一. 选择题（234）

1.下面中哪两个可以在 A 的子类中使用：（　　）

```
class A {
protected int method1 (int a,  int b) {
return 0;
}
}
```

A. public int method 1 (int a, int b) { return 0; }

B. private int method1 (int a, int b) { return 0; }

C. private int method1 (int a, long b) { return 0; }

D. public short method1 (int a, int b) { return 0; }

解答：AC

主要考查子类重写父类的方法的原则

B，子类重写父类的方法，访问权限不能降低

C，属于重载

D，子类重写父类的方法 返回值类型要相同或是父类方法返回值类型的子类

2.Abstract method cannot be static. True or False ?

A True

B False

解答：A

抽象方法可以在子类中被重写，但是静态方法不能在子类中被重写，静态方法和静态属性与对象是无关的，只与类有关，这与 abstract 是矛盾的，所以 abstract 是不能被修饰为 static，否则就失去了 abstract 的意义了

3.What will be the output when you compile and execute the following program.

```
class Base
{
void test() {
```

```
System.out.println("Base.test()");
}
}
public class Child extends Base {
void test() {
System.out.println("Child.test()");
}
static public void main(String[] a) {
Child anObj = new Child();
Base baseObj = (Base)anObj;
baseObj.test();
}
}
```

Select most appropriate answer.

A Child.test()
Base.test()

B Base.test()
Child.test()

C Base.test()

D Child.test()

解答：D

测试代码相当于：Base baseObj = new Child();父类的引用指向子类的实例，子类又重写了父类的 test 方法，因此调用子类的 test 方法。

4.What will be the output when you compile and execute the following program.

```
class Base
{
static void test() {
System.out.println("Base.test()");
}
}
```

```
public class Child extends Base {
void test() {
System.out.println("Child.test()");
Base.test(); //Call the parent method
}

static public void main(String[] a) {
new Child().test();
}
}
```

Select most appropriate answer.

A Child.test()
Base.test()

B Child.test()
Child.test()

C Compilation error. Cannot override a static method by an instance method

D Runtime error. Cannot override a static method by an instance method

解答：C

静态方法不能在子类中被重写

5.What will be the output when you compile and execute the following program.

```
public class Base{
private void test() {
System.out.println(6 + 6 + "(Result)");
}
static public void main(String[] a) {
new Base().test();
}
}
```

Select most appropriate answer.

A 66(Result)

B 12(Result)

C Runtime Error.Incompatible type for +. Can't convert an int to a string.

D Compilation Error.Incompatible type for +. Can't add a string to an int.

解答：B

字符串与基本数据类型链接的问题,如果第一个是字符串那么后续就都按字符串处理，比如上边例子要是 System.out.println("(Result)"+6 + 6);那么结果就是(Result)66，如果第一个和第二个。。。第 n 个都是基本数据第 n+1 是字符串类型，那么前 n 个都按加法计算出结果在与字符串连接

6..What will be the output when you compile and execute the following program. The symbol ' ' means space.

```
1:public class Base{
2:
3:   private void test() {
4:
5:    String aStr = "  One  ";
6:    String bStr = aStr;
7:    aStr.toUpperCase();
8:    aStr.trim();
9:    System.out.println("[" + aStr + "," + bStr + "]");
7:   }
8:
9:   static public void main(String[] a) {
10:   new Base().test();
11:  }
12: }
```

Select most appropriate answer.

A [ONE, One]

B [One ,One]

C [ONE,One]

D [ONE,ONE]

E [One , One]

解答：E

通过 String bStr = aStr;这句代码使 bStr 和 aStr 指向同一个地址空间，所以最后 aStr 和 bStr 的结果应该是一样，String 类是定长字符串，调用一个字符串的方法以后会形成一个新的字符串。

7.下面关于变量及其范围的陈述哪些是不正确的（　　）：

A. 实例变量是类的成员变量

B. 实例变量用关键字 static 声明

C. 在方法中定义的局部变量在该方法被执行时创建

D. 局部变量在使用前必须被初始化

解答：BC

由 static 修饰的变量称为类变量或是静态变量

方法加载的时候创建局部变量

8.下列关于修饰符混用的说法，错误的是（　）：

A. abstract 不能与 final 并列修饰同一个类

B. abstract 类中可以有 private 的成员

C. abstract 方法必须在 abstract 类中

D. static 方法中能处理非 static 的属性

解答 D

:静态方法中不能引用非静态的成员

9.执行完以下代码 int [ ] x = new int[25]；后，以下哪项说明是正确的（　　）：

A、　x[24]为 0

B、　x[24]未定义

C、　x[25]为 0

D、　x[0]为空

解答：A

x 属于引用类型，该引用类型的每一个成员是 int 类型，默认值为：0

10.编译运行以下程序后，关于输出结果的说明正确的是 （　　）：

```
public  class   Conditional{
        public  static  void  main(String  args[  ]){
                int  x=4;
                System.out.println(“value  is  “+ ((x>4) ? 99.9 :9));
```

}

}

A、 输出结果为：value is 99.99

B、 输出结果为：value is 9

C、 输出结果为：value is 9.0

D、 编译错误

解答：C

三目运算符中：第二个表达式和第三个表达式中如果都为基本数据类型，整个表达式的运算结果由容量高的决定。99.9 是 double 类型 而 9 是 int 类型，double 容量高。

11.关于以下 application 的说明，正确的是（ ）：

```
1.  class  StaticStuff
2.  {
3.                  static  int  x=10;
4.                  static  { x+=5; }
5.                  public  static  void  main (String  args[ ])
6.                  {
7.                       System.out.println(“x=” + x);
8.                  }
9.                  static  { x/=3;}
10.  }
```

A、 4 行与 9 行不能通过编译，因为缺少方法名和返回类型

B、 9 行不能通过编译，因为只能有一个静态初始化器

C、 编译通过，执行结果为：x=5

D、编译通过，执行结果为：x=3

解答：C

自由块是类加载的时候就会被执行到的，自由块的执行顺序是按照在类中出现的先后顺序执行。

12.关于以下程序代码的说明正确的是（ ）：

```
1. class  HasStatic{
2.     private  static  int  x=100;
3.     public  static  void  main(String  args[ ]){
```

```
4.          HasStatic  hs1=new  HasStatic(  );
5.          hs1.x++;
6.          HasStatic  hs2=new  HasStatic(  );
7.          hs2.x++;
8.          hs1=new  HasStatic( );
9.          hs1.x++;
10.         HasStatic.x--;
11.         System.out.println(“x=”+x);
12.     }
13. }
```

A、5 行不能通过编译，因为引用了私有静态变量

B、10 行不能通过编译，因为 x 是私有静态变量

C、程序通过编译，输出结果为：x=103

D、程序通过编译，输出结果为：x=102

解答：D

静态变量是所有对象所共享的，所以上述代码中的几个对象操作是同一静态变量 x， 静态变量可以通过类名调用。

13.下列说法正确的有（）

A. class 中的 constructor 不可省略

B. constructor 必须与 class 同名，但方法不能与 class 同名

C. constructor 在一个对象被 new 时执行

D. 一个 class 只能定义一个 constructor

解答：C

构造方法的作用是在实例化对象的时候给数据成员进行初始化

A. 类中如果没有显示的给出构造方法，系统会提供一个无参构造方法

B. 构造方法与类同名，类中可以有和类名相同的方法

D. 构造方法可以重载

14.下列哪种说法是正确的（）

A. 实例方法可直接调用超类的实例方法

B. 实例方法可直接调用超类的类方法

C. 实例方法可直接调用其他类的实例方法

D. 实例方法可直接调用本类的类方法

解答：D

A. 实例方法不可直接调用超类的私有实例方法

B. 实例方法不可直接调用超类的私有的类方法

C. 要看访问权限

15.下列哪一种叙述是正确的（ ）

A. abstract 修饰符可修饰字段、方法和类

B. 抽象方法的 body 部分必须用一对大括号{ }包住

C. 声明抽象方法，大括号可有可无

D. 声明抽象方法不可写出大括号

解答：D

abstract 可以修饰方法和类，不能修饰属性。抽象方法没有方法体，即没有大括号{}

16.下面代码的执行结果是？

```
import java.util.*;
public class ShortSet{
    public static void main(String args[])
    {
         Set<Short> s=new HashSet<Short>();
         for(Short i=0;i<100;i++)
         {
          s.add(i);
          s.remove(i-1);
         }
         System.out.println(s.size());
    }
}
```

A.1

B.100

C.Throws Exception

D.None of the Above

解答：B

i 是 Short 类型　i-1 是 int 类型,其包装类为 Integer,所以 s.remove(i-1);不能移除 Set 集合中 Short 类型对象。

17.链表具有的特点是：(选择 3 项)

A、不必事先估计存储空间

B、可随机访问任一元素

C、插入删除不需要移动元素

D、所需空间与线性表长度成正比

解答：ACD

A.采用动态存储分配，不会造成内存浪费和溢出。

B. 不能随机访问，查找时要从头指针开始遍历

C. 插入、删除时，只要找到对应前驱结点，修改指针即可，无需移动元素

D. 需要用额外空间存储线性表的关系，存储密度小

18.Java 语言中，String 类的 IndexOf()方法返回的类型是？

A、Int16　B、Int32　C、int D、long

解答：C

indexOf 方法的声明为：public int indexOf(int ch)

在此对象表示的字符序列中第一次出现该字符的索引；如果未出现该字符，则返回 -1。

19.以下关于面向对象概念的描述中，不正确的一项是（）。(选择 1 项)

A.在现实生活中，对象是指客观世界的实体

B.程序中的对象就是现实生活中的对象

C.在程序中，对象是通过一种抽象数据类型来描述的，这种抽象数据类型称为类（class）

D.在程序中，对象是一组变量和相关方法的集合

解答：B

20..执行下列代码后,哪个结论是正确的 String[] s=new String[10];

A.　s[9] 为 null;

B.　s[10] 为 "";

C.　s[0] 为 未定义

D.　s.length 为 10

解答：AD

s 是引用类型，s 中的每一个成员都是引用类型，即 String 类型，String 类型默认的值为 null

s 数组的长度为 10。

21.属性的可见性有。(选择 3 项)

A.公有的

B.私有的

C.私有保护的

D.保护的

解答：ABD

属性的可见性有四种：公有的（public） 保护的（protected） 默认的 私有的（private）

22..在字符串前面加上_____符号，则字符串中的转义字符将不被处理。(选择 1 项)

A @

B \

C #

D %

解答：B

23.下列代码哪行会出错: (选择 1 项)

```
1) public void modify() {
    2) int I, j, k;
    3) I = 100;
    4) while ( I > 0 ) {
    5) j = I * 2;
    6) System.out.println (" The value of j is " + j );
    7) k = k + 1;
    8) I--;
    9) }
    10) }
```

A. 4

B. 6

C. 7

D. 8

解答：C

k 没有初始化就使用了

24.对记录序列{314，298，508，123，486，145}按从小到大的顺序进行插入排序，经过两趟排序后的结果为：(选择 1 项)

A {314，298，508，123，145，486}

B {298，314，508，123，486，145}

C {298，123，314，508，486，145}

D {123、298，314，508，486，145}

解答：B

插入排序算法：

```
public static void injectionSort(int[] number) {
    // 第一个元素作为一部分，对后面的部分进行循环
    for (int j = 1; j < number.length; j++) {
        int tmp = number[j];
        int i = j - 1;
        while (tmp < number[i]) {
            number[i + 1] = number[i];
            i--;
            if (i == -1)
                break;
        }
        number[i + 1] = tmp;
    }
}
```

25.栈是一种。(选择 1 项)

A 存取受限的线性结构

B 存取不受限的线性结构

C 存取受限的非线性结构

D 存取不受限的非线性结构

解答：A

栈（stack）在计算机科学中是限定仅在表尾进行插入或删除操作的线性表。

26.下列哪些语句关于内存回收的说明是正确的。(选择 1 项)

A 程序员必须创建一个线程来释放内存

B 内存回收程序负责释放无用内存

C 内存回收程序允许程序员直接释放内存

D 内存回收程序可以在指定的时间释放内存对象

解答：B

垃圾收集器在一个 Java 程序中的执行是自动的，不能强制执行，即使程序员能明确地判断出有一块内存已经无用了，是应该回收的，程序员也不能强制垃圾收集器回收该内存块。程序员唯一能做的就是通过调用 System. gc 方法来"建议"执行垃圾收集器，但其是否可以执行，什么时候执行却都是不可知的。

27.Which method must be defined by a class implementing the java.lang.Runnable interface?

A. void run()

B. public void run()

C. public void start()

D. void run(int priority)

E. public void run(int priority)

F. public void start(int priority)

解答：B

实现 Runnable 接口，接口中有一个抽象方法 run，实现类中实现该方法。

28 Given:

```
public static void main(String[] args) {
Object obj = new Object() {
public int hashCode() {
return 42;
}
};
System.out.println(obj.hashCode());
}
```

What is the result?

A. 42

B. An exception is thrown at runtime.

C. Compilation fails because of an error on line 12.

D. Compilation fails because of an error on line 16.

E. Compilation fails because of an error on line 17.

解答：A

匿名内部类覆盖 hashCode 方法。

29 Which two are reserved words in the Java programming language? (Choose two)

A. run

B. import

C. default

D. implements

解答：BD

import 导入包的保留字，implements 实现接口的保留字。

30. Which two statements are true regarding the return values of property written hashCodeand equals methods from two instances of the same class? (Choose two)

A. If the hashCode values are different, the objects might be equal.

B. If the hashCode values are the same, the object must be equal.

C. If the hashCode values are the same, the objects might be equal.

D. If the hashCode values are different, the objects must be unequal.

解答：CD

先通过 hashcode 来判断某个对象是否存放某个桶里，但这个桶里可能有很多对象，那么我们就需要再通过 equals 来在这个桶里找到我们要的对象。

31. What is the numerical range of a char?

A. 0 ... 32767

B. 0 ... 65535

C. -256 ... 255

D. -32768 ... 32767

E. Range is platform dependent.

解答：B

在 Java 中，char 是一个无符号 16 位类型，取值范围为 0 到 65535。

32. Given:

```
public class Test {
private static float[] f = new float[2];
public static void main(String args[]) {
System.out.println(“f[0] = “ + f[0]);
}
}
```

What is the result?

A. f[0] = 0

B. f[0] = 0.0

C. Compilation fails.

D. An exception is thrown at runtime.

解答：B

33. Given:

```
public class Test {
public static void main(String[] args) {
String str = NULL;
System.out.println(str);
}
}
```

What is the result?

A. NULL

B. Compilation fails.

C. The code runs with no output.

D. An exception is thrown at runtime.

解答：B

null 应该小写

34、Exhibit:

```
1.public class X implements Runnable {
2.    private int x;
3.    private int y;
4.    public static void main(String [] args) {
5.         X that = new X();
6.        (new Thread(that)).start();
7.        (new Thread(that)).start();
8.    }
9.  public synchronized void run( ){
10.      for (;;) {
11.         x++;
12.         y++;
13.        System.out.println(“x = “ + x + “, y = “ + y);
14.     }
15.    }
16.}
```

What is the result?

A. An error at line 11 causes compilation to fail.

B. Errors at lines 7 and 8 cause compilation to fail.

C. The program prints pairs of values for x and y that might not always be the same on the same line (for example, “x=2, y=1”)

D. The program prints pairs of values for x and y that are always the same on the same line (for example, “x=1, y=1”. In addition, each value appears twice (for example, “x=1, y=1” followed by “x=1, y=1”)

E. The program prints pairs of values for x and y that are always the same on the same line (for example, “x=1, y=1”. In addition, each value appears twice (for example, “x=1, y=1” followed by “x=2, y=2”)

解答：E

多线程共享相同的数据，使用 synchronized 实现数据同步。

35、Which two CANNOT directly cause a thread to stop executing? (Choose Two)

A. Existing from a synchronized block.

B. Calling the wait method on an object.

C. Calling notify method on an object.

D. Calling read method on an InputStream object.

E. Calling the SetPriority method on a Thread object.

解答：AD

stop 方法.这个方法将终止所有未结束的方法，包括 run 方法。当一个线程停止时候，他会立即释放 所有他锁住对象上的锁。这会导致对象处于不一致的状态。 当线程想终止另一个线程的时候，它无法知道何时调用 stop 是安全的，何时会导致对象被破坏。所以这个方法被弃用了。你应

该中断一个线程而不是停止他。被中断的线程会在安全的时候停止。

36、 Which two statements are true regarding the creation of a default constructor? (Choose Two)

A. The default constructor initializes method variables.

B. The default constructor invokes the no-parameter constructor of the superclass.

C. The default constructor initializes the instance variables declared in the class.

D. If a class lacks a no-parameter constructor,, but has other constructors, the compiler creates a default constructor.

E. The compiler creates a default constructor only when there are no other constructors for the class.

解答：CE

构造方法的作用是实例化对象的时候给数据成员初始化，如果类中没有显示的提供构造方法，系统会提供默认的无参构造方法，如果有了其它构造方法，默认的构造方法不再提供。

37、 Given:

```
public class OuterClass {
private double d1 = 1.0;
//insert code here
}
```

You need to insert an inner class declaration at line2. Which two inner class declarations are valid? (Choose Two)

A. static class InnerOne { public double methoda() {return d1;} }

B. static class InnerOne { static double methoda() {return d1;} }

C. private class InnerOne { public double methoda() {return d1;} }

D. protected class InnerOne { static double methoda() {return d1;} }

E. public abstract class InnerOne { public abstract double methoda(); }

解答：CE

AB.内部类可以声明为 static 的，但此时就不能再使用外层封装类的非 static 的成员变量；

D.非 static 的内部类中的成员不能声明为 static 的，只有在顶层类或 static 的内部类中才可声明 static 成员

38、 Which two declarations prevent the overriding of a method? (Choose Two)

A. final void methoda() {}

B. void final methoda() {}

C. static void methoda() {}

D. static final void methoda() {}

E. final abstract void methoda() {}

解答：AD

final 修饰方法，在子类中不能被重写。

39、Given:

```
public class Test {
public static void main (String args[]) {
  class Foo {
    public int i = 3;
 }
   Object o = (Object) new Foo();
   Foo foo = (Foo)o;
   System.out.println(foo.i);
 }
}
```

What is the result?

A. Compilation will fail.

B. Compilation will succeed and the program will print “3”

C. Compilation will succeed but the program will throw a ClassCastException at line 6.

D. Compilation will succeed but the program will throw a ClassCastException at line 7.

解答：B

局部内部类的使用

40、 Given:

```
public class Test {
public static void main (String [] args)  {
String foo = “blue”;
String bar = foo;
foo = “green”;
System.out.println(bar);
}
}
```

What is the result?

A. An exception is thrown.

B. The code will not compile.

C. The program prints “null”

D. The program prints “blue”

E. The program prints “green”

解答：D

采用 String foo = “blue”定义方式定义的字符串放在字符串池中，通过 String bar = foo;

他们指向了同一地址空间，就是同一个池子，当执行 foo = “green”; foo 指向新的地址空间。

41、Which code determines the int value foo closest to a double value bar?

A. int foo = (int) Math.max(bar);

B. int foo = (int) Math.min(bar);

C. int foo = (int) Math.abs(bar);

D. int foo = (int) Math.ceil(bar);

E. int foo = (int) Math.floor(bar);

F. int foo = (int) Math.round(bar);

解答：DEF

A B 两个选项方法是用错误，都是两个参数。

abs 方法是取 bar 的绝对值，

ceil 方法返回最小的（最接近负无穷大）double 值，该值大于等于参数，并等于某个整数。

floor 方法返回最大的（最接近正无穷大）double 值，该值小于等于参数，并等于某个整数。

round 方法　返回最接近参数的 long。

42、 Exhibit:

```
1.package foo;
2.import java.util.Vector;
3.private class MyVector extends Vector {
4.int i = 1;
5.public MyVector()  {
6.i = 2;
7. }
8.}
9.public class MyNewVector extends MyVector {
10.public MyNewVector ()  {
11. i = 4;
12.}
13.public static void main (String args [])  {
14.MyVector v = new MyNewVector();
15. }
16.}
```

The file MyNewVector.java is shown in the exhibit. What is the result?

