什么是GC？为什么要有GC？Java是如何实现GC的？

GC是垃圾收集的意思（Gabage Collection）,内存处理是编程人员容易出现问题的地方，忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃，Java提供的GC功能可以 自动监测对象是否超过作用域从而达到自动回收内存的目的，Java语言没有提供释放已分配内存的显示操作方法

下面程序运行的结果是（   ）

class Example{

    public static void main(String[] args){

        int i = 45678;

        int j = ~i;

        System.out.println(j);

   }

}

选择一项：

A. 输出 -45679

B. 输出-45677

C. 编译错误

D. 输出45677

MySQL数据库的JDBC驱动程序类名及完整包路径为

com.mysql.jdbc.Driver

哪些不是Java关键字? (    )

选择一项或多项：

A. void

B. sizeof

sizeof是C/C++/C#等语言的的方法，java中没有这个方法，可以使用length

C. super

D. const

E. TRUE

java是严格区分大小写的，应为true

下面哪个方法可以通知Java虚拟机调用垃圾回收器来回收已废弃对象？（   ）

选择一项：

A. Runtime.gc()

B. System.freeMemory

C. System.gc()

D. Runtime.getRuntime().growHeap()

反馈

正确答案是：

System.gc()

开发与运行Java程序需要经过的三个主要步骤为(  编辑源程序   )、(      ) 和( )。

选择一项或多项：

A. 执行程序

B. 编译生成字节码

C. 解释运行字节码

D. 调试源文件

一个java Application 运行后，在系统中是作为一个(     )

选择一项：

A. 线程

B. 进程

C. 进程或线程

D. 不可预知

为实现多线程之间的通信，需要使用下列那种流才合适？（   ）

选择一项：

A. Filter stream

B. Random access stream

C. Pipde stream

D. File stream

下列程序的功能是在控制台上，每隔1秒钟显示字符串”Hello”,能够填写在程序中下划线位置，使程序完整，并能够正确运行的语句是(        )

public class  Test implements Runnable{

         public static void main(String[] args){

                   Test t=new Test();

                   Thread tt=new Thread(t);

                   tt.start();

         }

         public void run(){

                   for(;;){

                            try{

                                \_\_\_\_\_\_\_\_\_\_

                            }

                            catch(\_\_\_\_\_\_\_\_\_  e ){}

                            System.out.println(“Hello”);

                   }

         }

}

选择一项：

A. t. sleep (1000)                InterruptedException

B. sleep (1000)             InterruptedException

C. Thread. sleep (1000)       InterruptedException

D. Thread. sleep (1000)       RuntimeException

下列代码运行的结果为（    ）

public class Example implements Runnable {

    public void run() {

        System.out.print(“running ”);

    }

    public static void main(String[] args){

        Thread t = new Thread(new Example());

        t.run();

        t.run();

        t.start();

    }

}

选择一项：

A. running running running

B. running running

C. 编译错误

D. 运行异常

下列代码运行的结果为（   ）

public class Example extends Thread{

    public void run() {

        System.out.print (“Before start method ”);

        this.stop();

        System.out.println(“After stop method”);

    }

    public static void main(String[] args){

        Example a = new Example();

        a.start();

    }

}

选择一项：

A. 编译错误

B. Before start method After stop method

C. 运行异常

D. Before start method

当使用Thread t=new Thread(r)创建一个线程时，表达式：r instanceof Thread的值为 。

### 反馈

r instanceof Runnable 才是 true

设计一个线程，使得字符串”大连是一个美丽的海滨城市” 每隔1秒钟输出一个字符。

         运行结果如下：(多行输出)

         大

         大连

         大连是

         ……

         大连是一个美丽的海滨城市

class WordDisplay extends Thread{

         private String text="大连是一个美丽的海滨城市";

         public void run(){

                   for(int i=1;i<=text.length();i++){

                            try{

                                     System.out.println(text.substring(0,i));

                                     sleep(1000);//休眠1s

                            }

                            catch(InterruptedException e){

                                     e.print

                            }

                   }

         }

}

或

class WordDisplay implements Runnable{

         private String text="大连是一个美丽的海滨城市";

         public void run(){

                   for(int i=1;i<=text.length();i++){

                            try{

                                     System.out.println(text.substring(0,i));

                                     sleep(1000);//休眠1s

                            }

                            catch(InterruptedException e){

                            }

                   }

         }

}

线程的基本概念、线程的基本状态以及状态之间的关系？

线程指在程序执行过程中，能够执行程序代码的一个执行单位，每个程序至少都有一个线程，也就是程序本身。

Java中的线程有四种状态分别是：运行、就绪、挂起、结束。

多线程有几种实现方法,都是什么?同步有几种实现方法,都是什么?

