGraphics 创建的。例如使用如下这二个方法：

create, getGraphics

针对该类的一些缺点，出现了Graphics2D类，利用java.awt.Graphics2D类可以绘制各种图形，矩形，圆，二次曲线，饼形以及它们的填充图形等-,Java,2D图形编程/2D Graphic。

5) 在JAVA编程中，将鼠标放在按钮上以后，用鼠标单击按钮，将会发生鼠标事件和组件激活事件，就鼠标事件而言，将调用（）个监听器方法。（选择一项）
a) 1
b) 2
c) 3
d) 4

【解析】B

鼠标事件在单击按钮时实际包含了二个事件：

鼠标压下 和 鼠标 弹起

在Java的事件机制中，每个事件都必须有一个事件监听者。故选B

6) JAVA中，为了辨别用户关闭窗口的时间，要实现监听器接口（）。（选择一项）
a) MouseListener

b) ActionLisener
c) WindowListener
d) 以上都要

【解析】D

在Java程序中这一般是通过 实现适当的事件监听者接口来完成的。比如如果需要响应按钮事件，就需要实现ActionListener 监听者接口；如果需要响应窗口事件，就需要实现WindowListener监听者接口。此处当用鼠标点击关闭时，会产生一个MouseEvent，此类需要实现MouseListener接口。故本题选D

7) 在Java语言中，如果你有下面的类定义：

```
abstract class Shape {
    abstract void draw();
}
class Square extends Shape{}
```

如果你试图编译上面的代码会发生（）。（选择一项）

a) 一切成功编译
b) Shape可以编译Square不能编译
c) Square可以编译Shape不能编译
d) Shape.Square都不能编译

【解析】D

由于Square类没有实现父类Shape的方法draw，从而导致这二个类都不能编译。

8) 对于布局管理器解释正确的有（）。（选择两项）

a) FlowLayout 以由上到下的方式从左到右排列组件
b) BorderLayout使用“东”、“西”、“南“、“北”、“居中”来指定组件的位置
c) GridLayout提供了类似于选项卡式的对话框的功能
d) CardLayout是最灵活的布局方案

【解析】 B D

Java中的布局方式：

BorderLayout 边界布局方式：使用“东”、“西”、“南“、“北”、“居中”来指定组件的位置

FlowLayout 流式（顺序）布局：按照从左到右由上到下的方式排列组件。

GridLayout 网格布局。把组件按照网格来放置。

CardLayout 卡片布局。 对象是容器的布局管理器。它将容器中的每个组件当作一个卡片来处理。在某 一时间，只有一个卡片是可见的，容器象一个卡片堆栈一样工作。 ... CardLayout 定义了 一系列方法，来允许一个应用顺序地翻动这些卡片，或显示一个指定的卡片。是最灵活的布局方案。

9)

```
int[] my_Array;
my_Array=new int[5];
for(int count=0;count<=5;count++)
    System.out.println(my_Array[count]);
```

以上Java代码运行的结果是（）。（选择一项）

a) 将1，2，3，4，5输出到屏幕
b) 将0，1，2，3，4输出到屏幕
c) 将0，1，2，3，4，5输出到屏幕
d) 将出现运行时异常

【解析】D

数组my_Array包含5个元素，元素的索引从0到4；而在for循环体中出现了当count=5时，会超出索引边界；故my_Array[5]会产生异常。

10) 下面描述（）是正确的。（选择两项）

a) Java的源程序必须以”.java”或”.jav”扩展名保存

b) Java VM可以是软件也可以是硬件
c) Java使用解释器执行代码
d) 用高级语言编写的代码可以让计算机理解并执行它们

【解析】B C

Java的源程序必须以".java"结尾。

高级语言编写的程序必须经过编译或解释为低级语言；也即要翻译为计算机可以理解的机器语言。

Java VM（虚拟机）可以是软件也可以是硬件。

Java使用解释器执行代码。

11) 在JAVA编程中，基于线程的多任务处理环境中，执行特定任务的可执行代码的最小单位是（）。（选择一项）
a) 进程
b) 线程
c) 应用程序
d) 服务

【解析】B

进程是由一个或多个线程组成，进程中可执行代码的最小单位就是线程。

12) 下列选项中，属于Java语言的关键字的是（）。（选择两项）
a) goto
b) malloc
c) extends
d) FALSE

【解析】 A C

先做下面几个题目：

1, which of the following are keywords or reserved words in java ?
a) if b)then c)goto d)while e)case f)sizeof

2, which of the following are java key words ?
a)double b)Switch c)then d)instanceof

3, which of these are key words in java ?
a) default b)NULL c)String d)throws e)long f)true

答案： 1，acde 2，ad 3，adef 作对了吗^_^

解释来了:

1，then和sizeof都不是java的关键字，熟悉c或者c＋＋，写惯了asp的高手就要小心喽。

2，所有的关键字都是小写的，所以Switch不是关键字。instanceof看上去像方法名，但其实是关键字；

3，大写的NULL不是java语言的关键字。String是java语言的一个封装类的类名，也不是关键字。

再来点系统的:

正确识别java语言的关键字（keyword）和保留字（reserved word）是十分重要的。Java的关键字对java的编译器有特殊的意义，他们用来表示一种数据类型，或者表示程序的结构等。保留字是为java预留的关键字，他们虽然现在没有作为关键字，但在以后的升级版本中有可能作为关键字。