A. Compilation will succeed.

B. Compilation will fail at line 3.

C. Compilation will fail at line 6.

D. Compilation will fail at line 9.

E. Compilation will fail at line 14.

解答：B

类 MyVector 不能是私有的

43、Given:

```
public class Test {
public static void main (String[]args) {
  String foo = args[1];
  String bar = args[2];
  String baz = args[3];
  System.out.println(“baz = ” + baz);
 }
}
```

And the output:

Baz = 2

Which command line invocation will produce the output?

A. java Test 2222

B. java Test 1 2 3 4

C. java Test 4 2 4 2

D. java Test 4 3 2 1

解答：C

数组下标从 0 开始

44、 Given:

```
1. public interface Foo{
2.int k = 4;
3. }
```

Which three are equivalent to line 2? (Choose Three)

A. final int k = 4;

B. Public int k = 4;

C. static int k = 4;

D. Private int k = 4;

E. Abstract int k = 4;

F. Volatile int k = 4;

G. Transient int k = 4;

H. protected int k = 4;

解答：BDE

static：修饰的静态变量

final 修饰的是常量

abstract 不能修饰变量

Volatile 修饰的成员变量在每次被线程访问时，都强迫从共享内存中重读该成员变量的值。而且，当成员变量发生变化时，强迫线程将变化值回写到共享内存。这样在任何时刻，两个不同的线程总是看到某个成员变量的同一个值。

Transient：对不需序列化的类的域使用 transient 关键字,以减少序列化的数据量。

int k=4 相当于 public static final int k=4; 在接口中可以不写 static final

45、 Given:

```
public class foo {
static String s;
public static void main (String[]args) {
System.out.println (“s=” + s);
}
}
```

What is the result?

A. The code compiles and “s=” is printed.

B. The code compiles and “s=null” is printed.

C. The code does not compile because string s is not initialized.

D. The code does not compile because string s cannot be referenced.

E. The code compiles, but a NullPointerException is thrown when toString is called.

解答：B

String 为禁用数据类型，引用类型数据成员的默认值为 null

46、Which two create an instance of an array? (Choose Two)

A. int[] ia = new int [15];

B. float fa = new float [20];

C. char[] ca = “Some String”;

D. Object oa = new float[20];

E. Int ia [][] = (4, 5, 6) (1, 2, 3)

解答：AD

任何类的父类都是 Object，数组也数据引用类型，Object oa = new float[20];这种写法相当于父类的用指向之类的实例。

47、Given:

```
public class ExceptionTest {
class TestException extends Exception {}
public void runTest () throws TestException {}
public void test () /* Point X*/  {
runTest ();
 }

}
```

At point X on line 4, which code can be added to make the code compile?

A. throws Exception

B. Catch (Exception e).

C. Throws RuntimeException.

D. Catch (TestException e).

E. No code is necessary.

解答：A

方法上使用 throws 抛出异常，Exception 是异常类的超类。

48、Exhibit:

```
public class SwitchTest {
public static void main (String []args)  {
System.out.println (“value =” +switchIt(4));
}
public static int switchIt(int x)  {
int j = 1;
switch (x) {
```

```
case 1: j++;
case 2: j++;
case 3: j++;
case 4: j++;
case 5: j++;
default:j++;
}
return j + x;
}
}
```

What is the output from line 3?

A. Value =3

B. Value =4

C. Value =5

D. Value =6

E. Value =7

F. Value =8

解答：F

由于 case 块没有 break 语句，那么从 case 4：向下的代码都会执行。

49、Which four types of objects can be thrown using the throw statement? (Choose Four)

A. Error

B. Event

C. Object

D. Exception

E. Throwable

F. RuntimeException

解答：ADEF

能够抛出的对象类型要是 Throwable 或是 Throwable 的子类

50. 在下面程序的第 6 行补充上下列哪个方法,会导致在编译过程中发生错误?

```
1)  class Super{
2)    public float getNum(){
3)         return 3.0f;
4)      }
}
5)      pubhc class Sub extends Super{
6)
7)   }
```

A,public float getNum(){retun 4.0f;}

B.public void getNum(){}

C.public void getNum(double d){}

D.public double getNum(float d){ retun 4.0f ;}

解答：B

方法重写的问题。子类中有和父类的方法名相同，但是参数不同，不会出编译错误，认为是子类的特有的方法，但是如果子类中方法和父类的方法名，参数，访问权限，异常都相同，只有返回值类型不同会编译不通过。

51.下面关于 import, class 和 package 的声明顺序哪个正确？（ ）

A. package, import, class

B. class, import, package

C. import, package, class

D. package, class, import

解答：A

52.下面哪个是正确的？（ ）

A. String temp [] = new String {"a" "b" "c"};

B. String temp [] = {"a" "b" "c"}

C. String temp = {"a", "b", "c"}

D. String temp [] = {"a", "b", "c"}

解答：D

53.关于 java.lang.String 类，以下描述正确的一项是（ ）

A. String 类是 final 类故不可以继承；

B. String 类是 final 类故可以继承；

C. String 类不是 final 类故不可以继承；

D. String 类不是 final 类故可以继承；

解答：A

String 类是 final 的，在 java 中 final 修饰类的不能被继承

54.关于实例方法和类方法，以下描述正确的是：（　　）

A. 实例方法只能访问实例变量

B. 类方法既可以访问类变量，也可以访问实例变量

C. 类方法只能通过类名来调用

D. 实例方法只能通过对象来调用

解答：D

A 实例方法可以访问类变量

B 类方法只能访问类变量

C 类方法可以通过对象调用

55.接口是 Java 面向对象的实现机制之一，以下说法正确的是：（　　）

A.　Java 支持多重继承，一个类可以实现多个接口；

B.　Java 只支持单重继承，一个类可以实现多个接口；

C.　Java 只支持单重继承，一个类只可以实现一个接口；

D.　Java 支持多重继承，但一个类只可以实现一个接口。

解答：B

Java 支持单重继承，一个类只能继承自另外的一个类，但是一个类可以实现多个接口。

56.下列关于 interface 的说法正确的是：（　　）

A. interface 中可以有 private 方法

B. interface 中可以有 final 方法

C. interface 中可以有 function 实现

D. interface 可以继承其他 interface

解答：D

A.接口中不可以有 private 的方法

B. 接口中不可以有 final 的方法 接口中的方法默认是 public abstract 的

C. 接口中的方法不可以有实现

57.已知 A 类被打包在 packageA ， B 类被打包在 packageB ，且 B 类被声明为 public ，且有一个成员变量 x 被声明为, protected 控制方式 。C 类也位于 packageA 包，且继承了 B 类 。则以下说话正确的是（　）

A. A 类的实例不能访问到 B 类的实例

B. A 类的实例能够访问到 B 类一个实例的 x 成员

C. C 类的实例可以访问到 B 类一个实例的 x 成员

D. C 类的实例不能访问到 B 类的实例

解答：C

不同包子类的关系， 可以访问到父类 B 的 protected 成员

58.以下程序正确的输出是（　）

```
package test;
public class FatherClass {
     public FatherClass() {
      System.out.println("FatherClass Create");
     }
}
package test;
import test.FatherClass;
public class ChildClass extends FatherClass {
     public ChildClass() {
      System.out.println("ChildClass Create");
     }

     public static void main(String[] args) {
      FatherClass fc = new FatherClass();
      ChildClass cc = new ChildClass();
     }
}
```

A.

FatherClass Create

FatherClass Create

ChildClass Create

B.

FatherClass Create

ChildClass Create

FatherClass Create

C.

ChildClass Create

ChildClass Create

FatherClass Create

D.

ChildClass Create

FatherClass Create

FatherClass Create

解答：A

在子类构造方法的开始默认情况下有一句 super();来调用父类的构造方法

59.给定如下代码，下面哪个可以作为该类的构造函数 （ ）

```
public class Test {
     …
 }
```

A. public void Test() {…}

B. public Test() {…}

C. public static Test() {…}

D. public static void Test() {…}

解答：B

构造方法：与类同名没有放回类型

60.题目:

```
1. public class test (
2.   public static void main (String args[]) {
```

```
3.      int i = 0xFFFFFFF1;
4.      int j = ~i;
5.
6.   }
7. )
```

程序运行到第 5 行时,j 的值为多少?(　　)

A. －15

B. 0

C. 1

D. 14

E. 在第三行的错误导致编译失败

解答：D

int i = 0xFFFFFFF1;相当于 int i=-15 然后对 i 进行取反即取绝对值再减一

61.关于 sleep()和 wait()，以下描述错误的一项是（　　）

A. sleep 是线程类（Thread）的方法，wait 是 Object 类的方法；

B. sleep 不释放对象锁，wait 放弃对象锁；

C. sleep 暂停线程、但监控状态仍然保持，结束后会自动恢复；

D. wait 后进入等待锁定池，只有针对此对象发出 notify 方法后获得对象锁进入运行状态。

解答：D

sleep 是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用 sleep 不会释放对象锁。

wait 是 Object 类的方法，对此对象调用 wait 方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出 notify 方法（或 notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

62.下面能让线程停止执行的有（多选）(　　)

A. sleep();

B. stop();

C. notify();

D. synchronized();

E. yield();

F. wait();

G. notifyAll();

解答：ABDEF

sleep：导致此线程暂停执行指定时间

stop: 这个方法将终止所有未结束的方法，包括 run 方法。

synchronized():对象锁

yield：当前正在被服务的线程可能觉得 cpu 的服务质量不够好，于是提前退出，这就是 yield。

wait：当前正在被服务的线程需要睡一会，醒来后继续被服务

63.下面哪个可以改变容器的布局？（　　）

A. setLayout(aLayoutManager);

B. addLayout(aLayoutManager);

C. layout(aLayoutManager);

D. setLayoutManager(aLayoutManager);

解答：A

Java 设置布局管理器 setLayout()

64.下面哪个是 applet 传递参数的正确方式？（　　）

A. <applet code=Test.class age=33 width=100 height=100>

B. <param name=age value=33>

C. <applet code=Test.class name=age value=33 width=100 height=100>

D. <applet Test 33>

解答：B

65.提供 Java 存取数据库能力的包是（）

A. java.sql

B. java.awt

C. java.lang

D. java.swing

解答：A

java.sql 是 JDBC 的编程接口

java.awt 和 java.swing 是做图像界面的类库

java.lang: Java 编程语言进行程序设计的基础类

66.不能用来修饰 interface 的有（）

A. private

B. public

C. protected

D. static

解答：ACD

修饰接口可以是 public 和默认

67.下列说法错误的有（）

A. 在类方法中可用 this 来调用本类的类方法

B. 在类方法中调用本类的类方法时可直接调用

C. 在类方法中只能调用本类中的类方法

D. 在类方法中绝对不能调用实例方法

解答：ACD

A.在类方法中不能使用 this 关键字

C. 在类方法中可以调用其它类中的类方法

D. 在类方法中可以通过实例化对象调用实例方法

68.从下面四段（A，B，C，D）代码中选择出正确的代码段（）

A. abstract class Name {

private String name;

public abstract boolean isStupidName(String name) {}

}

B. public class Something {

void doSomething () {

private String s = "";

int l = s.length();

}

}

C. public class Something {

      public static void main(String[] args) {

Other o = new Other();

```
new Something().addOne(o);
}
      public void addOne(final Other o) {
o.i++;
}
}
class Other {
public int i;
}
```

D.
```
public class Something {
public int addOne(final int x) {
return ++x;  }
}
```

解答：C

A..抽象方法不能有方法体

B. 方法中定义的是局部变量，不能用类成员变量修饰符 private

D. final 修饰为常量，常量的值不能被改变

69.选择下面代码的运行结果：()。

```
public class Test{
          public void method()
          {
                  for(int i = 0; i < 3; i++)
                  {
                      System.out.print(i);
                  }
                  System.out.print(i);
          }
}
```

A. 0122

B. 0123

C. 编译错误

D. 没有任何输出

解答：C

i 变量的作用范围是整个 for 循环

70.请看如下代码

```
class Person {
     private int a;
     public int change(int m){return m;}
    }
public class Teacher extends Person{
     public int b;
     public static void main(String arg[]){
       Person p = new Person();
       Teacher t = new Teacher();
       int i;
       // point x
      }
    }
```

下面哪些放在// point x?行是正确的?

A, i = m;

B, i = b;

C, i = p.a;

D, i = p.change(30);

E, i = t.b;

解答：DE

A.不同的作用域

B. 静态方法中不能直接使用非静态成员变量

C. 类外不能访问其它类私有的成员

D，E. 在类方法中可以通过实例化对象调用类中的实例成员。

71 下面那几个函数是 public void method(){...}的重载函数？ ()

A.public void method(int m){...}

B.public int method(){...}

C.public void method2(){...}

D.public int method(int m, float f){...}

解答：A

重载：方法名相同，参数列表不同

72.给出如下声明：

String s = “Example”;

合法的代码由哪些？

A）s>>>=3　B）s[3]= “X”　C）int i = s.iength()　D）s = s +10

解答：D

A.　移位运算，要是整数类型。

B. s 不是数组

C. String 类取长度的方法为：length()

D. 字符串相加

73.如下哪些不是 java 的关键字？（ ）

A.const

B.NULL

C.false

D.this

E.native

解答：BC

虽然 null false 还有 true 不是 java 的关键字，但是都有特殊用途，不建议作为标识符。

74.已知表达式 int m [ ] = {0, 1, 2, 3, 4, 5, 6};

下面哪个表达式的值与数组下标量总数相等？（ ）

A .m.length()

B.m.length

C.m.length()+1

D.m.length+1

解答：B

解答：数组下标是从零开始的，但是数据下标的总量和数据长度相同。

75.方法 resume()负责恢复哪些线程的执行（ ）

A 通过调用 stop()方法而停止的线程。

B 通过调用 sleep()方法而停止的线程。

C 通过调用 wait()方法而停止的线程。

D 通过调用 suspend()方法而停止的线程。

解答：D

Suspend 可以挂起一个线程，就是把这个线程暂停了，它占着资源，但不运行，用 Resume 是恢复挂起的线程，

让这个线程继续执行下去。

76.有关线程的哪些叙述是对的（ ）

A 一旦一个线程被创建，它就立即开始运行。

B 使用 start()方法可以使一个线程成为可运行的，但是它不一定立即开始运行。

C 当一个线程因为抢先机制而停止运行，它被放在可运行队列的前面。

D 一个线程可能因为不同的原因停止并进入就绪状态。

解答： BCD

在抢占式线程模型中，操作系统可以在任何时候打断线程。通常会在它运行了一段时间（就是所谓的一个

时间片）后才打断它。这样的结果自然是没有线程能够不公平地长时间霸占处理器。

77.已知如下代码：（ ）

```
public class Test
{
public static void main(String arg[] )
{
int i = 5;
do{
System.out.print(i);
}while(--i>5)
System.out.print(“finished”);
}
```

}

执行后的输出是什么？

A 5

B 4

C 6

D finished

解答：AD

输出 5finished，do…while 循环中循环体一定会执行一次

78.下面的哪些声明是合法的？（ ）

A.long 1 = 4990

B.int i = 4L

C.float f =1.1

D.double d = 34.4

解答：AD

B.4L 应该是 long 类型的写法，

C.1.1 是 double 类型 ,float f=1.1f 是正确写法

79.给出如下代码：( )

```
class Test{
private int m;
public static void fun() {
//some code…
}
}
```

如何使成员变量 m 被函数 fun()直接访问？（）

A.将 private int m 改为 protected int m

B.将 private int m 改为 public int m

C.将 private int m 改为 static int m

D.将 private int m 改为 int m

解答：C

静态的方法中可以直接调用静态数据成员

80.以下哪个方法用于定义线程的执行体？（）

A.start()

B.init()

C.run()

D.main()

E.synchronized()

解答：run 方法是线程的执行体

81.给出下面的代码段：（ ）

```
public class Base{
int w, x, y, z;
public Base(int a, int b)
{x=a; y=b;
}
public Base(int a, int b, int c, int d)
{
//assignment x=a, y=b
w=d;z=c;
}}
```

在代码说明//assignment x=a, y=b 处写下如下哪几个代码是正确的？（）

A.Base(a, b);

B.x=a, y=b;

C.x=a; y=b;

D.this(a，b);

解答：CD

C 是直接给 x，y 赋值

D 是使用 this 调用本类中其它的构造方法

82.关于运算符>>和>>>描述正确的是

A.>>执行移动

B.>>执行翻转

C.>>执行有符号左移，>>>执行无符号左移

D.>>执行无符号左移，>>>执行有符号左移

解答：C

83. 选择 Java 语言中的基本数据类型（多选）

A.byte

B.Integer

C.String

D.char

E.long

答案：ADE

基本数据类型总共有 8 个：byte,short,int,long,char,boolean,float,double

84.从下列选项中选择正确的 Java 表达式

A.int k=new String(“aa”)

B.String str=String(“bb”)

C.char c=74;

D.long j=8888;

解答：BCD

85. Java I/O 程序设计中，下列描述正确的是

A. OutputStream 用于写操作

B. InputStream 用于写操作

C. I/O 库不支持对文件可读可写 API

解答：A

B.InputStream 用于读操作

C. I/O 支持对文件的读写

86. 下述代码的执行结果是

```
class Super {
     public int getLength() {return 4;}
}
public class Sub extends Super {
     public long getLength() {return 5;}
 public static void main (String[]args) {
```

```
    Super sooper = new Super ();
    Super  sub = new Sub();
    System.out.printIn(sooper.getLength()+ “,” + sub.getLength() };
}
}
```

A. 4, 4

B. 4, 5

C. 5, 4

D. 5, 5

E. 代码不能被编译

解答：E

方法重写返回值类型与父类的一致

87、Which two demonstrate a "has a" relationship(Choose two)?

A.
```
public interface Person { }
public class Employee extends Person{ }
```

B.
```
public interface Shape { }
public interface Rectandle extends Shape { }
```

C.
```
public interface Colorable { }
public class Shape implements Colorable
{ }
```

D.
```
public class Species{ }
public class Animal{private Species species;}
```

E.
```
interface Component{ }
class Container implements Component{
private Component[] children;
}
```

解答：D

“has a”是关联关系，关联分双向关联和单向关联,双向关联是 A,B 类分别持有对方的引用(有是对方的属性).

单向关联是一方持另一方的引用.