多线程有两种实现方法，分别是继承Thread类与实现Runnable接口

同步的实现方面有两种，分别是synchronized,wait与notify

String是最基本的数据类型吗?简述对String的理解。

基本数据类型包括byte、int、char、long、float、double、boolean和short。

java.lang.String类是final类型的，因此不可以继承这个类、不能修改这个类。为了提高效率节省空间，我们应该用StringBuffer类

下面哪一个方法是属于Math类的？（  ）

选择一项：

A. log()

B. sine()

C. absolute()

D. cosine()

下面程序运行的结果是（  ）

public class Example{

    int x = 12;

    public void method(int x){

        x += x;

        System.out.println(x);

    }

    public static void main(String[] args){

        Example t = new Example();

        t.method(5);

    }

}

选择一项：

A. 24

B. 10

C. 17

D. 12

E. 5

运行时异常与一般异常有何异同？

异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机的通常操作中可能遇到的异常，是一种常见运行错误。java编译器要求方法必须声明抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。

error和exception有什么区别?

error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。

exception 表示一种设计或实现问题。也就是说，它表示如果程序运行正常，从不会发生的情况。

final, finally, finalize的区别？

final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。  
finally是异常处理语句结构的一部分，表示总是执行。  
finalize是Object类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等

在下列选项中，t变量的值为（    ）

String s = “hypertext”;

String t = s.substring(2,5);

选择一项：

A. “ype”

B. “per”

C. “yper”

D. “pert”

反馈

正确答案是：

“per”

**String 和StringBuffer的区别**

JAVA平台提供了两个类：String和StringBuffer，它们可以储存和操作字符串，即包含多个字符的字符数据。这个String类提供了数值不可改变的字符串。而这个StringBuffer类提供的字符串进行修改。当你知道字符数据要改变的时候你就可以使用StringBuffer。典型地，你可以使用StringBuffers来动态构造字符数据。

下面代码运行的结果为（     ）

class Example{

    public static void main(String[] args){

        String str = "Welcome";

        str.concat("to Java!");

        System.out.println(str);

    }

}

选择一项：

A. Prints “Welcometo Java! ”

B. 运行异常

C. Prints “Welcome”

D. 编译错误

反馈

正确答案是：

Prints “Welcome”

数组有没有length()这个方法? String有没有length()这个方法？

数组没有length()这个方法，有length的属性。String有有length()这个方法

此方法返回表示由字符串参数的字符此对象字符拼接字符串。

Str 并没有接收这个返回值

FileInputStream是字节流；BufferedWriter是字符流；ObjectOutputStream是(      )对象流

以下那个是java.io包中一个兼有输入输出功能的类(    )

选择一项：

A. Java.io中不存在这样的类

B. Object

C. RandomAccessFile

D. Serializable

Character流与Byte流的区别是 (     )

选择一项：

A. 前者带有缓冲，后者没有

B. 二者没有区别，可以互换使用

C. 前者是字符读写，后者是字节读写

D. 每次读入的字节数不同

下列不是InputStream子类的是（ ）

选择一项：

A. FileInputStream

B. ZipInputStream

C. ObjectInputStream

D. CharInputStream

在读字符文件Post.dat时，使用该文件作为参数的类(     )

选择一项：

A. DataOutputStream

B. FileInputStream

C. BufferedReader

D. DataInputStream

反馈

正确答案是：

FileInputStream

在服务器程序中，线程在那些情况下可能会进入阻塞状态？多选(       )

选择一项或多项：

A. 线程执行Socket的getOutputStream()方法获得输出流

B. 线程执行ServerSocket的Accept()方法

C. 线程执行Socket的输入流读取数据

D. 线程执行Socket的输出流写一批数据

E. 线程执行Socket的getInputStream()方法获得输入流

简述TCP和UDP的工作原理和区别。

tcp协议和udp协议的差别

TCP UDP

是否连接 面向连接 面向非连接

传输可靠性 可靠 不可靠

应用场合 传输大量数据 少量数据

速度 慢 快

默认情况下，SocketChannel处于什么模式(     )