关键字列表

abstract boolean break byte case
catch char class continue default
do double else extends false
final finally float for if
implements import instanceof int interface
long native new null package
private protected public return short
static super switch synchronized this
throw throws transient true try

void volatile while

保留字

const，goto

注意点：

识别java语言的关键字，不要和其他语言如c/c++的关键字混淆。

const和goto是java的保留字。

所有的关键字都是小写

friendly，sizeof不是java的关键字

13) 在JAVA语言中，Panel默认的布局管理器是（）。（选择一项）

a) Borderlayout
b) FlowLayout
c) GridLayout
d) GridBagLayout

【解析】A

常见的几种布局：

BorderLayout 边界布局方式：使用“东”、“西”、“南“、“北”、“居中”来指定组件的位置.为默认的布局管理器。

FlowLayout 流式（顺序）布局：按照从左到右由上到下的方式排列组件。

GridLayout 网格布局。把组件按照网格来放置。

14) 在JAVA语言中，使我们能够使用和更改字体来显示或键入文本的类是（）。（选择一项）

a) Java.awt .Font
b) Java.awt.Graphics.Font
c) Java.Graphics.Font
d) Java.Font

【解析】A

Font类位于java.awt包中。

15) 在Java中，关键字（）使类不能派生出子类。（选择一项）

a) final
b) public
c) private
d) volatile
e) native

【解析】A

在类的前面加了final关键字，说明该类是一个最终的类，当然也就不能被别人继承，也就不能派生出子类了。

16) 在JAVA编程中，Swing包中的组件处理事件时，下面（）是正确的。（选择一项）

a) Swing包中的组件也是采用事件的授权处理模型来处理事件的
b) Swing包中的组件产生的事件类型，也都带有一个J字母，如：JMouseEvent
c) Swing包中的组件也可以采用事件的传递处理机制
d) Swing包中的组件所对应的事件适配器也是带有J字母的，如：JMouseAdapter

【解析】A

Swing是由100%纯Java实现的，Swing组件是用Java实现的轻量级（ light-weight）组件，没有本地代码，不依赖操作系统的支持，这是它与AWT组件的最大区别。由于AWT组件通过与具体平台相关的对等类（Peer）实现，因此Swing比AWT组件具有更强的实用性。Swing在不同的平台上表现一致，并且有能力提供本地窗口系统不支持的其它特性。

Swing包中的组件也是采用事件的授权处理模型来处理事件的。它的事件类型和事件适配器类依然是按照awt包中的方式来定义的，并没有附带一个J字母。

17) 在JAVA编程中，Java编译器会将java程序转换为（）。（选择一项）
a） 字节码
b） 可执行代码
c） 机器代码
d） 以上所有选项都不正确
【解析】A
Java编译器将java程序转换为字节码。
18) 在JAVA编程中，以下（）命令用来执行java类文件。（选择一项）
a） javac
b） java
c） appletviewer
d） 以上所有选项都不正确
【解析】B
Javac是用来编译java的源文件的命令。
Java执行经过编译后产生的类文件。
Appletviewer是用来查看java Applet程序的工具。

19) 在Java中，下列代码段允许按钮注册一个action事件的是（）。（选择一项）
a） button.enableActionEvents();
b） button.addActionListener(anActionListener);
c） button.enableEvents(true);
d） button.enableEvents(AWTEvent.ACTION_EVENT_MASK);
【解析】B
向组件注册事件监听器的方法是addXXXListener。其中XXX代表事件。
事件监听器(event listener)就是一个实现listener接口的对象。所以，程序员要做的就是创建一个listener对象，然后向发起事件的组件注册这个对象。注册的过程就是调用组件的addXXXListener( )方法，这里"XXX"表示组件所发起的事件的类型。只要看一眼"addListener"方法的名字就能知道组件能处理哪种事件了，所以如果你让它听错了事件，那么编译就根本通不过。到后面你就会看到，JavaBean在决定它能处理哪些事件时，也遵循"addListener"的命名规范。

20) 在Java语言中，下面变量命名合法的有（）。（选择两项）
a） variable123
b） 123variable
c） private
d） selg_asd
【解析】A D
命名规范：
定义这个规范的目的是让项目中所有的文档都看起来像一个人写的，增加可读性，减少项目组中因为换人而带来的损失。（这些规范并不是一定要绝对遵守，但是一定要让程序有良好的可读性）
● 普通变量 的命名
以字母或下划线开头，由字母、数字、下划线等组成。但不能是java的关键字。
● Package 的命名
Package 的名字应该都是由一个小写单词组成。
● Class 的命名
Class 的名字必须由大写字母开头而其他字母都小写的单词组成
● Class 变量的命名
变量的名字必须用一个小写字母开头。后面的单词用大写字母开头。
● Static Final 变量的命名
Static Final 变量的名字应该都大写，并且指出完整含义。

● 参数的命名

参数的名字必须和变量的命名规范一致。

● 数组的命名

数组应该总是用下面的方式来命名：

byte[] buffer;

而不是：

byte buffer[];

● 方法的参数

使用有意义的参数命名，如果可能的话，使用和要赋值的字段一样的名字：

```
SetCounter(int size){
this.size = size;
}
```

21) 在JAVA编程中，关于TextField的语句如下：

JTextField t = new JTextField(“they are good”,40);

下面（）是正确的。（选择两项）

a) 在这个JTextField中，最多只能输入40个字符

b) 在这个JTextField中，最少可以输入40个字符

c) 在这个JTextField中，能够输入的字符不一定是40个

d) 在这个JTextField中，用户可以编辑所输入的字符

【解析】C D

JTextField构造方法的第二个参数表示文本框的列数。但并不限制它能输入的字符数。

public JTextField(String text,
int columns)

创建一个新的文本域，并用待显示的指定文本初始化，且它的宽度足以显示指定的字符数。

参数：

text - 显示的文本。

columns - 字符数

22) String s1 = new String(“Hello”);

String s2 = new String(“there”);

String s3 = new String();

上面是Java程序中的一些声明，选项中能通过编译的是（）。（选择一项）

a) s3 = s1+ s2

b) s3 = s1 & s2

c) s3 = s1 || s2

d) s3 = s1 && s2

【解析】A

Java的String对象是一个不可变的对象，每次修改或者构造字符串都会有一个新的String对象被创建。本题中只能对字符串对象进行+操作。读者可以从下面来认识String。