88. Given the folowing classes which of the following will compile without error?

```
interface IFace{}
class CFace implements IFace{}
class Base{}
public class ObRef extends Base{
                public static void main(String argv[]){
                ObRef ob = new ObRef();
                Base b = new Base();
                Object o1 = new Object();
                IFace o2 = new CFace();
              }
}
```

A. o1=o2;

B. b=ob;

C. ob=b;

D. o1=b;

解答：B

b 和 ob 对应的类之间没有任何关系，要想 b=ob 成立要么是父子关系，要么是接口实现类的关系

89. 关于 Java 语言，下列描述正确的是（多选）

A. switch 不能够作用在 String 类型上

B. List, Set, Map 都继承自 Collection 接口

C. Java 语言支持 goto 语句

D. GC 是垃圾收集器，程序员不用担心内存管理

解答：AD

B. Map 没有继承 Collection 接口

C. java 不支持 goto 语句

90. 指出下列程序运行的结果

```
public class Example{
    String str=new String("good");
    char[]ch={'a','b','c'};
```

```
    public static void main(String args[]){
        Example ex=new Example();
        ex.change(ex.str,ex.ch);
        System.out.print(ex.str+" and ");
        System.out.print(ex.ch);
    }
    public void change(String str,char ch[]){
        str="test ok";
        ch[0]='g';
    }
}
```

A good and abc

B good and gbc

C test ok and abc

D test ok and gbc

解答：B

数组和字符串都是引用类型。

91. 下列描述中，哪些符合 Java 语言的特征

A. 支持跨平台(Windows,Linux,Unix 等)

B. GC(自动垃圾回收)，提高了代码安全性

C. 支持类 C 的指针运算操作

D. 不支持与其它语言书写的程序进行通讯

解答：AB

92、关于异常(Exception)，下列描述正确的是

A. 异常的基类为 Exception，所有异常都必须直接或者间接继承它

B. 异常可以用 try{ . . .}catch(Exception e){ . . .}来捕获并进行处理

C. 如果某异常继承 RuntimeException，则该异常可以不被声明

D. 异常可以随便处理，而不是抛给外层的程序进行处理

解答：ABC

93. 下面的代码实现了设计模式中的什么模式

```
public class A {
private A instance;
private A() {
}
public static A getInstance {

if ( A == null )
 instance = new A();
return instance;
}
 }
```

A. Factory

B. Abstract Factory

C. Singleton

D. Builder

解答：Singleton 单例模式：该设计模式确保某一个类只有一个实例，而且自行实例化并向整个系统提

供这个实例

94. MAX_LENGTH 是 int 型 public 成员变量，变量值保持为常量 100，用简短语句定义这个变量。

A .public int MAX_LENGTH=100;

B. final int MAX_LENGTH=100;

C. final public int MAX_LENGTH=100;

D. public final int MAX_LENGTH=100.

解答：D 通过题的描述就是定义常量，在 java 中常量命名规范是所有字母都大写用下划线分割每个单词

95. String s=new String(“hello”);

String t =new String(“hello”);

char c [ ] ={ ‘h’,’ e’,’ l’,’ l’,’ o’ };

下列哪些表达式返回 true ？

A. s.equals(t);

B. t.equals(c);

C. s= =t ;

D. t.equals (new String(“hello”));

E. t= = c;

解答：AD String 类的 equals 方法已经覆盖了 Object 类的 equals 方法，比较的是两个字符串的内容是否

相等，双等号比较的是两个对象的内存地址是否相等

96. 类 Teacher 和 Student 是类 Person 的子类;

  Teacher t;

Student s;

// t and s are all non-null.

if (t instanceof Person){ s=(Student)t; }

最后一条语句的结果是:

A. 将构造一个 Student 对象;

B. 表达式是合法的;

C. 表达式是错误的;

D. 编译时正确, 但运行时错误。

解答：D

instanceof 是 Java 的一个二元操作符，它的作用是测试它左边的对象是否是它右边的类的实例，返回 boolean 类型的数据。

Teahcer 和 Student 之间没有继承关系不能做强制类型转换。

97. 关于线程设计，下列描述正确的是

A. 线程对象必须实现 Runnable 接口

B. 启动一个线程直接调用线程对象的 run()方法

C. Java 提供对多线程同步提供语言级的支持

D. 一个线程可以包含多个进程

解答：C

98. 欲构造 ArrayList 类得一个实例，此类继承了 List 接口，下列哪个方法是正确的：

A ArrayList myList = new Object();

B List myList = new ArrayList();

C ArraylList myList = new List();

D List myList = new List();

解答：B

99. 关于线程设计，下列描述正确的是

A. 线程对象必须实现 Runnable 接口

B. 启动一个线程直接调用线程对象的 run()方法

C. Java 提供对多线程同步提供语言级的支持

D. 一个线程可以包含多个进程

解答：C

100.以下各 DOS 命令能够显示出本机 DNS 服务器地址的是：( )

A. ping -a

B. ipconfig -all

C. netstat

D. telnet

解答：DOS 命令的使用

ping 命令：利用它可以检查网络是否能够连通，用好它可以很好地帮助我们分析判定网络故障

ifconfig all ：显示或设置网络设备

netstat: 用于查看当前基于 NETBIOS 的 TCP/IP 连接状态,通过该工具你可以 获得远程或本地

的组名和机器名。

telnet: 使用 telnet 命令访问远程计算机

101.下列 IP 地址中，属于 C 类 IP 地址的是( )

A. 201.222.1.65

B. 10.2.1.1

C. 127.63.2.99

D.255.255.255.255

解答：A

A 类地址（1.0.0.0-126.255.255.255）用于最大型的网络，该网络的节点数可达 16,777,216 个。

B 类地址（128.0.0.0-191.255.255.255）用于中型网络，节点数可达 65,536 个。

C 类地址（192.0.0.0-223.255.255.255）用于 256 个节点以下的小型网络的单点网络通信。

D 类地址（224.0.0.0-239.255.255.255。）并不反映网络的大小，只是用于组播，用来指定所分配的接收组播的节点组，这个节点组由组播订阅成员组成。

E 类（240.0.0.0-255.255.255.254）地址用于试验。

102.在使用匿名登录 ftp 时，用户名为( )？ (选择 1 项)

A、login users

B、anonymous

C、root

D、guest

解答：B

103.管理计算机通信的规则称为

A.协议

B.介质

C.服务

D.网络操作系统

解答：A

104.TCP 通信建立在连接的基础上，TCP 连接的建立要使用几次握手的过程。

A.2

B.3

C.4

D.5

解答:B

105.路由器工作在 ISO/OSI 参考模型的

A. 数据链路层

B.网络层

C. 传输层

解答：B

网络层属于 OSI 中的较高层次了，从它的名字可以看出，它解决的是网络与网络之间，即网际的通信问题，而不是同一网段内部的事。网络层的主要功能即是提供路由，即选择到达目标主机的最佳路径，并沿该路径传送数据包。除此之外，网络层还要能够消除网络拥挤，具有流量控制和拥挤控制的能力。网络边界中的路由器就工作在这个层次上，现在较高档的交换机也可直接工作在这个层次上，因此它

们也提供了路由功能，俗称“第三层交换机”.

106.OSI 体系结构定义了一个几层模型。

A.6

B.7

C.8

解答:B

OSI-RM ISO／OSI Reference Model

该模型是国际标准化组织（ISO）为网络通信制定的协议，根据网络通信的功能要求，它把通信过程分为七层，分别为物理层、数据链路层、网络层、传输层、会话层、表示层和应用层，每层都规定了完成的功能及相应的协议。

107.以下哪个命令用于测试网络连通。

A.telnet

B. netstat

C. ping

D. ftp

解答：C

108.在一个办公室内，将 6 台计算机用交换机连接成网络，该网络的屋里拓扑结构为

A 星型

B 总线型

C 树型

D 环型

解答：C

选项 A：星型拓扑结构 是一种以中央节点为中心，把若干外围节点连接起来的辐射式互联结构。这种结构适用于局域网，特别是近年来连接的局域网大都采用这种连接方式。这种连接方式以双绞线或同轴电缆作连接线路。

优点：结构简单、容易实现、便于管理，通常以集线器（Hub）作为中央节点，便于维护和管理。缺点：中心结点是全网络的可靠瓶颈，中心结点出现故障会导致网络的瘫痪。

选项 B：总线拓扑结构 是将网络中的所有设备通过相应的硬件接口直接连接到公共总线上，结点之间按广播方式通信，一个结点发出的信息，总线上的其它结点均可“收听”到。

优点：结构简单、布线容易、可靠性较高，易于扩充，节点的故障不会殃及系统，是局域网常采用的

拓扑结构。

缺点：所有的数据都需经过总线传送，总线成为整个网络的瓶颈；出现故障诊断较为困难。另外，由于信道共享，连接的节点不宜过多，总线自身的故障可以导致系统的崩溃。最著名的总线拓扑结构是以太网（Ethernet）。

选项 C ：树型拓扑结构 是一种层次结构，结点按层次连结，信息交换主要在上下结点之间进行，相邻结点或同层结点之间一般不进行数据交换。

优点：连结简单，维护方便，适用于汇集信息的应用要求。

缺点：资源共享能力较低，可靠性不高，任何一个工作站或链路的故障都会影响整个网络的运行。

选项 D： 环形拓扑结构 各结点通过通信线路组成闭合回路，环中数据只能单向传输，信息在每台设备上的延时时间是固定的。特别适合实时控制的局域网系统。

优点：结构简单，适合使用光纤，传输距离远，传输延迟确定。

缺点：环网中的每个结点均成为网络可靠性的瓶颈，任意结点出现故障都会造成网络瘫痪，另外故障诊断也较困难。最著名的环形拓扑结构网络是令牌环网（Token Ring）

109.下列既可用作输入设备又可用作输出设备的是。

A.鼠标器

B.磁盘

C.键盘

D.显示器

解答：B

110.计算机的电源切断之后，存储内容全部消失的存储器是。

A.软磁盘

B.只读存储器

C.硬盘

D.随机存储器

解答:D

111.中央处理单元（CPU）的两个主要组成部分是运算器和什么。

A.寄存器

B.主存储器

C.控制器

D.辅助存储器

解答：C

控制器:由程序计数器、指令寄存器、指令译码器、时序产生器和操作控制器组成，它是发布命令的“决策机构”，即完成协调和指挥整个计算机系统的操作。

运算器：arithmetic unit，计算机中执行各种算术和逻辑运算操作的部件。运算器由：算术逻辑单元（ALU）、累加器、状态寄存器、通用寄存器组等组成。主要功能：执行所有的算术运算；执行所有的逻辑运算，并进行逻辑测试，如零值测试或两个值的比较。

112.网上“黑客”是指什么样的用户。

A.匿名上网;

B.总在晚上上网;

C.在网上私闯他人计算机系统;

D.不花钱上网;

解答：C

113.防火墙系统的主要作用是。

A. 防止系统崩溃

B. 防病毒

C. 保护内部网络安全

D. 预防火灾

解答：C

114.下列说法中哪项是错误的是。

A. 不同 CPU 的计算机有不同的机器语言和汇编语言;

B. 回收站是硬盘中的一块区域，而剪贴板是内存中的一块区域;

C. 在 Dos 中可以用的文件名在 windows 98 中一定可用;

D. 计算机病毒不可能会驻留在 BIOS 中。

解答：D

BIOS 是英文"Basic Input Output System"的缩略语，直译过来后中文名称就是"基本输入输出系统"。其实，它是一组固化到计算机内主板上一个 ROM 芯片上的程序，它保存着计算机最重要的基本输入输出的程序、系统设置信息、开机后自检程序和系统自启动程序。 其主要功能是为计算机提供最底层的、最直接的硬件设置和控制。

115.用于电子邮件的协议是。

A. IP

B. TCP

C. SNMP

D. SMTP

解答：D

116.Java 网络程序设计中,下列正确的描述是

A. Java 网络编程 API 建立在 Socket 基础之上

B. Java 网络接口只支持 TCP 以及其上层协议

C. Java 网络接口只支持 UDP 以及其上层协议

D. Java 网络接口支持 IP 以上的所有高层协议

解答:AD

117.序列图描述对象是如何交互的并且将重点放在消息序列上。

A、正确　　B、不正确

解答：B

序列图主要用于按照交互发生的一系列顺序，显示对象之间的这些交互

118.一个 XML 必须有 DTD 或 Schemas。

A、正确　　B、不正确

解答：B

DTD 和 Schemas 都可以定义 XML 文档中出现的元素,这些元素出现的次序,它们如何相互 嵌套以及 XML 文档结构的其它详细信息。但不是必须的

119.下列哪些是 J2EE 的体系。

A、JSP

B、JAVA

C、Servlet

D、WebService

解答：ACD

J2EE 现在更多使用的名字是 Java EE　JSP 是 JavaEE 设计模式 MVC 中的显示部分，Servlet 是控制部分，WebService 是 JavaEE 的服务器。

120.在 Struts 中实现页面跳转主要通过什么方法来实现？

A、server.transfer

B、response.redirect

C、mapping.findForward

D、response.sendRedirect

解答：C

121.EJB 的优点有哪些？(选择 2 项)

A、技术领先　B、价格低廉　C、性能优越　D、强大的容器支持

解答：CD

122.以下哪些接口能够实现对 Web 访问者的身份认证？(选择 1 项)

A、Http Servlet Request

B、Http Servlet Response

C、Http Session

D、Http Servlet

解答：C

123.无状态会话 Bean、有状态会话 Bean、CMP 与 BMP 中，哪一种 Bean 不需要自己书写连接数据库的代码？

A、无状态会话 Bean

B、有状态会话 Bean

C、CMP

D、BMP

解答：C

BMP 是在 Bean 中完成对数据库 JDBC 的各种调用

CMP 是由 EJB 容器自动完成对数据库的操作

会话 Bean 主要处理业务逻辑

124.假设 web 应用的文档根目录为 MyApp，那么可以从哪里找到 database.jar 文件。

A.　MyApp 目录下

B.　MyApp\images 目录下

C.　MyApp\WEB-INF 目录下

D.　MyApp\WEB-INF\lib 目录下

解答：D

Web 工程的 lib 是放置.jar 文件的地方。

125.从以下哪一个选项中可以获得 Servlet 的初始化参数。

A.Servlet

B.ServletContext

C.ServletConfig

D.GenericServlet

解答：C

servlet 的生命周期的方法中有一个 init 方法，其中一个重载的 init 方法的参数为 ServletConfig 可以获取初始化参数。

126.哪一个对象可以用于获得浏览器发送的请求。

A.HttpServletRequest

B.HttpServletResponse

C.HttpServlet

D.Http

解答：A

HttpServletRequest 中有一些方法可以获取浏览器发送的请求信息。

127.运行 jsp 需要安装_____Web 服务器。

A. Apache

B. tomcat

C. WebLogic

D. IIS

解答：BC

Apache 是 PHP 程序运行的服务器，IIS 是.net 程序运行的服务器。

128.如何取得数据源。

A.通过 Http

B.通过 ftp

C.JNDI

D.通过 Connection 对象

解答：C

在服务器上配置好数据以后，通过 JNDI 技术获取到数据源。

JNDI：Java Naming and Directory Interface 一种标准的 Java 命名系统接口

129.下列哪一个接口定义了用于查找、创建和删除 EJB 实例

A.Home

B.Remote

C.Local

D.Message

解答：A

remote 接口定义了业务方法，用于 EJB 客户端调用业务方法。 home 接口是 EJB 工厂用于创建和移除查找 EJB 实例

130. 在服务器的网络编程中，解决会话跟踪的方法有：

A. 使用 Cookie。

B. 使用 URL 重写。

C. 使用隐藏的表单域。

D. 以上方法都不能单独使用。

解答：ABC

URL 重写就是首先获得一个进入的 URL 请求然后把它重新写成网站可以处理的另一个 URL 的过程

隐藏域是在页面级保存信息。与其他用户标准控件的区别是，隐藏域不被呈现在页面中。当页面提交的时候，隐藏域中的值将被一同发送给服务端。

Cookie 是以文本存储于计算机中，使用 name-value 匹配。一般用户存储标识用户信息

131.与 HttpSessionListener 接口有关的方法是。

A.sessionInitialized()

B.sessionCreated()

C.sessionFinialized()

D.sessionDestroyed()

解答：BD

132.关于 JSP 生命周期的叙述，下列哪些为真？

A.JSP 会先解释成 Servlet 源文件，然后编译成 Servlet 类文件

B.每当用户端运行 JSP 时，jspInit()方法都会运行一次

C.每当用户端运行 JSP 时，_jspService()方法都会运行一次

D.每当用户端运行 JSP 时，jspDestroy()方法都会运行一次

解答：AC

133.以下声明正确的是。

A. <xml-stylesheet type="txt/css" href="abc.css">

B. <?xml-stylesheet type='txt/css' href='abc.css'?>

C. <?xml-stylesheet type="txt/css" href="abc.css"?>

D. <%xml-stylesheet type="txt/css" href="abc.css"%>

解答 BC

单引号，双引号都可以使用在属性上。

134.下列哪个为 JSP 的隐含对象。

A.env

B.page

C.jspinfo

D.context

解答：B

JSP 有九个隐士对象

request 对象：保存了很多客户端请求的信息。

response 对象：生成服务器端响应，然后将响应结果发送到客户端

out 对象：表示输出流，此输出流将作为请求发送到客户端

session 对象：我们写个对象放在这个 session 对象中,这个对象就在我们的会话中都存在。

application 对象:我们写个对象放在这个 application 对象中,这个对象就在整个应用程序中都存在

pageContext 对象相当于当前页面的容器，可以访问当前页面的所有对象。

paget 对象:一般我们使用 Page 指令来替代使用这个对象

exception 对象：用来处理异常的

config 对象：一样的我们在页面中是使用很少的，一般会在 Servlet 中使用这个

135.下面的那一个不属于 MVC 模式中的对象？

A. Model

B. View

C. Collection

D. Controller

解答：C

MVC 是三个单词的缩写,分别为： 模型(Model),视图(View)和控制 Controller)。 MVC 模式的目的就是实现 Web 系统的职能分工。 Model 层实现系统中的业务逻辑，通常可以用 JavaBean 或 EJB 来实现。

View 层用于与用户的交互，通常用 JSP 来实现。 Controller 层是 Model 与 View 之间沟通的桥梁，它可以分派用户的请求并选择恰当的视图以用于显示，同时它也可以解释用户的输入并将它们映射为模型层可执行的操作。

136.要创建一个 EJB，必须要至少编写哪些 Java 类和接口？

A. 定义远程(或业务)接口

B. 定义本地接口

C. 定义 Bean 接口

D. 编写 Bean 的实现

解答：ABC

137.XML 是一种元语言，可以用它来描述其他语言。

A. 正确

B. 错误

解答：B

XML（Extensible Markup Language）即可扩展标记语言，它与 HTML 一样，都是 SGML(Standard Generalized Markup Language,标准通用标记语言)。Xml 是 Internet 环境中跨平台的，依赖于内容的技术，是当前处理结构化文档信息的有力工具。扩展标记语言 XML 是一种简单的数据存储语言，使用一系列简单的标记描述数据，而这些标记可以用方便的方式建立，虽然 XML 占用的空间比二进制数据要占用更多的空间，但 XML 极其简单易于掌握和使用。

138.类图用来表示系统中类和类与类之间的关系，它是对系统动态结构的描述。(选择 1 项)

A.正确

B.不正确

解答：B

类图是对系统静态结构的描述。

139.哪一个不是 EL 定义的隐式对象？(选择 1 项)

A cookie

B.pageContext

C.attributes

D initParam

解答：C

1)pageContext：JSP 页的上下文。它可以用于访问 JSP 隐式对象。

2)Param：将请求参数名称映射到单个字符串参数值（通过调用 ServletRequest.getParameter (String name) 获得)。getParameter (String) 方法返回带有特定名称的参数。表达式 $(param.name) 相当于 request.getParameter (name)。

3)paramValues：将请求参数名称映射到一个数值数组（通过调用 ServletRequest.getParameter (String name) 获得)。它与 param 隐式对象非常类似，但它检索一个字符串数组而不是单个值。表达式 ${paramvalues.name} 相当于 request.getParamterValues(name)。

4)header 将请求头名称映射到单个字符串头值（通过调用 ServletRequest.getHeader(String name) 获得)。表达式 ${header.name} 相当于 request.getHeader(name)。

5)headerValues 将请求头名称映射到一个数值数组(通过调用 ServletRequest.getHeaders(String) 获得)。它与头隐式对象非常类似。表达式 ${headerValues.name} 相当于 request.getHeaderValues(name)。

6)cookie 将 cookie 名称映射到单个 cookie 对象。向服务器发出的客户端请求可以获得一个或多个 cookie。表达式 ${cookie.name.value} 返回带有特定名称的第一个 cookie 值。如果请求包含多个同名的 cookie，则应该使用 ${headerValues.name} 表达式。

7)initParam 将上下文初始化参数名称映射到单个值（通过调用 ServletContext.getInitparameter(String name) 获得)。

8)pageScope 将页面范围的变量名称映射到其值。例如,EL 表达式可以使用 ${pageScope.objectName} 访问一个 JSP 中页面范围的对象，还可以使用 ${pageScope.objectName.attributeName} 访问对象的属性。