选择一项：

A. 非阻塞模式

B. 阻塞模式

URLConnection openConnection(); //建立URL网络连接

myURL = new URL("http://java.sun.com");

getProtocol 协议

getHost

getFile

getPort

getRef

阅读下列代码后    
public class Person{    
      int arr[]=new int[10];    
      public static void main(String args[]){    
            System.out.println(arr[1]);    
      }    
}    
正确的说法是(   )。

选择一项：

A. 编译时将产生错误

静态方法中无法访问非静态的成员

B. 编译时正确，运行时将产生错误

C. 输出空

D. 输出零

设有数组定义：int   MyIntArray[ ] = { 10 , 20 , 30 , 40 , 50 , 60 , 70};   则执行以下几个语句后的输出结果是(    ) 。  
    int  s = 0 ;  
    for  ( int  i = 0 ; i < MyIntArray.length ; i + + )  
                 if  ( i % 2 = = 1 )

                            s += MyIntArray[i] ;  
   System.out.println( s );

120

将键值对(key--value)保存到映射(Map)对象中的方法是（    ）。

选择一项：

A. get(Object key,Object value)

B. insert(Object key,Object value)

C. add(Object key,Object value)

D. put(Object key,Object value)

有时间再看看它的方法

下列数组a中，能够在程序运行时动态调整大小的是(    )

选择一项：

A. A=new Array()

B. A=new ArrayList()

C. String[] a

D. int[] a

集合类保存在类包（　      ）

选择一项：

A. java.text

B. java.lang

C. java.util

D. java.io

集合分为和\_\_\_\_\_\_\_\_\_\_\_两个最顶端的接口。

反馈

正确答案是：Collection Map

获得集合中元素总数的方法是

abstract class和interface有什么区别?

声明方法的存在而不去实现它的类被叫做抽象类（abstract class），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该类的情况。不能创建abstract 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。

接 口（interface）是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有 程序体。接口只可以定义static final成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。 然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到 接口类型或从接口类型转换，instanceof 运算符可以用来决定某对象的类是否实现了接口。

接口是否可继承接口? 抽象类是否可实现(implements)接口? 抽象类是否可继承实体类(concrete class)?

接口可以继承接口。抽象类可以实现(implements)接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数

Overload和Override的区别。Overloaded的方法是否可以改变返回值的类型?

方法的重写Overriding和重载Overloading是Java多态性的不同表现。重写Overriding是父类与子类之间多态性的一种表现，重载Overloading是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被“屏蔽”了。如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载(Overloading)。Overloaded的方法是可以改变返回值的类型。

**class** V1 {

**public** **int** x = 4;

         V1() {

                   System.out.println("V1");

         }

**void** display(){

                   System.out.println("V1 方法");

         }

}

**class** V2 **extends** V1 {

**public** **int** x = 7;

         V2() {

                   System.out.println("V2");

         }

**void** display(){

                   System.out.println("V2 方法");

         }

}

**class** A {

**public** **static** **void** main(String[] s) {

                   V2 e2 = new V2();  
                   V1 e1= e2;  
                   System.out.println(e1.x);  
                   System.out.println(e2.x);  
                   e1.display();

         }

}

V1  
V2  
4  
7  
V2 方法

下列哪个类声明是正确的？(  )。

选择一项：

A. protected private number;

B. abstract final class HI{···｝

C. abstract private move(){···}

D. public abstract class Car{···}

反馈

正确答案是：

public abstract class Car{···}

在Java中，用Package语句说明一个包时，该包的层次结构必须是(      )。

选择一项：

A. 与文件大小相同

B. 与文件的结构相同

C. 与文件目录的层次相同

D. 与文件类型相同

在Java中，由Java编译器自动导入，而无需在程序中用import导入的包是(     )。

选择一项：

A. java.applet

B. java.util

C. java.awt

D. java.lang

不允许作为类的访问控制符的是(   )。

选择一项：

A. static

只有 缺省，public, abstract ， final 可以用来修饰类。

B. abstract

C.  final

D. public

为AB类的一个无形式参数无返回值的方法method书写方法头，使得使用类名AB作为前缀就可以调用它，该方法头的形式为(      )。

选择一项：

A. public  void  method( )

B. abstract  void  method( )

C. final  void  method( )

D. static  void  method( )

创建一个名为 MyPackage 的包的语句是(     )，该语句应该放在程序的位置为：(      )。

答案：



反馈

正确答案是：package MyPackage，应该在程序第一句

简述java中的访问修饰符及其访问范围。

默认(default):此情况下,包外的类不能对其进行访问.只能是包内的调用.  
保护(protected):包外的子类可以对其进行访问.  
私有(private):都不能访问.只能内部访问.(一般会通过内部购建的方法创建对象)  
公共(public):都可以访问.