问题一：我声明了什么！

String s = "Hello world!";

许多人都做过这样的事情，但是，我们到底声明了什么？回答通常是：一个String，内容是“Hello world!”。这样模糊的回答通常是概念不清的根源。如果要准确的回答，一半的人大概会回答错误。

这个语句声明的是一个指向对象的引用，名为“s”，可以指向类型为String的任何对象，目前指向"Hello world!"这个String类型的对象。这就是真正发生的事情。我们并没有声明一个String对象，我们只是声明了一个只能指向String对象的引用变量。所以，如

果在刚才那句语句后面，如果再运行一句：

String string = s;

我们是声明了另外一个只能指向String对象的引用，名为string，并没有第二个对象产生，string还是指向原来那个对象，也就是，和s指向同一个对象。

问题二："=="和equals方法究竟有什么区别？

==操作符专门用来比较变量的值是否相等。比较好理解的一点是：

int a=10;

int b=10;

则a==b将是true。

但不好理解的地方是：

String a=new String("foo");

String b=new String("foo");

则a==b将返回false。

根据前一帖说过，对象变量其实是一个引用，它们的值是指向对象所在的内存地址，而不是对象本身。a和b都使用了new操作符，意味着将在内存中产生两个内容为"foo"的字符串，既然是“两个”，它们自然位于不同的内存地址。a和b的值其实是两个不同的内存地址的值，所以使用"=="操作符，结果会是false。诚然，a和b所指的对象，它们的内容都是"foo"，应该是“相等”，但是==操作符并不涉及到对象内容的比较。

对象内容的比较，正是equals方法做的事。

所以当你是用equals方法判断对象的内容是否相等，请不要想当然。因为可能你认为相等，而这个类的作者不这样认为，而类的equals方法的实现是由他掌握的。如果你需要使用equals方法，或者使用任何基于散列码的集合（HashSet,HashMap,HashTable），请察看一下java doc以确认这个类的equals逻辑是如何实现的。

问题三：String到底变了没有？

没有。因为String被设计成不可变(immutable)类，所以它的所有对象都是不可变对象。请看下列代码：

String s = "Hello";

s = s + " world!";

s所指向的对象是否改变了呢？从本系列第一篇的结论很容易导出这个结论。我们来看看发生了什么事情。在这段代码中，s原先指向一个String对象，内容是"Hello"，然后我们对s进行了+操作，那么s所指向的那个对象是否发生了改变呢？答案是没有。这时，s不指向原来那个对象了，而指向了另一个String对象，内容为"Hello world!"，原来那个对象还存在于内存之中，只是s这个引用变量不再指向它了。

通过上面的说明，我们很容易导出另一个结论，如果经常对字符串进行各种各样的修改，或者说，不可预见的修改，那么使用String来代表字符串的话会引起很大的内存开销。因为String对象建立之后不能再改变，所以对于每一个不同的字符串，都需要一个String对象来表示。这时，应该考虑使用StringBuffer类，它允许修改，而不是每个不同的字符串都要生成一个新的对象。并且，这两种类的对象转换十分容易。

23) 在Java语言中，按“东，西，南，北，中”指定组件的位置的布局管理器是（）。（选择一项）

a) FlowLayout
b) GridLayout
c) BorderLayout
d) CardLayout
e) GridBagLayout

【解析】C

Java中的布局方式：

BorderLayout 边界布局方式：使用“东”、“西”、“南“、“北”、“居中”来指定组件的位置

FlowLayout 流式（顺序）布局：按照从左到右由上到下的方式排列组件。

GridLayout 网格布局。把组件按照网格来放置。

CardLayout 卡片布局。对象是容器的布局管理器。它将容器中的每个组件当作一个卡片来处理。在某一时间，只有一个卡片是可见的，容器象一个卡片

堆栈一样工作。 ... CardLayout 定义了 一系列方法，来允许一个应用顺序地翻动这些卡片，或显示一个指定的卡片。是最灵活的布局方案。

24) 在Java中允许创建多线程应用程序的接口是（）。（选择一项）
a) Threadable
b) Runnable
c) Clonable
d) 以上均不是
【解析】A

25) 在Java语言中，把组件放在BorderLayout的（）区域时，它会自动垂直调整大小，但不是水平调整。（选择一项）
a) North或South
b) East或West
c) Center
d) North,South或Center
【解析】A
在BorderLayout布局方式时，当把组件放在North或South区域时，会自动水平调整组件的大小。相反当在East或West方位时，会自动垂直调整组件的大小。当在Center方位时，垂直和水平方位都会自动调整。

26) 在JAVA编程中，以下（）命令能够将Java源文件转换为类文件。（选择一项）
a) appletviewer
b) java
c) javac
d) 以上所有选项都不正确
【解析】C
Javac是用来编译java的源文件的命令。产生类文件，也就是字节文件。Java执行经过编译后产生的类文件。
Appletviewer是用来查看java Applet程序的工具。

27) 在JAVA编程中，Java具有下列（）特点。（选择三项）
a) 面向对象
b) 跨平台
c) 安全
d) 集中式体系结构
e) 可编译成机器代码
【解析】 A B C
JAVA语言是完全面向对象的、能够跨平台、高安全性的分布式体系结构。可以被虚拟机编译为字节码。

28) 在Java中，欲定义某类所在的包外的所有类都能访问这个类，则应用的关键字是（）。（选择一项）
a) protected
b) private
c) public
【解析】C
Public 访问修饰符，该类所在包内和包外的类都可以访问。最开放。
Protected访问修饰符，只有该类所在包内或它的派生类才能访问。
Private访问修饰符 ，只有该类所在包内的类才能访问。