9)requestScope 将请求范围的变量名称映射到其值。该对象允许访问请求对象的属性。例如，EL 表达式可以使用 ${requestScope.objectName} 访问一个 JSP 请求范围的对象，还可以使用 ${requestScope.objectName.attributeName} 访问对象的属性。

10)sessionScope 将会话范围的变量名称映射到其值。该对象允许访问会话对象的属性。例

11)applicationScope 将应用程序范围的变量名称映射到其值。该隐式对象允许访问应用程序范围的对象。

140.下面哪些属于 JSTL 中的表达式操作标签。(选择 1 项)

A. <c:out>

B. <c:if>

C. <c:url>

D. <c:catch>

解答：A

141.Struts 框架可以支持以下哪种程序开发语言？(选择 1 项)

A.C

B.C++

C.Java

D.C#

解答：C

142.在 Servlet 处理请求的方式为。(选择 1 项)

A、以进程的方式

B、以程序的方式

C、以线程的方式

D、以响应的方式

解答：C

Servlet 采用多线程来处理多个请求同时访问，Servlet 容器维护了一个线程池来服务请求。

143.javax.Servlet 的包中，属于类的是。(选择 1 项)

A、Servlet

B、GenericServlet

C、ServletRequest

D、ServletContext

解答：B

ServletContext 和 ServletRequest 是该包下的接口。

144.以下说法正确的是。(选择 2 项)

A. #CDATA 标记表示该元素包含的数据将被解析器解释

B. #PCDATA 标记表示该元素包含的数据将不被解析器解释

C. #PCDATA 标记表示该元素包含的数据将被解析器解释

D. #CDATA 标记表示该元素包含的数据将不被解析器解释

解答：AC

在 DTD 中，指定某个标签中的内容是字符数据时，使用(#PCDATA)。而指定标签中某个属性的类型为字符型时，使用 CDATA，都将被解析器解释。

145.Http 缺省的请求方法是。(选择 1 项)

A.PUT

B.GET

C.POST

D.TRACE

解答：B

146.在 XML 中用于注释的符号是。(选择 1 项)

A. <!-- -->

B. <?-- --?>

C. <% %>

D. <!-- --!>

解答：A

147.DTD 与 XML Schema 都是 XML 文档。(选择 1 项)

A. 正确

B. 不正确

解答：DTD 不是 XML 文件， schema 是 XML 文档

148.JDBC 中，用于表示数据库连接的对象是。(选择 1 项)

A. Statement

B. Connection

C. DriverManager

D. PreparedStatement

解答：B

Statement 和 PreparedStatement 都是用来发送和执行 SQL 语句的

DriverManager 管理一组驱动程序

149.用于调用存储过程的对象是。(选择 1 项)

A.ResultSet

B.DriverManager

C.CallableStatemet

D.PreparedStatement

解答：C

ResultSet 是结果集对象

DriverManager 管理一组驱动程序

PreparedStatement 预编译的，用来发送和执行 SQL 语句的

150.如果没有指定 Cookie 的时效，那么默认的时效是。(选择 1 项)

A. 一天

B. 永不过期

C. 会话级别

D.一分钟

解答：C

这是 API 的原文：By default, -1 indicating the cookie will persist until browser shutdown.

151.<?xml version="1.0" encoding="GB2312"?>

<!ELEMENT Customer EMPTY>

<!ATTLIST Customer 称呼 CDATA #IMPLIED 姓名 CDATA #REQUIRED 职位 CDATA #REQUIRED>关于上述 DTD 定义的描述正确的是。(选择 1 项)

A. Customer 元素能包含子元素，并且能为空

B. Customer 元素能包含文本，并且能为空

C. Customer 元素不能包含文本，也不能包含子元素

D. Customer 元素的所有实例的“称呼”属性必须有值，不能为空

解答：C

EMPTY 表示元素不能包含文本，也不能包含子元素

#IMPLIED 属性可以没有值

#REQUIRED 属性必须有值

152.实现现下列哪一种接口的对象，并不需要在 web.xml 文件内进行额外的设定，Servlet 容器就能够回应该对象加入 HTTP 会话所发生的事件？(选择 1 项)

A. ServletContextListener

B. HttpSessionListener

C. HttpSessionAttributeListener

D. HttpSessionBindingListener

解答：D

HttpSessionListener 只需要设置到 web.xml 中就可以监听整个应用中的所有 session。

HttpSessionBindingListener 必须实例化后放入某一个 session 中，才可以进行监听

153.下列哪个为 JSP 的小脚本的标签？(选择 1 项)

A. <% %>

B. <@ %>

C. <%! %>

D. <%-- %>

解答：A

154.以下不属于 JSP 的标准指令的是。(选择 1 项)

A.Taglib

B.Include

C.Import

D.Page

解答：C

import 是 page 指令的一个属性。

155.对于每一个网站访问用户都要访问的变量，应该将它设为_____变量。(选择 1 项)

A. Session

B. Reques

C. Response

D. Application

解答：D

Application 应用程序级变量

156.EJB 类库存在于 Java 的哪个版本中？(选择 1 项)。

A. J2SE

B. J2EE

C. J2ME

D. J2NE

解答：B

Java 分为三个体系

JavaSE(Java2 Platform Standard Edition，java 平台标准版)，

JavaEE(Java 2 Platform,Enterprise Edition，java 平台企业版)，

JavaME(Java 2 Platform Micro Edition，java 平台微型版)。

EJB 属于 JavaEE 版本

157. 察看下列 JSP 内容

```
<html><body>
<%  for (int    i=0;i<3;i++){   %>
    out.print(i*2);
<%  }   %>
</body></html>
```

当这个 JSP 被运行时，其结果是什么？(选择 1 项)

A.此 JSP 因为语法错误，无法运行

B.显示出 0，2，4

C.显示出 0，2，4，6

D.显示出 out.print(i*2) out.print(i*2) out.print(i*2)

解答：D

158 假设 A.jsp 内设定一个<jsp:useBean>元素：

```
<jsp:useBean id=”bean1”  class=”myBean”  />
```

下列哪一个为真？(选择 1 项)

A.bean1 的存取范围（scope）默认为 application

B.在 HTTP 会话内可以存取 bean1

C.只有在 A.jsp 内可以存取 bean1

D.在 A.jsp 所属的 Web 应用程序内均可存取 bean1

解答：C

bean1 的存取范围（scope）默认为 page

(题有一点问题 javabean 的规则是要放在一个包中)

159 在 MVC 设计模式中，JavaBean 的作用是。(选择 1 项)

A、Controller

B、Model

C、业务数据的封装

D、View

解答：B

同 137 题

160 在 J2EE 中属于 Web 层的组件有(选择 1 项)

A. HTML

B. EJB

C. Applet

D. JSP

解答：D

161 单元测试是在软件开发过程中的哪个阶段完成的？ (选择 1 项)

A. 可行性研究和计划

B. 概要设计

C. 实现

D. 使用和维护

解答：C

162 在 J2EE 的 Web 应用中，编译后的 class 文件存放的目录为(选择 1 项)

A. classes 目录

B. images 目录

C. jar 目录

D. 任意位置

解答：A

163 HttpServlet 中，用来处理 POST 请求的方法是(选择 1 项)

A. doHead

B. doGet

C. doPost

D. doPut

解答：C

164.DOM 中 XMLDOMnodelist 的 length 属性的表示是：(选择 1 项)

A. 该对象中文本字符的长度

B. 该对象中元素节点的数量

C. 该对象中节点的数量

D. 该对象中文档对象的数量

解答：A

length 属性返回注释节点中的文本长度，以字符数计。

165.如何创建 Cookie? (选择 1 项)

A. 使用 new Cookie 语句

B. 调用 response.addCookie 方法

C. 使用 Cookie 的 setMaxAge 方法

D. setCookie 方法

解答：B

166. 关于 Web 应用程序，下列说法错误的是（ ）。

A. WEB-INF 目录存在于 web 应用的根目录下

B. WEB-INF 目录与 classes 目录平行

C. web.xml 在 WEB-INF 目录下

D. Web 应用程序可以打包为 war 文件

解答：B

classes 目录在 WEB-INF 目录下。

167 有关 Servlet 的生命周期说法正确的有（ ）。

A. Servlet 的生命周期由 Servlet 实例控制

B. init()方法在创建完 Servlet 实例后对其进行初始化，传递的参数为实现 ServletContext 接口的对象

C. service()方法响应客户端发出的请求

D. destroy()方法释放 Servlet 实例

解答：C

Servlet 生命周期就是指创建 Servlet 实例后响应客户请求直至销毁的全过程.

Serlvet 生命周期的三个方法:init()-->service()-->destroy(),

Servlet 生命周期的各个阶段: 实例化:Servlet 容器创建 Servlet 类的实例对象

初始化:Servlet 容器调用 Servlet 的 init()方法

服务:如果请求 Servlet,则容器调用 service()方法

销毁:销毁实例之前调用 destroy()方法

168. 以下 web.xml 片断（ ）正确地声明 servlet 上下文参数。

A

<init-param>

<param-name>MAX</param-name>

<param-value>100</param-value>

</init-param>

B

<context-param>

<param name="MAX" value="100" />

<context-param>

C

<context>

<param name="MAX" value="100" />

<context>

D

<context-param>

<param-name>MAX</param-name>

<param-value>100</param-value>

<context-param>

解答：A

169.以下（ ）可用于检索 session 属性 userid 的值。

A. session. getAttribute (“userid”);

B. session. setAttribute (“userid”);

C. request. getParameter (“userid”);

D. request. getAttribute (“userid”);

解答：A

170. 下列 JSP 代码：

<html>

<body>

<%

 for(int i = 0; i < 10; i++) {

//1

}

%>

</body>

</html>

以下（ ）可放置在//1 处，不会发生编译错误。

A <%= i %>

B i

C %><%= i %><%

D 不写任何内容

解答：CD

171.考虑下面两个 JSP 文件代码片断：

test1.jsp:

<HTML>

<BODY>

<% pageContext.setAttribute(”ten”,new Integer(10));%>

//1

</BODY>

</HTML>

test2.jsp:

数字为：<%= pageContext.getAttribute(”ten”)%>

以下（ ）放置在 test1.jsp 中的//1 处，当请求 test1.jsp 时正确输出 test2.jsp 中的内容。

A. <jsp:include page=”test2.jsp” />

B. <jsp:forword page=”test2.jsp” />

C. <%@ include file=”test2.jsp” %>

D. 由于 pageContext 对象的 scope 属性为 page,所以 test2.jsp 不能访问 test1.jsp 定义的属性

解答：C

pageContext setAttribute(”ten”,new Integer(10));能取到的范围是 page，也就是当前页面。

<%@includefile=”” %>：编译时包含,静态的，JSP 引擎将对所包含的文件进行语法分析.

<jsp:include page=”” />：运行时包含，静态和动态的都可以，JSP 引擎不对所包含的文件进行语

法分析，只获得处理的结果

172.有关 JSP 隐式对象，以下（ ）描述正确。

A. 隐式对象是 WEB 容器加载的一组类的实例，可以直接在 JSP 页面使用

B. 不能通过 config 对象获取 ServletContext 对象

C. response 对象通过 sendRedirect 方法实现重定向

D. 只有在出错处理页面才有 exception 对象

解答：ACD

可以通过 config 对象获取 ServletContext 对象

173. 考虑下面 JSP 文件代码片断：

```
<HTML>
<BODY>
<jsp:include page=”test2.jsp”>
<jsp:param name=”username” value=”accp”/>
</jsp:include>
</BODY>
</HTML>
```

以下（ ）代码片断放置在 test2.jsp 中不会导致错误。

A. <jsp:getParam name=”username”/>

B. <jsp:include param =”username”/>

C. <%=request.getParameter(“username”)%>

D. <%=request.getAttribute(“username”)%>

解答：C

174.以下是 login.jsp 文件的代码片断：

```
<%@ page  isELIgnored="false"%>
<html>
<body>
<FORM action="login.jsp" method="GET">
  <input type="text" name="name" value="${param['name']}">
  <input type="submit" value="提交">
</FORM>
```

<P>

用户名为: ${param.name}

</body>

</html>

以下（ ）描述正确。

A. 发生运行错误

B. 页面会出现一文本框，并且文本框中内容为${param['name']}

C. 当用户输入名字并单击“提交”按钮时，在同一页面中的“用户名为:”字样后面会显示用户输入的内容

D.当用户输入名字并单击“提交”按钮时,在同一页面中的“用户名为:”字样后面会显示${param.name}

解答：C

在每个 JSP 中也可以指定是否该 JSP 使用 EL。在 page directive 中的 isELIgnored 属性用来指定是否忽略。格式为：

＜%@ page isELIgnored＝"true|false"%＞

如果设定为真，那么 JSP 中的表达式被当成字符串处理。

175. doAfterBody()方法是在（ ）接口中定义的。

A. Tag

B. IterationTag

C. BodyTag

D. TagSupport

解答：B

176.下面代码片断在浏览器中输出结果为（ ）。

<c:set var=”myVar” scope =”session”>

2+3

</c:set>

<c:out value=”${myVar}” />

A.0

B.5

C.2+3

D.无输出

解答：C

<c:set>标签主要用来将变量保存到 JSP 的会话中或 JavaBean 的属性中。<c:set>的语法格式为：

<c:set value="value" var="varName"

[scope="{ page|request|session|application }"]/>

上述的表达式实现的功能是将 value 的值储存至范围为 scope 的 varName 变量之中，还可以将 value 的值储存至 target 对象的属性中.如果题中 2+3 改成${2+3}那么结果就是 5.

177.编写一个 Filter，除继承 HttpServlet 类外还需要（ ）。

A. 继承 Filter 类

B. 实现 Filter 接口

C. 继承 HttpFilter 类

D. 实现 HttpFilter 接口

解答：B

178.以下哪个 Hibernate 主键生成策略是实现主键按数值顺序递增的？

A、increment

B、identity

C、sequence

D、native

解答：A

increment 生成策略：当 Hibernate 准备在数据库表中插入一条新记录时，首先从数据库表中获取当前主键字段的最大值，然后在最大值基础上加 1，作为当前持久化对象的标识符属性值。这种策略即 increment 生成策略，用其生成的标识符属性的类型可以是 long、short、int 及其封装类的类型

identity 生成策略：在 MS SQL Server、MySQL 和 DB2 等数据库中可以设置表中某一个字段的数值自动增长，identity 生成策略通过这种方式为当前记录获取主键值的同时为持久化对象赋予标识符属性值。

sequence 生成策略：在 Oracle、DB2 和 PostgreSQL 等数据库中创建一个序列（sequence），然后 Hibernate 通过该序列为当前记录获取主键值，进而为持久化对象赋予标识符属性值。

native 生成策略：由 Hibernate 根据所使用的数据库支持能力从 identity、sequence 或者等生成策略中选择一种

179.在 jsp 中，page 指令的（）属性用来引入需要的包或类。

A、extends

B、import

C、language

D、contentType

解答：B

page 指令属性简要介绍：

language="java"

声明脚本语言的种类，暂时只能用"java"

extends="package.class"

标明 JSP 编译时需要加入的 Java Class 的全名，但是得慎重的使用它，它会限制 JSP 的编译能力.

import="{package.class | package.* }, ..."

需要导入的 Java 包的列表，这些包就作用于程序段，表达式，以及声明。

下面的包在 JSP 编译时已经导入了，所以你就不需要再指明了:

java.lang.*

javax.servlet.*

javax.servlet.jsp.*

javax.servlet.http.*

session="true | false"

设定客户是否需要 HTTP Session.如果它为 true,那么 Session 是有用的。

如果它有 false,那么你就不能使用 session 对象，以及定义了 scope=session 的<jsp:useBean>元素。这样的使用会导致错误. 缺省值是 true.

buffer="none | 8kb | sizekb"

buffer 的大小被 out 对象用于处理执行后的 JSP 对客户浏览器的输出。

缺省值是 8kb

autoFlush="true | false"

设置如果 buffer 溢出，是否需要强制输出，如果其值被定义为 true(缺省值)，输出正常，如果它被设置为 false,如果这个 buffer 溢出,就会导致一个意外错误的发生.如果你把 buffer 设置为 none,那么你就不能把 autoFlush 设置为 false.

isThreadSafe="true | false"

设置 Jsp 文件是否能多线程使用。缺省值是 true,也就是说，JSP 能够同时处理多个用户的请求，

如果设置为 false，一个 jsp 只能一次处理一个请求

info="text"

一个文本在执行 JSP 将会被逐字加入 JSP 中，你能够使用

Servlet.getServletInfo 方法取回。

errorPage="relativeURL"

设置处理异常事件的 JSP 文件。

isErrorPage="true | false"

设置此页是否为出错页，如果被设置为 true,你就能使用 exception 对象.

contentType="mimeType [;charset=characterSet]" | "text/html;charset=ISO-8859-1"

设置 MIME 类型 。缺省 MIME 类型是: text/html, 缺省字符集为 ISO-8859-1.

180.正则表达式”\d+\.?\d*”在匹配下列字符串时结果是失败的是?

A 12.5

B 1.25

C 以上都成功

D 以上都失败

解答：B

\d+ 表示可以出现 1 次或是 n 次数字

\. .? 表示可以“.”可以出现一次，也可以不出现

\d* 表示可以出现 0 次或是 n 次数字

181.下列没有直接采用 XML 技术的是（ ）

A. UDDI

B. SOAP

C. AJAX

D.DCOM

解答：D

DCOM（分布式组件对象模型,分布式组件对象模式）是一系列微软的概念和程序接口，利用这个接口，客户端程序对象能够请求来自网络中另一台计算机上的服务器程序对象。DCOM 基于组件对象模型（COM），COM 提供了一套允许同一台计算机上的客户端和服务器之间进行通信的接口（运行在 Windows95 或者其后的版本上）。

182.下列可以用来解析 XML 的是（ ）

A.CSS

B.DTD

C.SAX

D.XSL

解答：C

java 解析 xml 文件四种方式：SAX　DOM　JDOM　DOM4J

183.下面关于 XML 叙述不正确的是（　　）

A.XML 标记必须关闭

B.XML 是大小写敏感的

C.XML 文件只能跟 DTD 文件一块使用

D.XML 和 XSL 结合可以在浏览器上显示

解答：C

184.在不指定特殊属性的情况下，哪几种 HTML 标签可以手动输入文本：()

A. <TEXTAREA></TEXTAREA>

B. <INPUT　type=”text”/>

C. <INPUT　type=”hidden”/>

D. <DIV></DIV>

解答：AB

185.关于 IFrame 表述正确的有：()

A. 通过 IFrame，网页可以嵌入其他网页内容，并可以动态更改

B. 在相同域名下，内嵌的 IFrame 可以获取外层网页的对象

C. 在相同域名下，外层网页脚本可以获取 IFrame 网页内的对象

D. 可以通过脚本调整 IFrame 的大小

解答：CD

IFRAME 元素也就是文档中的文档，或者好像浮动的框架(FRAME)。

通过 iframe 对象所在页面的对象模型，你可以访问 iframe 对象的属性，但不能访问其内容。

186.关于表格表述正确的有：()

A. 表格中可以包含 TBODY 元素

B. 表格中可以包含 CAPTION 元素

C. 表格中可以包含多个 TBODY 元素

D. 表格中可以包含 COLGROUP 元素

E. 表格中可以包含 COL 元素

解答：ACDE

caption 标签用于定义一个表格标题。<caption>标签只能出现在 table 标签中，且必须紧随 table 标签之后。每个表格只能定义一个标题。

使用 <tbody> 标签，可以将表格分为一个单独的部分。<tbody> 标签可将表格中的一行或几行合成一组。

利用<colgroup>标签可以把表格按列划分为若干组，每组可包含一列或几列，然后可以对各组分别设置格式。

通常一个列组的各列格式是相同的，如果列与列有差异，可通过在组内加入<col>标签进行设置。<col>标签只能在<table>标签和<colgroup>标签中使用。

187.在 DHTML 中把整个文档的各个元素作为对象处理的技术是：（）

A.HTML

B.CSS

C.DOM

D.Script(脚本语言)

解答：C

DOM：文档对象模型

188.下面属于 javascript 对象的有：( )

A. Window

B. Document

C. Form

D. String

E. Navigator

解答：ACE

189.Servlet 程序的入口点是？（　　）

A.init（）

B、main（）

C、service（）

D、doGet（）

解答：C

190.不能在不同用户之间共享数据的方法是？（　）

A、通过 cookie

B、利用文件系统

C、利用数据库

D、通过 ServletContext 对象

解答：A

191.　模块内聚度越高，说明模块内各成分彼此结合的程度越

A 松散

B 紧密

C 无法判断

D 相同

解答：B

内聚度是指模块内部各成分之间的联结强度.内聚度越高,越容易理解、修改和维护.但内聚度本身是主观的、非形式化的概念,程序设计人员很难客观地评估一个模块的内聚度.为此,人们开发出许多度量准则用于量化模块的内聚度[1~3],为程序设计人员开发出高内聚度的模块提供指南.