简述Java中接口的概念及接口的性质？

Java语言规定，类只能实现单继承，但接口可以实现多重继承。接口中定义的方法都是抽象方法，定义的所有域都是常量。

因此接口可以看作是没有实现的方法和常量的集合。使用接口主要是解决Java语言不支持多重继承的问题。

性质：①接口不提供代码重用性。因为接口中的方法都是没有实现的方法，因此所有的方法体都要自己编写。

②接口可以通过多继承进行扩展。

③接口不能升级，即提供给用户的接口不能再改变。

int型public成员变量MAX\_LENGTH，该值保持为常数100，则定义这个变量的语句是(      )。

选择一项：

A. public int MAX\_LENGTH=100

B. final int MAX\_LENGTH=100

C. public const int MAX\_LENGTH=100

D. public final int MAX\_LENGTH=100

int 和 Integer 有什么区别？

Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。Int是java的原始数据类型，Integer是java为int提供的封装类。Java为每个原始类型提供了封装类。引用类型和原始类型的行为完全不同，并且它们具有不同的语义。引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型的数据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。对象引用实例变量的缺省值为 null，而原始类型实例变量的缺省值与它们的类型有关。

什么是类与对象？

所谓对象就是真实世界中的实体，对象与实体是一一对应的，也就是说现实世界中每一个实体都是一个对象，它是一种具体的概念。

类是具备某些共同特征的实体的集合，它是一种抽象的概念，用程序设计的语言来说，类是一种抽象的数据类型，它是对所具有相同特征实体的抽象。

面向对象的特征有哪些方面？并简述每一个特征。

1)      抽象：

抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。

2)      继承：

继 承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继 承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增 加新的方法使之更适合特殊的需要。

3)      封装：

封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。

只要有足够的方法，就没必要直接去操作对象属性，只要调用这些方法就可以实现要完成的任务，这种现象称为封装，它通过对象方法对其属性的操作把对象属性封装在一个对象内部，对象与外界打交道全部通过其自身的方法来实现，有效的把对象属性隐藏在对象内部。

4)      多态性：

多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。

构造器Constructor是否可被override?

构造器Constructor不能被继承，因此不能重写Overriding，但可以被重载Overloading。

简述Java中构造方法的概念及特点。

构造方法：构造方法是一种特殊的方法，用于初始化对象。

特点：①构造方法与类同名，可以有参数 ，也可无参数。

②构造方法没有返回值，在方法前也不需要写void。

③创建对象时，系统自动调用构造方法。

④用户在程序中不能直接调用构造方法，只能通过new操作符调用。

⑤一个类可以有多个同名构造方法，但参数必须不同。所谓参数不同是指：

⑥类中可以不定义构造方法，此时Java提供一个默认的无参数构造方法

char类型的取值范围是 (      )。

选择一项：

A. 0～216-1

B. -215 ～215 -1

C. 0～28-1

D. -2 ^7 ～ 2 ^7-1

反馈

正确答案是：

0～216-1

设 x = 1 , y = 2 , z = 3，则表达式  y＋＝z－－/＋＋x  的值是(    )。

选择一项：

A. 4

B. 3.5

C. 3

D. 5

在Java的基本数据类型中，char型采用Unicode编码方案，每个Unicode码占用(      )字节内存空间，这样，无论是中文字符还是英文字符，都是占用(      )字节内存空间。

选择一项：

A. 2,4

B. 2,2

C. 4,4

D. 1,1

设 x = 2 ，则表达式 （x++)/3 的值是(      )。

选择一项：

A. 1

B. 2

C. 0

D. 3

Java的基本数据类型有哪些？并写出整型数据的取值范围。

–整数类型  byte, short, int, long

–浮点类型  double, float

–字符类型  char

布尔类型  boolean

|  |  |  |
| --- | --- | --- |
| 类型 | 存储需求 | 取值范围 |
| int | 4字节 | -231~231-1 |
| short | 2字节 | -215~215-1 |
| long | 8字节 | -263~263-1 |
| byte | 1字节 | -128~127 |

简述&和&&的区别。

java中，＆连接两个数字的时候，它是表示两个位的按位与，而&&表示两个条件的并列

break语句最常见的用法是在switch语句中，通过break语句退出switch语句，使程序从整个switch 语句后面的 (      ) 开始执行。

选择一项：

A. break

B. 第一条语句

C. 不执行

D. default