29) 在Java中，根据你的理解，下列方法（）可能是类Orange的构造函数。（选择三项）
a) Orange(){…}
b) Orange(…){…}
c) Public void Orange(){…}
d) Public Orange(){…}

e) Public OrangeConstuctor(){…}

【解析】A B D

构造方法要注意的几个地方：

名字与类名一样。可以带参数重载。

不能带返回类型，包括void。

30) 在Java中，下列选项表示字符”a”值的是（）。（选择一项）

a) ‘a’

b) “a”

c) new Character(a)

d) \000a

【解析】A

字符应用单引号表示‘a’;

new Character(a)表示创建一个字符对象a

\000a 用ASCII表示应为 \097

31) 在Java中，可以使线程运行的方法是（）。（选择一项）

a) init();

b) start();

c) resume();

d) sleep();

【解析】B

32) 在JAVA语言中，下面是main()方法的部分代码：

Frame f=new Frame(“My Frame”);

f.setSize(100,100);

为在屏幕显示f,应增加的代码是（）。（选择一项）

a) f.appear();

b) f.setForeground();

c) f.setVisible();

d) f.enable();

【解析】C

框架Frame是一个顶层容器。在创建后，要在屏幕上显示只需将其设为可见。故选C。调用setVisible方法。

A项错误，没有该方法

B项设置该窗体的前景色。

D项使该窗体为可用。

33) JAVA中，对于drawImage(image,x,y,width,height,this)方法解释正确的是（）。（选择两项）

a) image是要绘制的图像

b) x,y是表示图像的中心的位置

c) width是源图像的宽度

d) this是容器

【解析】 A D

在指定位置并且按指定大小绘制指定的 Image。x,y是表示要画图像的位置坐标。

34) JAVA中，欲返回按钮的标签，其方法是（）。（选择一项）

a) getActionCommand()

b) setLabel(string str)

c) button()

d) getLabel()

【解析】A

欲返回按钮的标签使用getActionCommand()方法。

35) 在JAVA中，假设我们有一个实现ActionListener接口的类，以下方法，（）能够为一个Button类注册这个类。（选择一项）

a) addListener()
b) addActionListener()
c) addButtonListener()
d) setListener()

【解析】B

Swing的事件模型中，组件可以发起(或"射出")事件[译注1]。各种事件都是类。当有事件发生时，一个或多个"监听器(listener)"会得到通知，并做出反应。这样事件的来 源就同它的处理程序分隔开来了。一般说来，程序员是不会去修改Swing组件的，他们写的都是些事件处理程序，当组件收到事件[译注1]时，会自动调用这些代码，因此Swing的事件模型可称得上是将接口与实现分隔开来的绝好范例了。

事件监听器(event listener)就是一个实现listener接口的对象。所以，程序员要做的就是创建一个listener对象，然后向发起事件的组件注册这个对象。注册的过程就是调用组件的**addXXXListener( )方**法，这里"XXX"表示组件所发起的事件的类型。只要看一眼"addListener"方法的名字就能知道组件能处理哪种事件了，所以如果你让它听错了事件，那么编译就根本通不过。到后面你就会看到，JavaBean在决定它能处理哪些事件时，也遵循"addListener"的命名规范。

36) 在JAVA语言中，包pack1的类class1 中有成员方法：protected void method_1(){…},private void method_2(){ …},public void method_3(){ ….}和void method_4(){…},在包pack2中的类class2 不是class1的子类，你在class2 中可以调用的方法有（）。（选择一项）

a) method_1
b) method_2
c) method_3
d) method_4

【解析】C

本题可以通过画出包中类的结构来分析：

pack1: class1 : 受保护的method_1 method_4 私有的method_2 公共的method_3();

pack2: class2:

由于class2和class1 之间没有继承关系；故 class2只能调用class1种的public修饰的方法: method_3().

37) JAVA中，使用（）修饰符时，一个类能被同一包或不同包中的其他类访问。（选择一项）

a) private
b) protected
c) public
d) friendly

【解析】C

Public 访问修饰符，该类所在包内和包外的类都可以访问。最开放。

Protected访问修饰符，只有该类所在包内或它的派生类才能访问。

Private访问修饰符 ，只有该类所在包内的类才能访问。

Java中无friendly修饰符。

38) 在JAVA中，类Worker是类Person的子类，Worker的构造方法中有一句“super()“，该语句（）。（选择一项）

a) 调用类Worker中定义的super()方法
b) 调用类Person中定义的super()方法
c) 调用类Person的构造函数
d) 语法错误

【解析】C

有些时候经常需要在子类中调用父类的构造方法，此时需用super方法来调用。

39) 在JAVA语言中，下列语言（）可以画出一矩形框架，其距左边界为0像素，距上边界为10像素，宽为30像素，高为40像素。（选择一项）

a)
```
Graphics g=new Graphics();
g.drawRect(10,0,30,40);
```
b)
```
Graphics g=new Graphics();
g.drawRect(0,10,30,40);
```
c)
```
Graphics g =new Graphics();
g.drawRect(30,40,10,0)
```
d)
```
Graphics g =new Grphics();
g.drawRect(30,40,0,10);
```

【解析】B

Graphics类的方法drawRect的语法如下：

```
public void drawRect(int x,
                     int y,
                     int width,
                     int height)
```

绘制指定矩形的轮廓。矩形的左边和右边分别为 x 和 x + width 。顶部边沿和底部边沿分别为 y 和 y + height 。该矩形是使用图形上下文的当前颜色绘制的。

参数：

x - 将被绘制的矩形的 x 坐标。

y - 将被绘制的矩形的 y 坐标。

width - 将被绘制的矩形的宽度。

height - 将被绘制的矩形的高度

故本题选 B

40) 在JAVA中，类Animal中的方法printA()定义如下：

```
public void printA(){
    int a=10;
    int result=10%3;
    System.out.println(result);
```

在类Dog中方法printA()定义如下：

```
public void printA(){
    int a=10;
    Systme.out.println(a/3);
}
```

Dog类的定义如下：

```
    Class Dog extends Animal{…}
```

若有语句：

```
Animal animal=new Dog();
animal.printA();
```

则这段代码输出为（）。（选择一项）

a) 0
b) 3.3333
c) 2
d) 3

【解析】B

本题关键要理解如何使父类的对象调用子类的方法。因为通常都是子类调用父类的方法。

Animal animal=new Dog();

此处虽然声明的是父类Animal类型的对象，但在初始化时调用的是子类的构造方法。正是因为此，使得animal对象指向Dog类的引用。该对象animal在调用printA（）方法时不是调用父类的方法，而是调用在子类中重写的printA（）方法。故本题执行10/3=3.3333.