192 (单选)软件需求分析阶段的输出主要是

A. 需求说明书

B. 开发计划

C. 可行性报告

D. 设计说明书

解答：A

193. (单选)　以下选项中不是项目经理的职责的是？

A. 需求分析

B. 计划

C. 计划跟踪

D. 质量管理

解答：D

质量管理是 SQA（软件质量保证）人员的职责

194. (多选)配置管理能起到以下哪些作用？

A. 版本管理

B. 变更管理

C. 需求管理

D. 测试管理

解答：AB

195.下面的哪个方法在 servlet 的 response 的输出流在 URL 中保存 Session ID。（　　）

A.The encodeURL method of the HttpServletRequest interface.

B. The encodeURL method of the HttpServletResponse interface.

C.The rewriteURL method of the HttpServletRequest interface.

D. The rewriteURL method of the HttpServletResponse interface.

解答：B

196.看下面这个类

```
public class IfAttributsChanged implements ServletContextAttributeListener{
    public void attributeAdded(ServletContextAttributeEvent scab){
        System.out.println(“加入一个属性”);
    }
    public void attributeRemoved(ServletContextAttributeEvent scab){
        System.out.println(“删除一个属性”);
    }
}
```

关于 IfAttributsChanged 类的叙述，下列哪一个为真？(选择 1 项)

A.此类可以成功编译

B. 此类无法成功编译，原因是缺少 attributeChanged()方法。

C.此类无法成功编译，原因是缺少 attributeReplaced()方法。

D.此类无法成功编译，原因是缺少 attributeUpdated()方法。

解答：C

197.Oracle 数据库表空间与用户的关系是 ( )?

A.一对一

B.一对多

C.多对一

D.多对多

解答： D

一个用户可以使用一个或多个表空间，一个表空间也可以供多个用户使用。

198.Oracle 数据库表空间与数据文件的关系描述正确的是( )

A.一个表空间只能对应一个数据文件

B.一个表空间可以对应多个数据文件

C.一个数据文件可以对应多个表空间

D.表空间与数据文件没任何对应关系

解答：B

表空间和数据文件发生关系，数据文件是物理的，一个表空间可以包含多个数据文件，而一个数据文件只能隶属一个表空间

199.判断这 PL/SQL 代码块：(选择 1 项)

```
BEGIN
FOR i IN 1..6 LOOP
IF i = 2 OR i = 3 THEN null;
ELSE
INSERT INTO example(one) VALUES (i);
END IF;
ROLLBACK;
END LOOP;
COMMIT;
END;
```

有多少行被插入到表 EXAMPLE ？

A、0

B、1

C、2

D、3

解答：A

在循环结束前执行了 ROLLBACK 语句，数据被回滚。

200.你判断下面语句，有什么作用？(单选)

GRANT update ON inventory TO joe WITH GRANT OPTION;

A、一个系统权限被授予用户 JOE

B、一个对象权限被授予用户 JOE

C、用户 JOE 被授予在这个对象上的所有权限

D、一个系统权限和一个对象权限被授予用户 JOE

解答：B

with admin option 只能在赋予 system privilege 的时使用

with grant option 只能在赋予 object privilege 的时使用

201.表 CLASSES 和 表 SCHEDULE 结构如下:

CLASSES：

ID NUMBER(9)

CLASS_NAME VARCHAR2(20)

TEACHER_ID NUMBER(9)

SCHEDULE：

CLASS_TIME DATE

CLASS_ID NUMBER(9)

你建一个视图显示每一课的课名、课时，并按教师 ID 排序，判断下面语句将返回何种结果？ (选择 1 项)

CREATE VIEW class_schedule AS

SELECT C.class_name, s.class_time FROM classes c, schedule s WHERE C.id = s.class_id;

A .语句创建视图 CLASS_SCHEDULE 且可产生预期结果.

B .语句创建视图 CLASS_SCHEDULE 但不能产生预期结果.

C .语法错误，因为视图不可基于连接查询.

D .语法错，因为语句未包含 ORDER BY 子句.

解答：B

上述试图显示每一课的课名、课时，但是没有按教师 ID 排序

202.Oracle 数据库中，在 SQL 语句中连接字符串的方法是哪个？(选择 1 项)

A、cat

B、concat

C、join

D、+

解答：B

203.表 TEACHER 包含如下字段：

列名	可为空否？	数据类型
TEACHER_ID	NOT NULL	NUMBER(9)
NAME		VARCHAR2(25)
SALARY		NUMBER(7,2)
SUBJECT_ID	NOT NULL	NUMBER(3)
SUBJECT_DESCRIPTION		VARCHAR2(2)

你需要将理科教师的工资上浮 8%，理科教师的 SUBJECT_ID 是 011，你需用哪一句实现？(选择 1 项)

A.UPDATE teacher SET salary = salary * 1.08 WHERE subject_description LIKE ‘SCIENCE’

B.UPDATE teacher SET salary = salary * .08 WHERE subject_description LIKE ’SCIENCE’ AND subject_id = 011

C.UPDATE teacher SET salary = salary * 1.08 WHERE subject_id = 011;

D.UPDATE teacher SET salary = salary + (salary *.08) WHERE subject_description LIKE ’SCIENCE’ OR subject_id = 011

解答：C

204.定义游标如下：

```
DECLARE
CURSOR query_cursor(v_salary) IS
SELECT last_name, salary, dept_no
FROM employee
WHERE salary>v_salary;
```

这条语句为什么会错误？（选择 1 项）

A. 在游标定义中不允许出现 where 子句

B. select 语句中缺少 into 子句

C. 参数未指定为变量数据类型

D. 定义 cursor 的语法写错了

解答：C

205.在 PL/SQL 块的哪部分可以对初始变量赋予新值? (选择 1 项)

A.结尾部分

B.开头部分

C.执行部分

D.声明部分

解答：C

206.哪句可以实现显示 id 和 description ，条件满足订单时间在 January 1, 1997 以前的，且单价小于 1.00 或者大于 5.00 的， 结果用订单时间降序排列。(选择 1 项)

A.SELECT id_number, description FROM inventory

WHERE price IN (1.00, 5.00) OR order_date < '01-JAN-97'

ORDER BY order_date DESC;

B.SELECT id_number, description FROM inventory

WHERE price BETWEEN 1.00 AND 5.00 OR order_date < '01-JAN-1997'

ORDER BY order_date;

C.SELECT id_number, description FROM inventory

WHERE price < 1.00 OR price > 5.00 AND order_date < '01 -Jan-97'

ORDER BY order_date ASC;

D.SELECT id_number, description FROM inventory

WHERE (price <1.00 OR price > 5.00) AND order_date < '01-JAN-1997'

ORDER BY order_date DESC;

解答：D

207.判断下面句子，将返回什么值？ (选择 1 项)

SELECT id_number, description, price FROM inventory

WHERE manufacturer_id IN (SELECT manufacturer_id FROM inventory WHERE price > 8.00 OR quantity > 1000);

A 返回单价大于 8.00 且数量大于 1000 的存货的 货号、种类、单价信息

B 返回单价大于 8.00 或者数量大于 1000 的存货的 货号、种类、单价信息.

C 返回单价大于 8.00 或者数量大于 1000 且有制造商号的存货的 货号、种类、单价信息.

D 返回单价大于 8.00 或者数量大于 1000 的制造商的所有存货的 货号、种类、单价信息.

解答：C

208.考虑下列声明，那些是不合法的：（选择 3 项）

A. DECLARE v_name, v_dept VARCHAR2(14);

B. DECLARE v_test NUMBER(5);

C. DECLARE V_MAXSALARY NUMBER(7, 2) = 5000;

D. DECLARE V_JOINDATE BOOLEAN := SYSDATE;

解答：ACD

A. v_name 没有数据类型

C.:=是赋值

D. V_JOINDATE 是 boolean 类型，sysdate 是 Date 类型

209.关于 PL/SQL 块的执行部分下列说法正确的是？(选择 1 项)

A. PL/SQL 表达式可以包含分组函数.

B. PL/SQL 表达式不可以包含 SQL 函数.

C. 在 SQL 语句中部分分组函数可用.

D. 以上都不对

解答 A

210.表（TEACHER）包含以下列：

ID NUMBER(7) PK

SALARY NUMBER(7,2)

SUBJECT_ID NUMBER(7)

判断以下两个 SQL 语句：

（1）SELECT ROUND(SUM(salary),-2) FROM teacher ;

（2）SELECT subject_id, ROUND(SUM(salary),-2) FROM teacher GROUP BY subject_id ;

有什么不同结果？(选择 1 项)

A. 语句 1 将返回每个老师一个结果

B. 语句 2 将返回多个结果

C. 结果相同，显示不同

D. 将有一个句子产生错误

解答：B

语句 1 会返回一个结果，就是所有老师的工资和，使用了四舍五入函数

语句 2 是按照 subject_id 进行分组，那么分几组就会有几个结果

211.游标的哪一种属性指示 fetch 语句是否从活动集中返回行，如未能返回行，则此属性的值为 true ？(选择 1 项)

A. %FOUND

B. %NOTFOUND

C. %ROWCOUNT

D. %ISOPEN

解答：B

%FOUND 布尔型属性，当最近一次读记录时成功返回,则值为 TRUE；

%NOTFOUND 布尔型属性，与%FOUND 相反；

%ISOPEN 布尔型属性，当游标已打开时返回 TRUE；

%ROWCOUNT 数字型属性，返回已从游标中读取的记录数

212.哪一子句可实现 SELECT 语句查询员工平均工资小于 5000 的部门信息 ？(选择 1 项)

A. GROUP BY dept_id WHERE AVG(sal) < 5000

B.GROUP BY AVG(sal) HAVING AVG(sal) < 5000

C. GROUP BY dept_id HAVING AVG(sal) < 5000

D. GROUP BY AVG(sal) < 5000

解答：C

使用 HAVING 过滤分组。

213.在 PL/SQL 中使用哪几种语句来对变量进行赋值？ (选择 3 项)

A ：=

B SELECT INTO

C FETCH INTO

D =

解答：ABC

214.你试图用下面句子查询数据：(选择 1 项)

SELECT 100/NVL(quantity, 0) FROM inventory;

为何 QUANTITY 为 null 空值时，将导致出错？

A. 表达式企图被空值除.

B.换函数参数数据类型不一致.

C.空值不能被转成实际值

D. 表达式企图被零除.

解答：D

nvl(表达式 1，表达式 2) 这个函数的作用是如果表达式 1 的值为 null，那么取表达式 2 的值。

215.PL/SQL 的哪一部分实现对数据的操作? (选择 1 项)

A、头部分

B、列外部分

C、执行部分

D、声明部分

解答：C

216 下列哪个集合操作符返回两个查询所选择的所有的行。(选择 1 项)

A Union

B Union all

C Union only

D connect by

解答：B

Union 集合操作符返回两个查询所选择的去除重复行。

217.在 Oracle 中，当需要使用显式游标更新或删除游标中的行时，UPDATE 或 DELETE 语句必须使用（）子句。

A. WHERE CURRENT OF

B. WHERE CURSOR OF

C. FOR UPDATE

D. FOR CURSOR OF

解答：C

为了对正在处理(查询)的行不被另外的用户改动，ORACLE 提供一个 FOR UPDATE 子句来对所选择的行进行锁住。该需求迫使 ORACLE 锁定游标结果集合的行，可以防止其他事务处理更新或删除相同的行，直到您的事务处理提交或回退为止。

如果使用 FOR UPDATE 声明游标，则可在 DELETE 和 UPDATE 语句中使用 WHERE CURRENT OF cursor_name 子句，修改或删除游标结果集合当前行对应的数据库表中的数据行。

218. 在 Oracle 中，使用下列的语句：CREATE PUBLIC SYNONYM parts FOR Scott.inventory;

完成的任务是（）。

A. 将 Scott.inventory 对象的访问权限赋予所有用户

B. 指定了新的对象权限

C. 指定了新的系统权限

D. 给 Scott.inventory 对象创建一个公用同义词 parts

解答：D

使用同义词访问相同的对象，方便访问其它用户的对象，短对象名字的长度。

219.在 Oracle 中，执行如下 PL/SQL 语句后

```
CREATE TYPE car AS OBJECT ( id NUMBER, model VARCHAR2(25), color VARCHAR2(15) );
DECLARE
  myvar car.model%TYPE;
BEGIN
END;
```

变量 myvar 的数据类型为（）。

A.NUMBER

B.car 类型

C.VARCHAR2

D.OBJECT

解答：C

定义一个变量，其数据类型与已经定义的某个数据变量的类型相同，或者与数据库表的某个列的数据类型相同，这时可以使用%TYPE。

220.有如下 SQL 片段

```
a. select * from asdfh a where a.kehhao in (select kehhao from retail_vip )
b. select * from asdfh a where exists (select r. kehhao from retail_vip  r where r.kehhao
= a.kehhao)
```

则以下哪些描述是正确的：

A a,b 含义相同

B a 的效率高于 b

C b 的效率高于 a

D a,b 效率高低依赖于表 asdfh 和 kehhao 的结构

E a,b 效率高低需考虑表 asdfh 和 kehhao 的数据量

解答：E

in 可以分为三类：

形如 select * from t1 where f1 in （ a , b ），应该和以下两种比较效率

select * from t1 where f1= a or f1= b

或者 select * from t1 where f1 = a union all select * from t1 f1= b

形如 select * from t1 where f1 in （select f1 from t2 where t2.fx= x）

其中子查询的 where 里的条件不受外层查询的影响，这类查询一般情况下，自动优化会转成 exist 语句，也就是效率和 exist 一样。

形如 select * from t1 where f1 in （select f1 from t2 where t2.fx=t1.fx），

其中子查询的 where 里的条件受外层查询的影响，这类查询的效率要看相关条件涉及的字段的索引情况和数据量多少，一般认为效率不如 exists.

除了第一类 in 语句都是可以转化成 exists 语句的，一般编程习惯应该是用 exists 而不 in

221.有如下表结构

客户号（kehhao）,日期（riqi）,账户余额(zhhuye)

说明表 a 中记录客户不同日期账户余额，但仅在相邻两天任一客户（kehhao）账户余额（zhhuye）发生变化时，才在表 a 中添加新记录。

日期(riqi)，其中包含多条日期：如 20050101，20050110 等

则如下 SQL 片段含义为：

Select a.kehhao,c.riqi,a.zhhuye

(Select b.riqi as riqi , max(a.riqi) as riqix from b,a where a.riqi <= b.riqi) c

Left join a

On a.riqi = c. riqix

A. 选择表 a 中各个客户在给定日期（给定日期由表 b 存储）的余额信息

B. 选择表 a 中各个客户在除给定日期（给定日期由表 b 存储）的余额信息

C. 选择表 a 中各个客户最接近给定日期（给定日期由表 b 存储）的余额信息

D. 以上都不对

解答：D

所有包含于 SELECT 列表中，而未包含于组函数中的列都必须包含于 GROUP BY 子句中。

222 有如下 SQL 片段

Delete from asdfh a where a.kaihrq > ‘20091214’ and a.jiluzt <> ‘1’其含义为：

A. 从表 asdfh 中删除 kaihrq 不小于 2009 年 12 月 14 日，且 jiluzt 不为 1 的记录

B. 从表 asdfh 中删除 kaihrq 和 jiluzt 列

C. 对表 asdfh 中删除 kaihrq 不小于 2009 年 12 月 14 日，且 jiluzt 不为 1 的记录打删除标记

D. 以上都不正确

解答：C

提交或回滚前的数据状态

改变前的数据状态是可以恢复的

执行 DML 操作的用户可以通过 SELECT 语句查询之前的修正

其他会话不能看到当前用户所做的改变，直到当前会话结束事务。

DML 语句所涉及到的行被锁定， 其他会话不能操作。

223.只有满足联接条件的记录才包含在查询结果中，这种联接为?

A. 左联接

B. 右联接

C. 内部联接

D. 完全联接

解答：C

内连接 ：内连接查询操作列出与连接条件匹配的数据行

外连接：返回到查询结果集合中的不仅包含符合连接条件的行，而且还包括左表(左连接)、右表(右连接)或两个边接表(全外连接)中的所有数据行。

224.分机构统计 VIP 客户的数量下面 SQL 语句正确的是?

其中

表 a：jigou(机构)，kehhao(客户号)，jiaoyrq(交易日期),jioyje(交易金额)

表 b: kehhao(客户号)，表 b 为 VIP 客户号表

Select _____ from a inner join b on a.kehhao = b.kehhao _____

A. Count(a.kehhao), group by jigou

B. Count(a.kehhao), order by jigou

C. Sum(a.kehhao), order by jigou

D. Count(a.kehhao), having jigou

E. 以上都不正确

解答：A

因为是统计各个分机构的VIP客户数量 所以要按照机构进行分组，而有多少条记录那么就有多少个VIP客户所以使用count

225.SQL语言中修改表结构的命令是？

A. MODIFY TABLE

B. MODIFY STRUCTURE

C. ALTER TABLE

D. ALTER STRUCTURE

解答：C

226.Delete和 truncate 都可以用来删除表内容，一下描述正确的是？

A. Truncate 不需要 rollbacksegment

B. Delete 需要rollbacksegment

C. Truncate在 提交commit之前仍可回滚

D. Truncate还可以删除表结构

解答：AB

TRUNCATE TABLE 语句:删除表中所有的数据并且释放表的存储空间,可以使用 DELETE 语句删除数据，DELETE 产生 rollback，如果删除大数据量的表速度会很慢，同时会占用很多的 rollback segments .truncate 是DDL操作，不产生rollback，速度快一些

227.请给出两个影响系统效率的函数（选两个答案）。

A. UPPER

B. SUM

C. MAX

D. LOWER

解答：BC

228、QUESTION. description of the students table:

sid_id	number
start_date	date
end_date	date

which two function are valid on the start_date column?_____。

A. sum(start_date)

B. avg(start_date)

C. count(start_date)

D. avg(start_date,end_date)

E. min(start_date)

解答：CE

sum 和 avg 要求数字数据类型。

229、which are DML statements(choose all that apply)_____.

A.commit

B.merge

C.update

D.delete

E.creat

F.drop

解答：BCD

DML：Data manipulation language 数据操众语言 insert delete update merge

DDL：data definition language 数据定义语言 create alter drop

DCL：data control language 数据控制语言 grant revoke

230、Select 语句中用来连接字符串的符号是_____.