41) 在JAVA中，下列（）代码段允许按钮注册一个action事件。（选择一项）

a) button.enableActioinEvents();
b) button.addActionListener(anActionListener);
c) button.enableEvents(true);
d) button.enableEvents(AWTEvent.ACTION_EVENT_MASK);

【解析】B

Swing的事件模型中，组件可以发起(或"射出")事件[译注1]。各种事件都是类。当有事件发生时，一个或多个"监听器(listener)"会得到通知，并做出反应。这样事件的来 源就同它的处理程序分隔开来了。一般说来，程序员是不会去修改Swing组件的，他们写的都是些事件处理程序，当组件收到事件[译注1]时，会自动调用这些代码，因此Swing的事件模型可称得上是将接口与实现分隔开来的绝好范例了。

事件监听器(event listener)就是一个实现listener接口的对象。所以，程序员要做的就是创建一个listener对象，然后向发起事件的组件注册这个对象。注册的过程就是调用组件的**addXXXListener( )方**法，这里"XXX"表示组件所发起的事件的类型。只要看一眼"addListener"方法的名字就能知道组件能处理哪种事件了，所以如果你让它听错了事件，那么编译就根本通不过。到后面你就会看到，JavaBean在决定它能处理哪些事件时，也遵循"addListener"的命名规范。

42) 在JAVA编程中，实现Runnable接口时必须实现的方法是（）。（选择一项）

a) wait()
b) run()
c) stop()
d) start()

【解析】B

这是在JBuild中的工程中的main方法。程序从此处开始运行，需要运行进程。此处调用了Runnable接口的run方法，并对该方法进行了重写。

```
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
       try {
            UIManager.setLookAndFeel(UIManager.
                        getSystemLookAndFeelClassName());
        } catch (Exception exception) {
          exception.printStackTrace();
        }
```

Runnable接口有一个最重要的方法run，在进程要运行时，必须实现该方法。

43) 在JAVA语言中，下面关于类的描述正确的是（）。（选择一项）

a) 一个子类可以有多个超类
b) 一个超类可以有多个子类
c) 子类可以使用超类的所有
d) 子类一定比超类有更多的成员方法

【解析】B

Java语言中只能实现单重继承，也就是一个子类只能有一个父类。如果要实现多重继承，只能使用接口来实现。因为一个子类可以实现多重接口。

一个超类可以派生出多个子类。

子类的方法不一定就比父类多。

44) Java Applet的三种状态：1.Init(),2.Start(),3.Paint(),在Applet载入时的顺序是（）。（选择一项）

a) 1，2，3
b) 2，1，3
c) 2，3，1
d) 1，3，2

【解析】A

45) JAVA是一种完全面向（）的语言。（选择一项）

a) 过程
b) 对象
c) 组件
d) 服务

【解析】B

JAVA语言是完全面向对象的、能够跨平台、高安全性的分布式体系结构。可以被虚拟机编译为字节码。

46) JAVA中，实现继承的关键字是（）。（选择一项）

a) public
b) class
c) extends
d) implements

【解析】C

A extends B

表示A继承于B。extends关键字表示扩展自。用来实现java中的继承关系。

47) JAVA程序中，（）不能用来表示注释。（选择一项）

a) //注释
b) /*注释*/
c) /**注释*/
d) /注释/

【解析】D

Java中的三种注释：

对单行代码注释 //注释

对多行代码注释 /*注释*/

在代码开头部分注释一些版权信息 /**注释*/

48) JAVA线程编程中，如果让线程睡眠，可以用（）方法实现。（选择一项）

a) start()
b) close();
c) setDaemon();
d) sleep();

【解析】D

49) 在JAVA编程中，Jpanel缺省的布局管理器是（）。（选择一项）

a) 该组件没有缺省的布局管理器
b) FlowLayout
c) JflowLayout
d) BorerLayout

【解析】D

50) import java.awt.*;
```
import java.applet.*;
public class DrawRect extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.red);
        g.drawRoundRect(150,50,50,50,20,40);
        g.setColor(Color.green);
        g.fillRoundRect(150,140,50,50,20,20);
    }
}
```
关于上面JAVA代码的输出结果，叙述正确的有（）。（选择两项）

a) 两个圆角矩形的绘制弧高都为20

b) 两个圆角矩形的绘制弧宽相同

c) 两个圆角矩形是左对齐的

d) 两个圆角矩形是上对齐的

【解析】B C

Graphics类的drawRoundRect方法用于绘制圆角的矩形。语法如下：

```
public abstract void drawRoundRect(int x,
                                   int y,
                                   int width,
                                   int height,
                                   int arcWidth,
                                   int arcHeight)
```

使用该图形上下文的当前颜色绘制轮廓为圆角的矩形。矩形的左边和右边分别为 x 和 x + width 。矩形的顶部边沿和底部边沿分别为 y 和 y + height 。