A. “+” B. “&” C.“||” D.“|”

解答：C

231.从 “员工” 表的“姓名”字段中找出名字包含“玛丽”的人，下面哪条 select 语句正确：( )

A、 Select * from 员工 where 姓名 =’_玛丽_’

B 、Select * from 员工 where 姓名 =’%玛丽_’

C、 Select * from 员工 where 姓名 like ‘_玛丽%’

D、 Select * from 员工 where 姓名 like ‘%玛丽%’

解答：D

LIKE 运算选择类似的值，选择条件可以包含字符或数字，“%” 代表一个或多个字符，“_” 代表一个字符。

232 在关系数据库的询问优化中，事先处理文件，如排序、建立索引的目的是（）

A 优化表达式 B 减少中间结果 C 扩大缓冲数据 D 减少扫描文件的时间

解答：D

233 表 CUSTOMER 包含如下列:

CUSTOMER_ID NUMBER(9)

LAST_NAME VARCHAR2(20)

FIRST_NAME VARCHAR2(20)

CREDIT_LIMIT NUMBER(9,2)

如下代码:

DECLARE

CURSOR cust_cursor IS

SELECT customer_id, last_name, first_name

FROM customer;

cust_rec cust_cursor%ROWTYPE;

你如何操纵 CUST_REC 中的记录? (选择 1 项)

A. 添加一个 LOOP 到游标声明中.

B. 在 PL/SQL 块的执行部分，使用 INSERT INTO 语句.

C. 在 PL/SQL 块的执行部分，使用一个 LOOP 和 FETCH 语句.

D. 在 PL/SQL 块的执行部分，使用 SELECT 语句使用 INTO 操作.

解答:D

%ROWTYPE 操作符, 返回一个记录类型, 其数据类型和数据库表的数据结构相一致,

在这里就和游标查询语句中的数据结果保持一致。

例如:

DECLARE

CURSOR cust_cursor IS

SELECT CUSTOMER_ID, last_name, first_name

FROM customer;

cust_rec cust_cursor%ROWTYPE;

begin

open cust_cursor;

LOOP

FETCH cust_cursor INTO cust_rec;

EXIT WHEN cust_cursor%NOTFOUND;

```
DBMS_OUTPUT.PUT_LINE(cust_rec.last_name);
END LOOP;
end;
```

234.定义存储过程如下：

```
CREATE OR REPLACE PROCEDURE INSERT_TEAM
(V_ID in NUMBER,V_CITY in VARCHER2 DEFAULT 'AUSTIN', V_NAME
in VARCHER2)
IS
BEGIN
INSERT INTO TEAM (id, city,name)
VALUES (v_id,v_city,v_name);
COMMIT;
END;
```

以下哪些 PL/SQL 语句能够正确调用该过程？（选择 1 项）

A. EXECUTE INSERT_TEAM;

B. EXECUTE INSERT_TEAM (V_NAME=.>"LONG HORNS");

C. V_CITY=>"AUSTIN";

D. EXECUTE INSERT_TEAM (3,"AUSTIN","LONG HORNS")

解答：D

二 简答题（243）

1. J2EE 是什么？它包括哪些技术？

解答：从整体上讲，J2EE 是使用 Java 技术开发企业级应用的工业标准，它是 Java 技术不断适应和促进企业级应用过程中的产物。适用于企业级应用的 J2EE，提供一个平台独立的、可移植的、多用户的、安全的和基于标准的企业级平台，从而简化企业应用的开发、管理和部署。J2EE 是一个标准，而不是一个现成的产品。

主要包括以下这些技术：

1) Servlet

Servlet 是 Java 平台上的 CGI 技术。Servlet 在服务器端运行，动态地生成 Web 页面。与传统的 CGI 和许多其它类似 CGI 的技术相比，Java Servlet 具有更高的效率并更容易使用。对于 Servlet，重复的请求不会导致同一程序的多次转载，它是依靠线程的方式来支持并发访问的。

2)JSP

JSP(Java Server Page)是一种实现普通静态 HTML 和动态页面输出混合编码的技术。从这一点来看，非常类似 Microsoft ASP、PHP 等技术。借助形式上的内容和外观表现的分离，Web 页面制作的任务可以比较方便地划分给页面设计人员和程序员，并方便地通过 JSP 来合成。在运行时态，JSP 将会被首先转换成 Servlet，并以 Servlet 的形态编译运行，因此它的效率和功能与 Servlet 相比没有差别，一样具有很高的效率。

3) EJB

EJB 定义了一组可重用的组件：Enterprise Beans。开发人员可以利用这些组件，像搭积木一样建立分布式应用。

4)JDBC

JDBC(Java Database Connectivity，Java 数据库连接)API 是一个标准 SQL(Structured Query Language，结构化查询语言)数据库访问接口，它使数据库开发人员能够用标准 Java API 编写数据库应用程序。JDBC API 主要用来连接数据库和直接调用 SQL 命令执行各种 SQL 语句。利用 JDBC API 可以执行一般的 SQL 语句、动态 SQL 语句及带 IN 和 OUT 参数的存储过程。Java 中的 JDBC 相当于 Microsoft 平台中的 ODBC(Open Database Connectivity)。

2.测试生命周期、测试过程分为几个阶段，以及各阶段的含义？

解答：软件测试生命周期一般包括 6 个阶段：1）计划 2）分析，3）设计，4）构建，5）测试周期，6）最后测试和实施，

1） 计划：产品定义阶段

2). 分析:外部文档阶段

3). 设计:文档架构阶段

4). 构建:单元测试阶段

5). 测试周期:错误修正,重复系统测试阶段

6). 最后的测试和实施:代码冻结阶段

3.您做系统设计用何种工具？

解答：Visio, rational rose，power designer 等

4. 什么是 Web 容器?

解答：容器就是一种服务程序，在服务器一个端口就有一个提供相应服务的程序，而这个程序就是处理从客户端发出的请求，如 JAVA 中的 Tomcat 容器，ASP 的 IIS 或 PWS 都是这样的容器。

5. 运行时异常与一般异常有何异同？

解答：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。

6. Hibernate 中：不看数据库，不看 XML 文件，不看查询语句，怎么样能知道表结构？

解答：可以看与 XML 文件对应的域模型。

7.目前几种主流数据库软件的应用特点、适用范围各是什么？

解答：国际国内的主导关系型数据库管理系统有 SQL Server、ORACLE、SYBASE、INFORMIX 和 DB2。本文从性能，可伸缩性和并行性，安全性，操作简便，使用风险，开放性，易维护性和价格，数据库二次开发方面比较了 SQL Server，Oracle、SYBASE、DB2、INFORMIX 数据库：

1）性能

SQL Server：老版本多用户时性能不佳，新版本的性能有了明显的改善，各项处理能力都有了明显的提高。保持了多项 TPC-C（TPC-C 值被广泛用于衡量 C/S 环境下,由服务器和客户端构筑的整体系统的性能,它由事物处理性能委员会（TPC，Transaction Processing Corp）制定,TPC 为非赢利性国际组织。）纪录。

Oracle ：性能最高， 保持 Windows NT 下的 TPC-C 的世界记录。

SYBASE ：性能较高，支持 Sun、IBM、HP、Compaq 和 Veritas 的集群设备的特性，实现高可用性。适应于安全性要求极高的系统。

DB2 ：适用于数据仓库和在线事物处理，性能较高。 客户端支持及应用模式。

INFORMIX ：性能较高，支持集群，实现高可用性。适应于安全性要求极高的系统，尤其是银行，证券系统的应用。

2）可伸缩性,并行性

SQL Server ：以前版本 SQL Server 并行实施和共存模型并不成熟。很难处理大量的用户数和数据卷。伸缩性有限。新版本性能有了较大的改善，在 Microsoft Advanced Servers 上有突出的表现，超过了他的主要竞争对手。

Oracle ：平行服务器通过使一组结点共享同一簇中的工作来扩展 Window NT 的能力,提供高可用性和高伸缩性的簇的解决方案。如果 Windows NT 不能满足需要，用户可以把数据库移到 UNIX 中，具有很

好的伸缩性。

SYBASE ：新版本具有较好的并行性，速度快，对巨量数据无明显影响，但是技术实现复杂，需要程序支持，伸缩性有限。

DB2 ：DB2 具有很好的并行性。DB2 把数据库管理扩充到了并行的、多节点的环境。数据库分区是数据库的一部分，包含自己的数据、索引、配置文件、和事务日志。数据库分区有时被称为节点或数据库节点，伸缩性有限。

INFORMIX ：采用单进程多线程的技术，具有较好的并行性。但是仅运行于 UNIX 平台，伸缩性有限。

3）安全性

SQL server ：Microsoft Advanced Server 获得最高安全认证，服务器平台的稳定性是数据库的稳定性的基础，新版本的 SQL 的安全性有了极大的提高。

Oracle：获得最高认证级别的 ISO 标准认证。

SYBASE ：通过 Sun 公司 J2EE 认证测试，获得最高认证级别的 ISO 标准认证。

DB2 ：获得最高认证级别的 ISO 标准认证。

INFORMIX ：获得最高认证级别的 ISO 标准认证。

4）操作简便

SQL Server ：操作简单,采用图形界面。管理也很方便，而且编程接口特别友好(它的 SQL-DMO 让编程变得非常方便！），从易维护性和价格上 SQL Server 明显占有优势。

Oracle ：较复杂, 同时提供 GUI 和命令行，在 Windows NT 和 Unix， Linux 下操作相同。对数据库管理人员要求较高。

SYBASE ：复杂，使用命令行操作，对数据库管理人员要求较高。

DB2 ：操作简单,同时提供 GUI 和命令行，在 Windows NT 和 Unix 下操作相同。

INFORMIX ：使用和管理复杂，命令行操作。对数据库管理人员要求较高。

5）使用风险

SQL Server ：完全重写的代码，性能和兼容性有了较大的提高，与 Oracle，DB2 的性能差距明显减小。该产品的出台经历了长期的测试，为产品的安全和稳定进行了全面的检测，安全稳定性有了明显的提高。

Oracle ：长时间的开发经验，完全向下兼容，可以安全的进行数据库的升级，在企业，政府中得到广泛的应用。并且如果在 WINNT 上无法满足数据的要求，可以安全的把数据转移到 UNIX 上来。

SYBASE ：开发时间较长，升级较复杂，稳定性较好，数据安全有保障。风险小。在安全要求极高的银行，证券行业中得到了广泛的应用。

DB2 ：在巨型企业得到广泛的应用，向下兼容性好。风险小。

INFORMIX ：开发时间较长，升级较复杂，稳定性较好，数据安全有保障。风险小。在安全要求极高的银行，证券行业中得到了广泛的应用。

6）开放性

SQL Server：只能在 Windows 上运行，C/S 结构，只支持 Windows 客户，可以用 ADO,DAO,OLEDB,ODBC 连接。Windows9X 系列产品是偏重于桌面应用，NT server 适合各种大中小型企业。 操作系统的稳定对数据库是十分重要的。Windows 平台的可靠性，安全性经过了最高级别的 C2 认证的。在处理大数据量的关键业务时提供了较好的性能。

Oracle ：能在所有主流平台上运行（包括 Windows）。完全支持所有的工业标准。采用完全开放策略。多层次网络计算，支持多种工业标准，可以用 ODBC,JDBC,OCI 等网络客户连接。可以使客户选择最适合的解决方案。对开发商全力支持。

SYBASE ：能在所有主流平台上运行，在银行业中得到了广泛的应用。

DB2 ：有较好的开放性，最适于海量数据。跨平台，多层结构，支持 ODBC,JDBC 等客户。在大型的国际企业中得到最为广泛的应用,在全球的 500 家最大的企业中,大部分采用 DB2 数据库服务器。

IINFORMIX ：仅运行在 UNIX 平台，包括 SUNOS、HPUX、 ALFAOSF/1。在银行中得到广泛的应用。

7）易维护性和价格

SQL Server ：从易维护性和价格上 SQL Server 明显占有优势。基于 Microsoft 的一贯风格,SQL Server 的图形管理界面带来了明显的易用性， 微软的数据库管理员培训进行的比较充分，可以轻松的找到很好的数据库管理员，数据库管理费用比较低，SQL Server 的价格也是很低的，但是在 License 的购买上会抬高价格。总体来说 SQL Server 的价格在商用数据库中是最低的。

Oracle :从易维护性和价格上来说 Oracle 的价格是比较高的，管理比较复杂，由于 Oracle 的应用很广泛，经验丰富的 Oracle 数据库管理员可以比较容易的找到，从而实现 Oracle 的良好管理。因此 Oracle 的性能价格比在商用数据库中是最好的。

SYBASE :SYBASE 的价格是比较低的，但是 SYBASE 的在企业和政府中的应用较少，很难找到经验丰富的管理员，运行管理费用较高。

DB2 :价格高， 管理员少，在中国的应用较少，运行管理费用都很高，适用于大型企业的数据仓库应用。

INFORMIX :价格在这些系统中居于中间，与 SYBASE 一样，在企业和政府中应用较少，仅在银行中得到了广泛的应用。经验丰富的管理人员较少，运行管理费用高。

8)数据库二次开发

SQL Server :数据库的二次开发工具很多，包括 Visual C++, Visual Basic 等开发工具，可以实现很好的 Windows 应用，开发容易。

Oracle :数据库的二次开发工具很多，涵盖了数据库开发的各个阶段，开发容易。

SYBASE :开发工具较少，经验丰富的人员很少。

DB2 :在国外巨型企业得到广泛的应用，中国的经验丰富的人员很少。

INFORMIX :在银行业中得到广泛的应用，但是在中国的经验丰富的人员很少。

8.存储过程和函数的区别

解答：

从参数的返回情况来看：

如果返回多个参数值最好使用存储过程，如果只有一个返回值的话可以使用函数。

从调用情况来看：

如果在 SQL 语句（DML 或 SELECT）中调用的话一定是存储函数或存储的封装函数不可以是存储过程，但调用存储函数的时候还有好多限制以及函数的纯度等级的问题，如果是在过程化语句中调用的话，就要看你要实现什么样的功能。函数一般情况下是用来计算并返回一个计算结果而存储过程一般是用来完成特定的数据操作（比如修改、插入数据库表或执行某些 DDL 语句等等），所以虽然他们的语法上很相似但用户在使用他们的时候所需要完成的功能大部分情况下是不同的。

9.试述数据库完整保护的主要任务和措施。

解答：数据库的完整性保护也就是数据库中数据正确性的维护。数据库完整性包括三个内容：实体完整性规则，参照物完整性规则以及用户定义完整性规则。 前两个是有 DBMS 自动处理。

实体完整性规则是说针对于基表中的关键字中属性值不能为空值，是数据库完整性的基本要求，主关键字和元组的唯一性对应。

参照物完整性规则是不允许引用不存在的元组：即基表中的外关键字要么为空，要么关联基表中必存在元组。

用户定义完整性规则针对具体的数据环境由用户具体设置的规则，它反应了具体应用中的语义要求。

一个完整性规则一般由下面三部分组成:完整性约束条件设置,完整性约束条件的检查以及完整性约束条件的处理.后两部分在数据库中一般有相应的模块处理。另外触发器也可以做完整性的保护，但触发器大量用于主动性领域。

10.请说明 SQLServer 中 delete from tablea & truncate table tablea 的区别

解答：两者都可以用来删除表中所有的记录。区别在于：truncate 是 DDL 操作，它移动 HWK，使 HWK 值为 0，不需要 rollback segment .而 Delete 是 DML 操作需要 rollback segment 且花费较长时间.

11. Oracle 安装完成后,如何用命令行启动和关闭数据库?

解答：

打开：STARTUP [FORCE] [RESTRICT] [PFILE= filename] [OPEN [RECOVER][database] | MOUNT | NOMOUNT]

STARTUP OPEN：STARTUP 缺省的参数就是 OPEN，打开数据库，允许数据库的访问。当前实例的控制文件中所描述的所有文件都已经打开。

STARTUP MOUNT：MOUNT 数据库，仅仅给 DBA 进行管理操作，不允许数据库的用户访问。仅仅只是当前实例的控制文件被打开，数据文件未打开。

STARTUP NOMOUNT：仅仅通过初始化文件，分配出 SGA 区，启动数据库后台进程，没有打开控制文件和数据文件。不能访问任何数据库。

STARTUP PFILE= filename：以 filename 为初始化文件启动数据库，不是采用缺省初始化文件。

STARTUP FORCE：中止当前数据库的运行，并开始重新正常的启动数据库。

STARTUP RESTRICT：只允许具有 RESTRICTED SESSION 权限的用户访问数据库。

STARTUP RECOVER：数据库启动，并开始介质恢复

关闭

SHUTDOWN 有四个参数:NORMAL、TRANSACTIONAL、IMMEDIATE、ABORT。缺省不带任何参数时表示是 NORMAL。

命令 SHUTDOWN NORMAL：不允许新的连接、等待会话结束、等待事务结束、做一个检查点并关闭数据文件。启动时不需要实例恢复。

SHUTDOWN TRANSACTIONAL：不允许新的连接、不等待会话结束、等待事务结束、做一个检查点并关闭数据文件。启动时不需要实例恢复。

SHUTDOWN IMMEDIATE：不允许新的连接、不等待会话结束、不等待事务结束、做一个检查点并关闭数据文件。没有结束的事务是自动 rollback 的。启动时不需要实例恢复。

SHUTDOWN ABORT：不允许新的连接、不等待会话结束、不等待事务结束、不做检查点且没有关闭数据文件。启动时自动进行实例恢复。

另外，对于 NORMAL、TRANSACTIONAL、IMMEDIATE，DB Buffer Cache 的内容写入了数据文件，没有提交的事务被回滚，所有的资源被释放，数据库被“干净”的关闭。

对于 ABORT，DB Buffer Cache 的内容没有写入数据文件，没有提交的事务也没有回滚。数据库没有 dismount 和关闭，数据文件也没有关闭。当数据库启动时，需要通过 redo log 恢复数据，通过回滚段对事务回滚，对资源进行释放。

12. 类有哪三个基本特性？各特性的优点？

解答：类具有封装性、继承性和多态性。

封装性：类的封装性为类的成员提供公有、缺省、保护和私有等多级访问权限，目的是隐藏类中的私有变量和类中方法的实现细节。

继承性：类的继承性提供从已存在的类创建新类的机制，继承（inheritance）使一个新类自动拥有被继承类（父类）的全部可继承的成员。

多态性：类的多态性提供类中方法执行的多样性，多态性有两种表现形式：重载和覆盖。

13.谈谈对 XML 的理解？说明 Web 应用中 Web.xml 文件的作用？

解答：XML（Extensible Markup Language）即可扩展标记语言，它与 HTML 一样，都是 SGML(Standard Generalized Markup Language,标准通用标记语言)。Xml 是 Internet 环境中跨平台的，依赖于内容的技术，是当前处理结构化文档信息的有力工具。扩展标记语言 XML 是一种简单的数据存储语言，使用一系列简单的标记描述数据，而这些标记可以用方便的方式建立，虽然 XML 占用的空间比二进制数据要占用更多的空间，但 XML 极其简单易于掌握和使用。

web.xml 的作用是配置欢迎页，servlet，filter，listener 等的。

14.jsp 有哪些内置对象？作用分别是什么？（至少三个）

解答：

1）request 表示 HttpServletRequest 对象。它包含了有关浏览器请求的信息，并且提供了几个用于获取 cookie, header 和 session 数据的有用的方法。

2）response 表示 HttpServletResponse 对象，并提供了几个用于设置送回 浏览器的响应的方法（如 cookies,头信息等）。

3）out 对象是 javax.jsp.JspWriter 的一个实例，并提供了几个方法使你能用于向浏览器回送输出结果。

4）pageContext 表示一个 javax.servlet.jsp.PageContext 对象。它是用于方便存取各种范围的名字空间、servlet 相关的对象的 API，并且包装了通用的 servlet 相关功能的方法。

5）session 表示一个请求的 javax.servlet.http.HttpSession 对象。Session 可以存贮用户的状态信息。

6）application 表示一个 javax.servle.ServletContext 对象。这有助于查找有关 servlet 引擎和 servlet 环境的信息。

7）config 表示一个 javax.servlet.ServletConfig 对象。该对象用于存取 servlet 实例的初始化参数。

8）page 表示从该页面产生的一个 servlet 实例。

9）exception 针对错误网页，未捕捉的例外

15.事务是什么？有哪些属性，并简要说明这些属性的含义。

解答：事务(Transaction)是访问并可能更新数据库中各种数据项的一个程序执行单元(unit)。事务通常由高级数据库操纵语言或编程语言（如 SQL，C++或 Java）书写的用户程序的执行所引起，并用形如 begin transaction 和 end transaction 语句（或函数调用）来界定。事务由事务开始(begin transaction)和事务结束(end transaction)之间执行的全体操作组成。