参数：

x - 将被绘制的矩形的 x 坐标。

y - 将被绘制的矩形的 y 坐标。

width - 将被绘制的矩形的宽度。

height - 将被绘制的矩形的高度。

arcWidth - 位于四个角上的弧的水平直径。 代表弧宽

arcHeight - 位于四个角上的弧的垂直直径 代表弧高

故本题选B C

51) 在JAVA语言中，不能被修改的变量是用关键字（）来修饰的。（选择一项）

a) final

b) class

c) system

d) void

【解析】A

52) 在JAVA编程中，（）可以实现跳转结构。（选择一项）

a) break

b) while

c) do-while

d) for

【解析】A

Break语句用于实现中止正在执行的循环，从而跳出该循环，转到其他语句执行。

53) 在JAVA中，关于捕获错误的语法try-catch-finally的下列描述正确的是（）。（选择两项）

a) try-catch必须配对使用

b) try可以单独使用
c) try-finally可以配对使用，finally也可单独使用
d) 在try-catch后如果定义了finally,则finally肯定会执行

【解析】AD

Java中的异常处理模型如下：

```
Try
{
}catch（）{}
   Finally {}
```

其中try不能单独使用，必须和catch一起使用。finally也不能单独使用。
如果有Finally部分，则不管是否捕捉到了错误程序都将会执行它。

54) 在JAVA SWING编程中，创建一个窗体使用组件（）。（选择一项）
a) JFrame
b) INT
c) CHAR
d) LONG

【解析】A

Swing是由100%纯Java实现的，Swing组件是用Java实现的轻量级（ light-weight）组件，没有本地代码，不依赖操作系统的支持，这是它与AWT组件的最大区别。由于AWT组件通过与具体平台相关的对等类（Peer）实现，因此Swing比AWT组件具有更强的实用性。Swing在不同的平台上表现一致，并且有能力提供本地窗口系统不支持的其它特性。

在AWT包中创建窗体是Frame类，为了区分，swing包中的许多类都在前面加了一个J字母。故为JFrame类。

55) 在JAVA编程中，编写一个APPLET需要继承（）。（选择一项）
a) JFRAME
b) APPLET
c) THREAD
d) FRAME

【解析】B

APPLET是Java小程序，专门用在网页中运行。通常需要继承Applet类。

56) 分析下列java代码：

```
class A
{
     public static void main(String[] args)
     {
         method();
     }
     static void method()
     {
         try
         {
             System.out.println(“Hello”);
         }
         finally
         {
             System.out.println(“good-bye”);
         }
     }
}
```

编译运行后，输出结果是（）。（选择一项）
a) ”Hello”

b) ”good-bye”
c) “Hello”
“good-bye”
d) 代码不能编译

【解析】C

Java中的异常处理模型如下：

```
Try
{
}catch（）{}
  Finally {}
```

其中try不能单独使用，必须和catch一起使用。finally也不能单独使用。
如果有Finally部分，则不管是否捕捉到了错误程序都将会执行它。

57）下面选项中，（）可以用来在HTML中嵌入APPLET程序。（选择一项）
a) <applet>..</applet>
b) <title></titile>
c)
</br>
d)

【解析】A

在HTML中要嵌入java小程序，使用<applet>……..</applet>标记。

58）JAVA中，按下和释放鼠标按钮的操作处理（）事件。（选择一项）
a) mouseEnterd
b) mouseExitied
c) mousePressed
d) mouseClicked

【解析】D

Java中的鼠标和键盘事件

Java中的鼠标和键盘事件
1、使用MouseListener借口处理鼠标事件
鼠标事件有5种：按下鼠标键，释放鼠标键，点击鼠标键，鼠标进入和鼠标退出
鼠标事件类型是MouseEvent，主要方法有：
getX(),getY() 获取鼠标位置
getModifiers() 获取鼠标左键或者右键
getClickCount() 获取鼠标被点击的次数
getSource() 获取鼠标发生的事件源
事件源获得监视器的方法是addMouseListener()，移去监视器的方法是removeMouseListener()
处理事件源发生的时间的事件的接口是MouseListener 接口中有如下的方法
mousePressed(MouseEvent) 负责处理鼠标按下事件
mouseReleased(MouseEvent) 负责处理鼠标释放事件
mouseEntered(MouseEvent) 负责处理鼠标进入容器事件
mouseExited(MouseEvent) 负责处理鼠标离开事件
mouseClicked(MouseEvent) 负责处理点击事件 （包含了按下和释放鼠标按钮的操作）
2、使用MouseMotionListener接口处理鼠标事件
事件源发生的鼠标事件有2种：拖动鼠标和鼠标移动
鼠标事件的类型是MouseEvent
事件源获得监视器的方法是addMouseMotionListener()
处理事件源发生的事件的接口是MouseMotionListener 接口中有如下的方法
mouseDragged() 负责处理鼠标拖动事件

mouseMoved() 负责处理鼠标移动事件
3、控制鼠标的指针形状
setCursor(Cursor.getPreddfinedCursor(Cursor.鼠标形状定义)) 鼠标形状定义见（书 P 210）
4、键盘事件
键盘事件源使用addKeyListener 方法获得监视器
键盘事件的接口是KeyListener 接口中有3个方法
public void keyPressed(KeyEvent e) 按下键盘按键
public void keyReleased(KeyEvent e) 释放键盘按键
public void keyTypde(KeyEvent e) 按下又释放键盘按键

59)
```
import java,applet,Applet;
import java.awt.*;
public class ImageDemo extends Applet
{
    Image img;
    public void init()
    {
        img=getImage(getCodeBase(),"11.gif");   //1
    }
    public void paint(Graphics g)
    {
        int w=img.getWidth(this);
        int h=img.getHeight(this);
        g.drawImage(img,120,60,w/2,h/2,this); //2
        g.drawImage(img,150,0,w*2,h*2,this);//3
    }
}
```
以上JAVA代码第（）行将图像放大。（选择一项）
a) 1
b) 2
c) 3
d) 代码并没有将图像放大
【解析】C
方法：drawIImage(image,x,y,width,height):在指定位置并且按指定大小绘制指定的 Image。x,y是表示要画图像的位置坐标。