事务应该具有 4 个属性：原子性、一致性、隔离性、持续性。这四个属性通常称为 ACID 特性。

原子性（atomicity)。一个事务是一个不可分割的工作单位，事务中包括的诸操作要么都做，要么都不做。

一致性（consistency)。事务必须是使数据库从一个一致性状态变到另一个一致性状态。一致性与原子性是密切相关的。

隔离性（isolation)。一个事务的执行不能被其他事务干扰。即一个事务内部的操作及使用的数据对并发的其他事务是隔离的，并发执行的各个事务之间不能互相干扰。

持久性（durability)。持续性也称永久性（permanence)，指一个事务一旦提交，它对数据库中数据的改变就应该是永久性的。接下来的其他操作或故障不应该对其有任何影响。

16、Collection 和 Collections 的区别？

解答：Collection 是 java.util 下的接口，它是各种集合的父接口，继承于它的接口主要有 Set 和 List；Collections 是个 java.util 下的类，是针对集合的帮助类，提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。

17、HashMap 与 TreeMap 的区别？

解答：HashMap 通过 hashcode 对其内容进行快速查找，而 TreeMap 中所有的元素都保持着某种固定的顺序，如果你需要得到一个有序的结果你就应该使用 TreeMap（HashMap 中元素的排列顺序是不固定的)。

18、ArrayList 和 Vector 的区别？

解答：同步性:Vector 是线程安全的，也就是说是同步的，而 ArrayList 是线程不安全的，不是同步的；数据增长:当需要增长时,Vector 默认增长为原来一培，而 ArrayList 却是原来的一半。

19、HashMap 和 Hashtable 的区别？

解答：HashMap 是 Hashtable 的轻量级实现（非线程安全的实现)，他们都实现了 Map 接口，主要区别在于 HashMap 允许空（null）键值（key),由于非线程安全，效率上高于 Hashtable。HashMap 允许将 null 作为一个 entry 的 key 或者 value，而 Hashtable 不允许。HashMap 把 Hashtable 的 contains

方法去掉了，改成 containsvalue 和 containsKey。因为 contains 方法容易让人引起误解。Hashtable 继承自 Dictionary 类，而 HashMap 是 Java1.2 引进的 Map interface 的一个实现。最大的不同是，Hastable 的方法是 synchronize 的，而 HashMap 不是，在多个线程访问 Hashtable 时，不需要自己为它的方法实现同步，而 HashMap 就必须为之提供同步。

20.请说出 ArrayList,Vector, LinkedList 的存储性能和特性

解答：ArrayList 和 Vector 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector 由于使用了 synchronized 方法（线程安全），通常性能上较 ArrayList 差，而 LinkedList 使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

21. 描述 J2EE 框架的多层结构，并简要说明各层的作用。

解答：

1) Presentation layer（表示层）

a. 表示逻辑（生成界面代码）

b. 接收请求

c. 处理业务层抛出的异常

d. 负责规则验证（数据格式，数据非空等）

e. 流程控制

2) Service layer（服务层/业务层）

a.封装业务逻辑处理，并且对外暴露接口

b.负责事务，安全等服务

3) Persistence layer（持久层）

a 封装数据访问的逻辑，暴露接口

b.提供方便的数据访问的方案（查询语言，API，映射机制等）

4）Domain layer（域层）

a. 业务对象以及业务关系的表示

b. 处理简单的业务逻辑

c. 域层的对象可以穿越表示层，业务层，持久层

软件分层结构使得代码维护非常方便，设计明确，各层独立，专注自己擅长的领域。

22.请谈谈对 SOA 的认识。

解答：面向服务的体系结构（Service-Oriented Architecture，SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以一种统一和通用的方式进行交互。

23.简要描述如何结合 struts、hibernate、spring 开发 Web 应用？

解答：Struts 可以将 jsp 页面的表单关联起来,就是把 JSP 页面的表单数据封装成 javaBean,这样的话,在 action 中你再也不需要使用传统的 request.getParameter("name");还有 struts 有一个控制器,你在 struts 编程中的控制器(XxxAction)都是继承总的 ActionServlet,它能集中处理请求,然后转到相关的页面。还有 struts 的表单验证组件,不用你写 js 验证了,只需要你配置一下文件就可以了。另外 struts 的令牌机制可以防表单重复提交。

Spring 是一个轻量级容器,非侵入性.包含依赖注入,AOP 等。它是为了解决企业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构，分层架构允许您选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成的框架。

Hibernate:它可以让我们以 OO 的方式操作数据库，这让我们看到了 hibernate 的强大之处，体验到操作数据的方便。但 hibernate 最耀眼之处是 hibernate 的缓存机制，而不是以 OO 的方式操作数据库。Hibernate 的缓存机制不外乎是一级缓存 session，二级缓存 sessionFactory，和第三方缓存如 ehcache。也就是 hibernate 的最强大的地方是它的缓存，理解了这个才能真正的理解 hibernate,Hibernate 的命名查询/命名参数查询，就是将 hql 语句放在一个单独的 xml 文件之中，它仍然让人们以面向对象的方式去操纵数据，而不用在以 OO 的方式写着代码的同时，然后再转变思维，用面向关系的方式去写那些 sql 语句。但 hibernate 不仅做了这些，它的 native sql 查询方式，完全满足 sql 语句的偏爱者，它像 ibatis 一样，将 sql 语句放在配置文件之中。

24.说明反转控制（IOC）和面向方向编程（AOP）在 spring 中的应用

解答：Spring 核心容器（Core）提供 Spring 框架的基本功能。核心容器的主要组件是 BeanFactory，它是工厂模式的实现。BeanFactory 使用控制反转（Ioc）模式将应用程序的配置和依赖性规范与实际的应用代码程序分开。Spring 的声明式事务基于 AOP 实现，却并不需要程序开发者成为 AOP 专家，亦可轻易使用 Spring 的声明式事务管理。

25.请看如下片段：

```
<set        name=“address”
            lazy=“true”
            inverse=“false”
```

```
        cascade=“all-delete-orphan”>
        <key column = “USERID”></key>
        <one-to-many class=“com.norteksoft.erm.model.Address”/>
    </ set >
```

解释 lazy、inverse、cascade 以及 all-delete-orphan 属性的含义；并给出示例代码，说明在如下组合情况下，对于 save、update、delete 一对多关系中的一方对象操作时的区别：

inverse	cascade
true	all-delete-orphan
false	all-delete-orphan
true	all
false	all
true	none
false	none

解答：

(一)lazy:延迟加载

Lazy 的有效期：只有在 session 打开的时候才有效；session 关闭后 lazy 就没效了。

lazy 策略可以用在：

a. <class>标签上：可以取值 true/false

b.<property>标签上，可以取值 true/false，这个特性需要类增强

c.<set>/<list>等集合上，可以取值为 true/false/extra

d.<one-to-one>/<many-to-one>等标签上，可以取值 false/proxy/no-proxy

1) get 和 load 的区别：

a.get 不支持延迟加载，而 load 支持。

b.当查询特定的数据库中不存在的数据时，get 会返回 null，而 load 则抛出异常。

2) 类(Class)的延迟加载：

a. 设置<class>标签中的 lazy="true",或是保持默认（即不配置 lazy 属性）

b. 如果 lazy 的属性值为 true，那么在使用 load 方法加载数据时，只有确实用到数据的时候才会发出 sql 语句；这样有可能减少系统的开销。

3) 集合(collection)的延迟加载：可以取值 true，false，extra

a.true:默认取值，它的意思是只有在调用这个集合获取里面的元素对象时，才发出查询语句，加载其集合元素的数据

b.false:取消懒加载特性，即在加载对象的同时，就发出第二条查询语句加载其关联集合的数据

c.extra:一种比较聪明的懒加载策略，即调用集合的 size/contains 等方法的时候，hibernate 并不会去加载整个集合的数据，而是发出一条聪明的 SQL 语句，以便获得需要的值，只有在真正需要用到这些集合元素对象数据的时候，才去发出查询语句加载所有对象的数据

4) Hibernate 单端关联懒加载策略：即在<one-to-one>/<many-to-one>标签上可以配置

懒加载策略。可以取值为：false/proxy/no-proxy

a.false:取消懒加载策略，即在加载对象的同时，发出查询语句，加载其关联对象

b.proxy:这是 hibernate 对单端关联的默认懒加载策略，即只有在调用到其关联对象的方法的时候才真正发出查询语句查询其对象数据，其关联对象是代理类

c.no-proxy:这种懒加载特性需要对类进行增强，使用 no-proxy，其关联对象不是代理类

注意：在 class 标签上配置的 lazy 属性不会影响到关联对象!!!

（二）inverse

inverse 是指的关联关系的控制方向，inverse=false 的 side（side 其实是指 inverse＝false 所位于的 class 元素）端有责任维护关系，而 inverse＝true 端无须维护这些关系

（三）cascade

cascade 指的是层级之间的连锁操作。在定义关联对象的映射时，使用 cascade="all"，cascade="save-update",cascade="all-delete-orphan"或 cascade="delete"

a. 如果父对象被保存，所有的子对象会被传递到 saveOrUpdate() 方法去执行(cascade="save-update")

b.如果父对象被传递到 update()或者 saveOrUpdate()，所有的子对象会被传递到 saveOrUpdate()方法去执行(cascade="save-update")

c.如果一个临时的子对象被一个持久化的父对象引用了，它会被传递到 saveOrUpdate()去执行(cascade="save-update")

d.如果父对象被删除了，所有的子对象对被传递到 delete()方法执行 (cascade="delete")

e.如果临时的子对象不再被持久化的父对象引用，什么都不会发生（必要时，程序应该明确的删除这个子对象），除非声明了 cascade="all-delete-orphan"，在这种情况下，成为“孤儿”的子对象会被删除。

(四) save、update、delete 一对多关系中的一方对象操作时的区别

1）当一方设置 inverse=true 时，所有由一方发出的操作都不会关联到多方。

2)当一方设置 inverse=false, cascade=all-delete-orphan 时，将删除不再和一方对象关联的所有多方对象。

3）当一方设置 inverse=false, cascade=all 时,当保存和删除一方对象时，级联保存和删除所有关联的多方对象。

4）当一方设置 inverse=false, cascade=none 时,当对一方操作时，不级联到关联的多方对象。

26.简单说明什么是递归？什么情况会使用？并使用 java 实现一个简单的递归程序。

解答：

1）递归做为一种算法在程序设计语言中广泛应用.是指函数/过程/子程序在运行过程中直接或间接调用自身而产生的重入现象。

2）递归算法一般用于解决三类问题：

a.数据的定义是按递归定义的。(Fibonacci（斐波那契）函数)

b.问题解法按递归算法实现。(回溯)

c.数据的结构形式是按递归定义的。(树的遍历，图的搜索)

3）.这是一个排列的例子，它所做的工作是将输入的一个字符串中的所有元素进行排序并输出，例如：你给出的参数是"abc" 则程序会输出：

abc

acb

bac

bca

cab

cba

a.算法的出口在于： l o w＝h i g h 也就是现在给出的排列元素只有一个时。

b.算法的逼近过程：先确定排列的第一位元素，也就是循环中 i 所代表的元素，然后 l o w＋1 开始减少排列元素，如此下去，直到 l o w＝h i g h

```
public class Foo {
    public static void main(String[] args) {
        permute("abc");
    }
    public static void permute(String str) {
```

```
        char[] strArray = str.toCharArray();
        permute(strArray, 0, strArray.length - 1);
    }
    public static void permute(char[] list, int low, int high) {
        int i;
        if (low == high) {
            String cout = "";
            for (i = 0; i <= high; i++)
                cout += list[i];
            System.out.println(cout);
        } else {
            for (i = low; i <= high; i++) {
                char temp = list[low];
                list[low] = list[i];
                list[i] = temp;
                permute(list, low + 1, high);
                temp = list[low];
                list[low] = list[i];
                list[i] = temp;
            }
        }
    }
}
```

27.列出自己常用的 jdk 包.

解答：JDK 常用的 package

java.lang： 这个是系统的基础类，比如 String 等都是这里面的，这个 package 是唯一一个可以不用 import 就可以使用的 Package

java.io: 这里面是所有输入输出有关的类，比如文件操作等

java.net: 这里面是与网络有关的类，比如 URL,URLConnection 等。

java.util : 这个是系统辅助类，特别是集合类 Collection,List,Map 等。

java.sql: 这个是数据库操作的类，Connection, Statememt，ResultSet 等

28.列出自己常用的 jdk 中的数据结构

解答：线性表，链表，哈希表是常用的数据结构。

29. List、Map、Set 三个接口存储元素时各有什么特点？

解答：

1）List 是有序的 Collection，使用此接口能够精确的控制每个元素插入的位置。用户能够使用索引（元素在 List 中的位置，类似于数组下标）来访问 List 中的元素，这类似于 Java 的数组。

2）Set 是一种不包含重复的元素的 Collection，即任意的两个元素 e1 和 e2 都有 e1.equals(e2)=false，Set 最多有一个 null 元素。

3）Map 接口 ：请注意，Map 没有继承 Collection 接口，Map 提供 key 到 value 的映射

30. 简述基于 Struts 框架 Web 应用的工作流程

解答：在 web 应用启动时就会加载初始化 ActionServlet,ActionServlet 从 struts-config.xml 文件中读取配置信息,把它们存放到各种配置对象中， 当 ActionServlet 接收到一个客户请求时,将执行如下流程.

1)检索和用户请求匹配的 ActionMapping 实例,如果不存在,就返回请求路径无效信息;

2)如果 ActionForm 实例不存在,就创建一个 ActionForm 对象,把客户提交的表单数据保存到 ActionForm 对象中;

3)根据配置信息决定是否需要表单验证.如果需要验证,就调用 ActionForm 的 validate()方法;

4)如果 ActionForm 的 validate()方法返回 null 或返回一个不包含 ActionMessage 的 ActionErrors 对象, 就表示表单验证成功;

5)ActionServlet 根据 ActionMapping 所包含的映射信息决定将请求转发给哪个 Action,如果相应的 Action 实例不存在,就先创建这个实例,然后调用 Action 的 execute()方法;

6)Action 的 execute()方法返回一个 ActionForward 对象,ActionServlet 在把客户请求转发给 ActionForward 对象指向的 JSP 组件;

7)ActionForward 对象指向 JSP 组件生成动态网页,返回给客户;

31. 在项目中用过 Spring 的哪些方面？及用过哪些 Ajax 框架？

解答：在项目使用过 Spring IOC ，AOP，DAO，ORM，还有上下文环境。

在项目使用过 Ext,Juery 等 Ajax 框架.

32、abstract class 和 interface 有什么区别?

解答：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。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。新型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，所有成员变量都是 public static final 的。一个类可以实现多个接口，当类实现接口时，必须实现接口的所有方法。抽象类在 Java 语言中表示的是一种单继承的关系，对于 interface 来说则不然，并不要求 interface 的实现者和 interface 定义在概念本质上是一致的，仅仅是实现了 interface 定义的契约而已。；抽象类中可以定义自己的成员变量，也可以包含非抽象的方法，而在接口中只能有静态的常量，所有方法必须是抽象的；实现抽象类时可以只实现其中的部分方法，而要是实现一个接口的话就必须实现这个接口中的所有抽象方法。

33.MVC 模式中 M，V，C 每个代表意义，并简述在 Struts 中 MVC 的表现方式。

解答：

MVC 是 Model-View-Controller 的缩写，Model 代表的是应用的业务逻辑（通过 JavaBean，EJB 组件实现），View 是应用的表示层（由 JSP 页面产生）Controller 是通过应用的处理过程控制，（一般是一个 servert）通过这种设计模型把应用逻辑，处理过程和显示逻辑分成不同的组件实现，这些组件可以进行交互和重用。

在 Struts 框架中 Controller 功能由 ActionServlet 和 ActionMapping 对象构成，核心是一个 Servlet 类型的对象 ActionServlet，它用来接收客户端的请求。ActionServlet 包括一组基于配置的 ActionMapping 对象，每个 ActionMapping 对象实现了一个请求到一个具体的 Model 部分的 Action 处理器对象之间的映射。Model 部分由 Action 和 ActionForm 对象构成。所有的 Action 处理器对象都是开发者从 Struts 的 Action 类派生的子类。Action 处理器对象封装了具体的处理逻辑，调用业务逻辑模块，并且把响应提交到合适的 View 组件以产生响应。Struts 提供的 ActionForm 组件对象可以通过定义属性描述客户端表单数据，开发者可以从它派生子类对象，并利用它和 Struts 提供的自定义标记库相结合，可以实现对客户端的表单数据的良好封装和支持，Action 处理器对象可以直接对它进行读写，而不再需要和 request、response 对象进行数据交互。通过 ActionForm 组件对象实现了对 View 和 Model 之间交互的支持（View 部分是通过 JSP 技术实现的）。Struts 提供了自定义的标记库，通过这些自定义标记库可以非常容易地和系统的 Model 部分交互，通过使用这些自定义标记库创建的 JSP 表单，可以实现对 Model 部分中的 ActionForm 的映射，完成对用户数据的封装。

34. java 语言中 public、private、protected 三个关键字的用法,重写和重载的区别。

解答：

作用域	当前类	同包	子类	其它
public	√	√	√	√
protected	√	√	√	×
default	√	√	×	×
private	√	×	×	×

重写：发生在父子类之间，方法名相同，参数的类型、个数、顺序相同，返回值相同，访问权限不能更封闭，抛出异常不能宽泛；

重载：发生在同一个类中，方法名相同，但是参数不同（类型不同或个数不同或参数的顺序不同），返回值可以不相同。

35.JSP 页面之间传递参数的方法有哪些？

解答：

1）request

2）session

3）application

4）提交表单

5）超链接

36.forward 和 redirect 的区别

解答：forward 是容器中控制权的转向，是服务器请求资源，服务器直接访问目标地址的 URL，把那个 URL 的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。 redirect 就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求，并且从浏览器的地址栏中可以看到跳转后的链接地址。前者更加高效，在前者可以满足需要时，尽量使用 forward()方法，并且，这样也有助于隐藏实际的链接；在有些情况下，比如，需要跳转到一个其它服务器上的资源，则必须使用 sendRedirect()方法。

37.Java 反射机制的作用？

解答：Java 反射机制的作用是：

1）在运行时判断任意一个对象所属的类。

2）在运行时构造任意一个类的对象。

3）在运行时判断任意一个类所具有的成员变量和方法。

4）在运行时调用任意一个对象的方法

38. 你是怎么理解 java 的泛型的?