60) MyProgram.Java被编译后，生成（ ）。（选择一项）
a) MyProgram. Obj
b) MyProgram. class
c) MyProgram.exe
d) MyProgram. bat
【解析】B
Java源代码（.java）文件通过javac命令编译成.class的字节文件。文件名不变。
61) JAVA中，访问修饰符限制性最高的是（ ）。（选择一项）
a) private
b) protected
c) public
d) friendly
【解析】A
Public 访问修饰符，该类所在包内和包外的类都可以访问。最开放。
Protected访问修饰符，只有该类所在包内或它的派生类才能访问。
Private访问修饰符，只有该类所在包内的类才能访问。限制性最高。
Java中没有friendly修饰符。

62) 在JAVA编程中，关于Swing包中的组件，下面（ ）是正确的。（选择一项）
a） Swing中的每个组件都是采用MVC模式设计的
b） JFrame窗口的关闭按钮默认不能使窗口关闭
c） Swing的组件和awt组件，在编程时不能混合使用
d） 以上都正确
【解析】A
Swing组件的设计都采用了MVC模式（Model/View/Control）；
JFrame窗口的关闭按钮默认能使窗口关闭
通常可以将swing的组件和awt的组件可以混合使用。

63) 在Java语言中，下列组件可以让用户选择多个选项有（ ）。（选择两项）
a） Checkbox
b） Radiobutton
c） List
d） Choice
【解析】A C
Checkbox为复选框按钮，可以让用户选择多个选项。
Radiobutton单选按钮，每次只能选择一个选项。
List为列表组件，可以让用户选择多个选项。
没有D项这样的组件。

64) 在JAVA编程中，（）可以实现跳转结构。（选择一项）
a） break
b） while
c） do-while
d） for
【解析】

65) 在JAVA中，关于捕获错误的语法try-catch-finally的下列描述正确的是（）。（选择两项）
a） try-catch必须配对使用
b） try可以单独使用
c） try-finally可以配对使用，finally也可单独使用
d） 在try-catch后如果定义了finally,则finally肯定会执行
【解析】

66) 在JAVA SWING编程中，创建一个窗体使用组件（）。（选择一项）
a） JFRAME
b） INT
c） CHAR
d） LONG
【解析】

67) 在JAVA编程中，编写一个APPLET需要继承（）。（选择一项）
a） JFRAME
b） APPLET
c） THREAD
d） FRAME
【解析】

68) 分析下列java代码：

```
class A
{
    public static void main(String[] args)
    {
        method();
    }
```

```
        static void method()
        {
            try
            {
                System.out.println(“Hello”);
            }
            finally
            {
                System.out.println(“good-bye”);
            }
        }
    }
```

编译运行后，输出结果是（）。（选择一项）

a) ”Hello”
b) ”good-bye”
”Hello”
c) “good-bye”
d) 代码不能编译

【解析】

69) 下面选项中，（）可以用来在HTML中嵌入APPLET程序。（选择一项）

a) <applet>..</applet>
b) <title></titile>
c)
</br>
d)

【解析】

70) JAVA中，按下和释放鼠标按钮的操作处理（）事件。（选择一项）

a) mouseEnterd
b) mouseExitied
c) mousePressed
d) mouseClicked

【解析】

71)
```
import java,applet,Applet;
import java.awt.*;
public class ImageDemo extends Applet
{
    Image img;
    public void init()
    {
        img=getImage(getCodeBase(),"11.gif");   //1
    }
    public void paint(Graphics g)
    {
        int w=img.getWidth(this);
        int h=img.getHeight(this);
        g.drawImage(img,120,60,w/2,h/2,this); //2
        g.drawImage(img,150,0,w*2,h*2,this);//3
    }
}
```

以上JAVA代码第（）行将图像放大。（选择一项）

a) 1
b) 2
c) 3
d) 代码并没有将图像放大

72) 在JAVA编程中，分析下面的Java代码：

```
import java.net.*;
class hello implements Runnable
{
        Public static void main (String args[])
        {
             Thread t=new Thread (this);
                 t.start ( );
        }
        Public void run ( )
        {
                 System.out.println （“你好”）;
         }
}
```

以下说法正确的是（ ）。（选择一项）

a) 此程序不能通过编译
b) 该程序运行时没有任何输出结果
c) 输出“你好”
d) 该程序能够通过编译，但运行时出错

【解析】

73) MyProgram.Java被编译后，生成（ ）。（选择一项）

a) MyProgram. Obj
b) MyProgram. class
c) MyProgram.exe
d) MyProgram. bat

【解析】

74) JAVA中，访问修饰符限制性最高的是（ ）。（选择一项）

a) private
b) protected
c) public
d) friendly

【解析】

75) 在JAVA编程中，关于Swing包中的组件，下面（ ）是正确的。（选择一项）

a) Swing中的每个组件都是采用MVC模式设计的
b) JFrame窗口的关闭按钮默认不能使窗口关闭
c) Swing的组件和awt组件，在编程时不能混合使用
d) 以上都正确

【解析】

76) 在Java语言中，下列组件可以让用户选择多个选项有（ ）。（选择两项）

a) Checkbox
b) Radiobutton
c) List
d) Choice

【解析】

77) 在Java语言中，在包Package1中包含包Package2,类Class_A直接隶属于包Package1,类Class_B直接隶属于包Package2。在类Class_C要应用Class_A的方法A和Class_B的方法B,你需要（）语句。（选择两项）

a) import Package1.*;
b) import Package1.Package2.*;
c) import Package2.*;
d) import Package2.Package1.*;

【解析】

78) 在Java语言中，如果你有下面的类定义：

```
abstract class Shape {
    abstract void draw();
}
class Square extends Shape{}
```

如果你试图编译上面的代码会发生（）。（选择一项）

a) 一切成功编译
b) Shape可以编译Square不能编译
c) Square可以编译Shape不能编译
d) Shape.Square都不能编译

【解析】

79)
```
int[] my_Array;
my_Array=new int[5];
for(int count=0;count<=5;count++)
    System.out.println(my_Array[count]);
```

以上Java代码运行的结果是（）。（选择一项）

a) 将1，2，3，4，5输出到屏幕
b) 将0，1，2，3，4输出到屏幕
c) 将0，1，2，3，4，5输出到屏幕
d) 将出现运行时异常

【解析】

80) 下面描述（）是正确的。（选择两项）

a) Java的源程序必须以”.java”或”.jav”扩展名保存
b) Java VM可以是软件也可以是硬件
c) Java使用解释器执行代码
d) 用高级语言编写的代码可以让计算机理解并执行它们

【解析】

81) 下列选项中，属于Java语言的关键字的是（）。（选择两项）

a) goto
b) malloc
c) extends
d) FALSE

【解析】

82) 在Java中，关键字（）使类不能派生出子类。（选择一项）

a) final
b) public
c) private
d) volatile
e) native

【解析】

83) 在JAVA编程中，Java编译器会将java程序转换为（）。（选择一项）

a) 字节码
b) 可执行代码
c) 机器代码
d) 以上所有选项都不正确

【解析】

84) 在JAVA编程中，以下（）命令用来执行java类文件。（选择一项）

a) javac
b) java
c) appletviewer
d) 以上所有选项都不正确

【解析】

85) 在Java语言中，下面变量命名合法的有（）。（选择两项）
a) variable123
b) 123variable
c) private
d) selg_asd
【解析】

86) String s1 = new String(“Hello”);
String s2 = new String(“there”);
String s3 = new String();
上面是Java程序中的一些声明，选项中能通过编译的是（）。（选择一项）
e) s3 = s1+ s2
f) s3 = s1 & s2
g) s3 = s1 || s2
h) s3 = s1 && s2
【解析】

87) 在JAVA编程中，Java具有下列（）特点。（选择三项）
a) 面向对象
b) 跨平台
c) 安全
d) 集中式体系结构
e) 可编译成机器代码
【解析】

88) 在Java中，欲定义某类所在的包外的所有类都能访问这个类，则应用的关键字是（）。
a) protected
b) private
c) public
【解析】

89) 在Java中，根据你的理解，下列方法（）可能是类Orange的构造函数。（选择三项）
a) Orange(){…}
b) Orange(…){…}
c) Public void Orange(){…}
d) Public Orange(){…}
e) Public OrangeConstuctor(){…}
【解析】

90) 在Java中，下列选项表示字符”a”值的是（）。（选择一项）
a) ‘a’
b) “a”
c) new Character(a)
d) \000a
【解析】

91) 在JAVA语言中，包pack1的类class1 中有成员方法：protected void method_1(){…},private void method_2(){ …},public void method_3(){ ….}和void method_4(){…},在包pack2中的类class2 不是class1的子类，你在class2 中可以调用的方法有（）。（选择一项）
a) method_1
b) method_2
c) method_3
d) method_4
【解析】

92) JAVA中，使用（）修饰符时，一个类能被同一包或不同包中的其他类访问。（选择一项）
a) private
b) protected
c) public
d) frientdly
【解析】

93) 在JAVA中，类Worker是类Person的子类，Worker的构造方法中有一句“super()“，该语句（）。（选择一项）
a) 调用类Worker中定义的super()方法
b) 调用类Person中定义的super()方法
c) 调用类person的构造函数
d) 语法错误
【解析】

94) 在JAVA中，类Animal中的方法printA()定义如下：

```
public void printA(){
    int a=10;
    int result=10%3;
    System.out.println(result);
```

在类Dog中方法printA()定义如下：

```
public void printA(){
    int a=10;
    Systme.out.println(a/3);
}
```

Dog类的定义如下：

```
    Class Dog extends Animal{…}
```

若有语句：

```
Animal animal=new Dog();
Animal.printA();
```

则这段代码输出为（）。（选择一项）
a) 0
b) 3.3333
c) 2
d) 3
【解析】

95) 在JAVA语言中，下面关于类的描述正确的是（）。（选择一项）
a) 一个子类可以有多个超类
b) 一个超类可以有多个子类
c) 子类可以使用超类的所有
d) 子类一定比超类有更多的成员方法
【解析】

96) JAVA程序中，（）不能用来表示注释。（选择一项）
a) //注释
b) /*注释*/
c) /**注释*/
d) /注释/
【解析】

97) 在JAVA编程中，（）可以实现跳转结构。（选择一项）
a) break
b) while
c) do-while
d) for
【解析】

98) 在JAVA中，关于捕获错误的语法try-catch-finally的下列描述正确的是（）。（选择两项）

a) try-catch必须配对使用
b) try可以单独使用
c) try-finally可以配对使用，finally也可单独使用
d) 在try-catch后如果定义了finally,则finally肯定会执行

【解析】

99) 分析下列java代码：

```
class A
{
     public static void main(String[] args)
     {
         method();
     }
     static void method()
     {
         try
         {
             System.out.println(“Hello”);
         }
         finally
         {
             System.out.println(“good-bye”);
         }
     }
}
```

编译运行后，输出结果是（）。（选择一项）

a) ”Hello”
b) ”good-bye”
c) “Hello”
“good-bye”
d) 代码不能编译

【解析】

100) JAVA中，访问修饰符限制性最高的是（ ）。（选择一项）

a) private
b) protected
c) public
d) friendly

【解析】

101）下面哪些类属于java.lang包中的类（） （选择二项）

a) StringBuffer
b) Date
c) Math
d) Exception

【解析】

102) 下面语句正确的是（）（选择二项）

a) Calendar obj1=new Calendar()
b) Calendar obj=Calendar.getInstance()
c) Date obj2= new Date()
d) Date obj2= Calendar.getInstance()

【解析】

103）( )用于创建动态数组（选择 一项）

a) ArrayList
b) HashMap
c) LinkedList
d) HashTable

【解析】