解答： 在 Java SE 1.5 之前，没有泛型的情况的下，通过对类型 Object 的引用来实现参数的“任意化”，“任意化”带来的缺点是要做显式的强制类型转换，而这种转换是要求开发者对实际参数类型可以预知的情况下进行的。对于强制类型转换错误的情况，编译器可能不提示错误，在运行的时候才出现异常，这是一个安全隐患。

泛型是 Java SE 1.5 的新特性，泛型的本质是参数化类型，也就是说所操作的数据类型被指定为一个参数。这种参数类型可以用在类、接口和方法的创建中，分别称为泛型类、泛型接口、泛型方法。

泛型的好处是在编译的时候检查类型安全，并且所有的强制转换都是自动和隐式的，提高代码的重用率。

39.JAVA 源文件中是否可以包括多个类,有什么限制

解答：一个 java 源文件中可以包含多个类，每个源文件中至多有一个 public 类，如果有的话，那么源文件的名字必须与之相同。如果源文件中没有 public 类，则源文件用什么名字都可以，但最好还是具有特定的意义，免得自己都不记得里面写的是什么了。

40. 在一个千万级的数据库查寻中，如何提高查询效率？分别说出在数据库设计、SQL 语句、java 等层面的解决方案。

解答：

1）数据库设计方面：

a. 对查询进行优化，应尽量避免全表扫描，首先应考虑在 where 及 order by 涉及的列上建立索引。

b. 应尽量避免在 where 子句中对字段进行 null 值判断，否则将导致引擎放弃使用索引而进行全表扫描，如：

```
select id from t where num is null
```

可以在 num 上设置默认值 0，确保表中 num 列没有 null 值，然后这样查询：

```
select id from t where num=0
```

c. 并不是所有索引对查询都有效，SQL 是根据表中数据来进行查询优化的，当索引列有大量数据重复时,查询可能不会去利用索引，如一表中有字段 sex，male、female 几乎各一半，那么即使在 sex 上建了索引也对查询效率起不了作用。

d. 索引并不是越多越好，索引固然可以提高相应的 select 的效率，但同时也降低了 insert 及 update 的效率，因为 insert 或 update 时有可能会重建索引，所以怎样建索引需要慎重考虑，视

具体情况而定。一个表的索引数最好不要超过 6 个，若太多则应考虑一些不常使用到的列上建的索引是否有必要。

e. 应尽可能的避免更新索引数据列，因为索引数据列的顺序就是表记录的物理存储顺序，一旦该列值改变将导致整个表记录的顺序的调整，会耗费相当大的资源。若应用系统需要频繁更新索引数据列，那么需要考虑是否应将该索引建为索引。

f. 尽量使用数字型字段，若只含数值信息的字段尽量不要设计为字符型，这会降低查询和连接的性能，并会增加存储开销。这是因为引擎在处理查询和连接时会逐个比较字符串中每一个字符，而对于数字型而言只需要比较一次就够了。

g. 尽可能的使用 varchar/nvarchar 代替 char/nchar ，因为首先变长字段存储空间小，可以节省存储空间，其次对于查询来说，在一个相对较小的字段内搜索效率显然要高些。

h. 尽量使用表变量来代替临时表。如果表变量包含大量数据，请注意索引非常有限（只有主键索引）。

i. 避免频繁创建和删除临时表，以减少系统表资源的消耗。

j. 临时表并不是不可使用，适当地使用它们可以使某些例程更有效，例如，当需要重复引用大型表或常用表中的某个数据集时。但是，对于一次性事件，最好使用导出表。

k. 在新建临时表时，如果一次性插入数据量很大，那么可以使用 select into 代替 create table，避免造成大量 log ，以提高速度；如果数据量不大，为了缓和系统表的资源，应先 create table，然后 insert。

l. 如果使用到了临时表，在存储过程的最后务必将所有的临时表显式删除，先 truncate table ，然后 drop table ，这样可以避免系统表的较长时间锁定。

2)SQL 语句方面：

a. 应尽量避免在 where 子句中使用!=或<>操作符，否则将引擎放弃使用索引而进行全表扫描。

b. 应尽量避免在 where 子句中使用 or 来连接条件，否则将导致引擎放弃使用索引而进行全表扫描，如：

```
select id from t where num=10 or num=20
```

可以这样查询：

```
select id from t where num=10
union all
select id from t where num=20
```

c. in 和 not in 也要慎用，否则会导致全表扫描，如：

```
select id from t where num in(1,2,3)
```

对于连续的数值，能用 between 就不要用 in 了：

select id from t where num between 1 and 3

d. 下面的查询也将导致全表扫描：

select id from t where name like '%abc%'

e. 如果在 where 子句中使用参数，也会导致全表扫描。因为 SQL 只有在运行时才会解析局部变量，但优化程序不能将访问计划的选择推迟到运行时；它必须在编译时进行选择。然而，如果在编译时建立访问计划，变量的值还是未知的，因而无法作为索引选择的输入项。如下面语句将进行全表扫描：

select id from t where num=@num

可以改为强制查询使用索引：

select id from t with(index(索引名)) where num=@num

f. 应尽量避免在 where 子句中对字段进行表达式操作，这将导致引擎放弃使用索引而进行全表扫描。如：

select id from t where num/2=100

应改为:

select id from t where num=100*2

g. 应尽量避免在 where 子句中对字段进行函数操作，这将导致引擎放弃使用索引而进行全表扫描。如：

select id from t where substring(name,1,3)='abc'--name 以 abc 开头的 id

select id from t where datediff(day,createdate,'2005-11-30')=0--‘2005-11-30’生成的 id

应改为:

select id from t where name like 'abc%'

select id from t where createdate>='2005-11-30' and createdate<'2005-12-1'

h. 不要在 where 子句中的“=”左边进行函数、算术运算或其他表达式运算，否则系统将可能无法正确使用索引。

i. 不要写一些没有意义的查询，如需要生成一个空表结构：

select col1,col2 into #t from t where 1=0

这类代码不会返回任何结果集，但是会消耗系统资源的，应改成这样：

create table #t(...)

j. 很多时候用 exists 代替 in 是一个好的选择：

select num from a where num in(select num from b)

用下面的语句替换：

select num from a where exists(select 1 from b where num=a.num)

k. 任何地方都不要使用 select * from t ，用具体的字段列表代替“*”，不要返回用不到的任何字段。

l. 尽量避免使用游标，因为游标的效率较差，如果游标操作的数据超过 1 万行，那么就应该考虑改写。

m. 尽量避免向客户端返回大数据量，若数据量过大，应该考虑相应需求是否合理。

n. 尽量避免大事务操作，提高系统并发能力。

3)java 方面：

a.尽可能的少造对象。

b.合理摆正系统设计的位置。大量数据操作，和少量数据操作一定是分开的。大量的数据操作，肯定不是 ORM 框架搞定的。，

c.使用 jDBC 链接数据库操作数据

d.控制好内存，让数据流起来，而不是全部读到内存再处理，而是边读取边处理；

e.合理利用内存，有的数据要缓存

41.请解释分布式事务管理？

解答：分布式事务是指事务的参与者、支持事务的服务器、资源服务器以及事务管理器分别位于不同的分布式系统的不同节点之上。为了实现分布式事务，需要使用下面将介绍的两阶段提交协议。 阶段一：开始向事务涉及到的全部资源发送提交前信息。此时，事务涉及到的资源还有最后一次机会来异常结束事务。如果任意一个资源决定异常结束事务，则整个事务取消，不会进行资源的更新。否则，事务将正常执行，除非发生灾难性的失败。为了防止会发生灾难性的失败，所有资源的更新都会写入到日志中。这些日志是永久性的，因此，这些日志会幸免遇难并且在失败之后可以重新对所有资源进行更新。 阶段二：只在阶段一没有异常结束的时候才会发生。此时，所有能被定位和单独控制的资源管理器都将开始执行真正的数据更新。 在分布式事务两阶段提交协议中，有一个主事务管理器负责充当分布式事务协调器的角色。事务协调器负责整个事务并使之与网络中的其他事务管理器协同工作。 为了实现分布式事务，必须使用一种协议在分布式事务的各个参与者之间传递事务上下文信息，IIOP 便是这种协议。这就要求不同开发商开发的事务参与者必须支持一种标准协议，才能实现分布式的事务。

42.请写出一段表单提交的 HTML 代码，表单名称为 form1，提交方式为 post，提交地址为 submit.asp

解答：

〈form name=”form1” method=”post” action=”submit.jsp”〉

〈input type=”submit”〉

〈/form

43.请写出一个超链接，点击链接后可以向 zhangsan@d-heaven.com 发送电子邮件。

〈a href=”mailto: zhangsan@d-heaven.com”〉发邮件〈/a〉

44.请说明 meta 标签的作用。

解答：

meta 是用来在 HTML 文档中模拟 HTTP 协议的响应头报文。meta 标签用于网页的＜head＞与＜/head＞中，meta 标签的用处很多。meta 的属性有两种：name 和 http-equiv。name 属性主要用于描述网页，对应于 content（网页内容），以便于搜索引擎机器人查找、分类（目前几乎所有的搜索引擎都使用网上机器人自动查找 meta 值来给网页分类）。这其中最重要的是 description（站点在搜索引擎上的描述）和 keywords（分类关键词），所以应该给每页加一个 meta 值。比较常用的有以下几个：

name 属性

1). ＜meta name="Generator" contect=""＞用以说明生成工具（如 Microsoft FrontPage 4.0）等；

2). ＜meta name="KEYWords" contect=""＞向搜索引擎说明你的网页的关键词；

3). ＜meta name="DEscription" contect=""＞告诉搜索引擎你的站点的主要内容；

4).＜meta name="Author" contect="你的姓名"＞告诉搜索引擎你的站点的制作的作者；

5).＜meta name="Robots" contect=

"all|none|index|noindex|follow|nofollow"＞

其中的属性说明如下：

设定为 all：文件将被检索，且页面上的链接可以被查询；

设定为 none：文件将不被检索，且页面上的链接不可以被查询；

设定为 index：文件将被检索；

设定为 follow：页面上的链接可以被查询；

设定为 noindex：文件将不被检索，但页面上的链接可以被查询；

设定为 nofollow：文件将不被检索，页面上的链接可以被查询。

6).http-equiv 属性

a、＜meta http-equiv="Content-Type" contect="text/html";charset=gbk"＞

和 ＜meta http-equiv="Content-Language" contect="zh-CN"＞用以说明主页制作所使用的文字以及语言；

b、＜meta http-equiv="Refresh" contect="n;url=http://yourlink"＞定时让网页在指定的时间 n 内，跳转到页面 http;//yourlink；

c、＜meta http-equiv="Expires" contect="Mon,12 May 2001 00:20:00 GMT"＞可以用于设定网页的到期时间，一旦过期则必须到服务器上重新调用。需要注意的是必须使用 GMT 时间格式；

d、＜meta http-equiv="Pragma" contect="no-cache"＞是用于设定禁止浏览器从本地机的缓存中调阅页面内容，设定后一旦离开网页就无法从 Cache 中再调出；

e、＜meta http-equiv="set-cookie" contect="Mon,12 May 2001 00:20:00 GMT"＞cookie 设定，如果网页过期，存盘的 cookie 将被删除。需要注意的也是必须使用 GMT 时间格式；

f、＜meta http-equiv="Pics-label" contect=""＞网页等级评定，在 IE 的 internet 选项中有一项内容设置，可以防止浏览一些受限制的网站，而网站的限制级别就是通过 meta 属性来设置的；

g、＜meta http-equiv="windows-Target" contect="_top"＞强制页面在当前窗口中以独立页面显示，可以防止自己的网页被别人当作一个 frame 页调用；

h、＜meta http-equiv="Page-Enter" contect="revealTrans(duration=10,transtion=50)"＞和＜meta http-equiv="Page-Exit" contect="revealTrans(duration=20，transtion=6)"＞设定进入和离开页面时的特殊效果，这个功能即 FrontPage 中的“格式/网页过渡”，不过所加的页面不能够是一个 frame 页面。

45.请写出一个隐藏控件。

解答：<input type=”hidden” name=”userId” value=”10001”>

46.如何将 HTML 页面的标题设置为“数字天堂”。

解答：

```
<html>
<head><title>数字天堂</title></head>
<body>body 部分</body>
</html>
```

47.请写出 JavaScript 中常用的三种事件。

解答： onclick，onblur，onChange

48. 请写出一段 JavaScript 代码，要求页面有一个按钮，点击按钮弹出确认框。程序可以判断出用户点击的是“确认”还是“取消”。

解答：

```
<HTML>
<HEAD>
    <TITLE>click</TITLE>
    <Script >
        function validateForm()
        {
            if(confirm("你确认提交这个表单么？"))
            {
                alert(“确定”);
            }else{
           alert(“取消”);
        }
        }
    </Script>
</HEAD>
<BODY>
    <FORM NAME="TEST" action="FirstJS.htm">
        <INPUT TYPE="button" NAME="SUB" VALUE="提交"onClick="JavaScript:validateForm()">
    </FORM>
</BODY>
</HTML>
```

49.JavaScript 如何实现计时功能。

解 答 ： <script> window.setTimeout("window.location = 'http://www.csdn.net ';", 35000); </script>

50.JavaScript 如何定义数组。

解答： var arrTest=new Array();

51.JavaScript 能否操作 cookie 和 session?

解答：JavaScript 可以操作 cookie，但是不能操作 session

52.请写出如下正则表达式的含义；

com | cn | net

^http://www.d-heaven.com/$

^http://www.sina.com.cn/new/newid=\d+

解答：com | cn | net 表示匹配 com 或是 cn 或是 net 中一个

^http://www.d-heaven.com/$ 表示匹配：http://www.d-heaven.com/

^http://www.sina.com.cn/new/newid=\d+

表示匹配 http://www.sina.com.cn/new/newid=其后可以出现 1 到多个数字

53.请用正则表达式匹配出 QQ 号（假设 QQ 号码为 5—10 位）；

解答： ^ \d{5,10}$

54. String, StringBuffer StringBuilder 的区别。

解答：String 的长度是不可变的；

StringBuffer 的长度是可变的，如果你对字符串中的内容经常进行操作，特别是内容要修改时，那么使用 StringBuffer，如果最后需要 String，那么使用 StringBuffer 的 toString()方法；线程安全；

StringBuilder 是从 JDK 5 开始，为 StringBuffer 该类补充了一个单个线程使用的等价类；通常应该优先使用 StringBuilder 类，因为它支持所有相同的操作，但由于它不执行同步，所以速度更快。

55.请写出 5 种常见到的 runtime exception。

解答：

NullPointerException：当操作一个空引用时会出现此错误。

NumberFormatException：数据格式转换出现问题时出现此异常。

ClassCastException：强制类型转换类型不匹配时出现此异常。

ArrayIndexOutOfBoundsException：数组下标越界，当使用一个不存在的数组下标时出现此异常。

ArithmeticException：数学运行错误时出现此异常

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

解答：数组没有 length（）方法 它有 length 属性

String 有 length（）方法。

57.请写出一个单例模式。

解答：单例模式（Singleton pattern）:确保一个类只有一个实例，并提供一个全局的访问点

```
public class EagerSingleton
{
```

```
private static final EagerSingleton m_instance =
new EagerSingleton();
/**
* 私有的默认构造子
*/
private EagerSingleton() { }
/**
* 静态工厂方法
*/
public static EagerSingleton getInstance()
{
return m_instance;
}
}
```

58.在 java 中，List 是个接口，那实现 List 接口的类有哪些，有什么区别？

解答：ArrayList 是使用数组方式存储数据,此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，

LinkedList 使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

59.char 类型能否存储一个中文字符？为什么

解答：可以。一个 char 是两个字节，而一个中文也是两个字节。

60.Switch 选择语句能否作用在 String【字符串】上，也就是能否这么写：Switch(一个字符串变量)？

解答：不可以，只能处理 int,byte,short,char，（其实是只能处理 int,其它三种是可以 promotion 到 int 型）不能处理 Sring 。

61.关键字 final 分别修饰一个类，一个方法，一个变量，各起什么作用

解答：final 修饰类是不能被继承

fianl 修饰方法不能在子类中被覆盖

final 修饰变量，称为常量，初始化以后不能改变值。

62.启动一个线程是用 run()还是 start()？

解答：start（）。

63.是否可以继承 String 类

解答：不可以。因为 String 类有 final 修饰。

64.Java 异常处理中，try {}里有一个 return 语句，那么紧跟在这个 try 后的 finally {}里的 code 会不会被执行，什么时候被执行，在 return 前还是后?

解答：会执行，在 return 前执行。

65.EJB 包含哪 3 种 bean

解答：session bean（会话 bean）， entity bean（实体 bean）， message bean（消息 bean）

66.Class.forName（String className）这个方法的作用

解答：通过类的全名获得该类的类对象

67.你认为在表上建立索引可以提高数据库系统的效率吗，为什么？

解答：不一定

建立太多的索引将会影响更新和插入的速度，因为它需要同样更新每个索引文件。对于一个经常需要更新和插入的表格，就没有必要为一个很少使用的 where 子句单独建立索引了，对于比较小的表，排序的开销不会很大，也没有必要建立另外的索引。

68.hibernate 中的 java 对象有几种状态，其相互关系如何（区别和相互转换）。

解答：在 Hibernate 中，对象有三种状态：临时状态、持久状态和游离状态。

临时状态：当 new 一个实体对象后，这个对象处于临时状态，即这个对象只是一个保存临时数据的内存区域，如果没有变量引用这个对象，则会被 jre 垃圾回收机制回收。这个对象所保存的数据与数据库没有任何关系，除非通过 Session 的 save 或者 SaveOrUpdate 把临时对象与数据库关联，并把数据插入或者更新到数据库，这个对象才转换为持久对象；

持久状态:持久化对象的实例在数据库中有对应的记录，并拥有一个持久化表示（ID）。对持久化对象进行 delete 操作后，数据库中对应的记录将被删除，那么持久化对象与数据库记录不再存在对应关系，持久化对象变成临时状态。

持久化对象被修改变更后，不会马上同步到数据库，直到数据库事务提交。在同步之前，持久化对象是脏的（Dirty）。

游离状态：当 Session 进行了 Close、Clear 或者 evict 后，持久化对象虽然拥有持久化标识符和与数据库对应记录一致的值，但是因为会话已经消失，对象不在持久化管理之内，所以处于游离状态（也叫：脱管状态）。游离状态的对象与临时状态对象是十分相似的，只是它还含有持久化标识。

69.对 hibernate 的延迟加载如何理解，在实际应用中，延迟加载与 session 关闭的矛盾是如何处理

的？

解答：延迟加载就是并不是在读取的时候就把数据加载进来，而是等到使用时再加载。那么 Hibernate 是怎么知识用户在什么时候使用数据了呢？又是如何加载数据呢？其实很简单，它使用了代理机制。返回给用户的并不是实体本身，而是实体对象的代理。代理对象在用户调用 getter 方法时就会去数据库加载数据。但加载数据就需要数据库连接。而当我们把会话关闭时，数据库连接就同时关闭了。这种情况就叫做未初始化的关系。

延迟加载与 session 关闭的矛盾一般可以这样处理：

1)、关闭延迟加载特性。

操作起来比较简单，因为 hibernate 的延迟加载特性是在 hbm 配置里面可控制的。默认 lazy="true"，具体配置可以查看一下相关文档，就不详细叙述了。

但使用这个解决办法带来的隐患是十分大的。

首先，出现 no session or session was closed 就证明了您已经在使用外键关联表，如果去掉延迟加载的话，则表示每次查询的开销都会变得十分的大，如果关联表越多，后果也可以想象得到。所以不建议使用这个方法解决。

2)、在 session 关闭之前把我们想要查询的数据先获取了。

首先需要了解一下 session 什么时候关闭，也就是它的生命周期。通常情况下 hibernate 会在查询数据关闭 session，而使用 getHibernateTemplate().get 方法查询后会延迟关闭的时间。会在事务结束后才关闭。

使用拦截器(Interceptor)或过滤器(Filter)控制 session。

spring 为解决 hibernate 这一特性提供的解决方案，可以有效的控制 session 生命周期。

70.什么是 AOP 和 OOP，IOC 和 DI 有什么不同?

解答：

1）面向对象编程（Object Oriented Programming，OOP，面向对象程序设计）是一种计算机编程架构。AOP 是 OOP 的延续，是 Aspect Oriented Programming 的缩写，意思是面向方面编程。 将通用需求功能从不相关类之中分离出来；同时，能够使得很多类共享一个行为，一旦行为发生变化，不必修改很多类，只要修改这个行为就可以。AOP 就是这种实现分散关注的编程方法，它将“关注”封装在“方面”中

2)控制反转 IOC(Inversion of Control) 控制指的就是程序相关类之间的依赖关系.传统观念设计中,通常由调用者来创建被调用者的实例, 在 Spring 里,创建被调用者的工作不再由调用者来完成,而是由 Spring 容器完成，依赖关系被反转了，称为控制反转，目的是为了获得更好的扩展性和良好的可

维护性。依赖注入(Dependency injection)创建被调用者的工作由 Spring 容器完成，然后注入调用者，因此也称依赖注入。控制反转和依赖注入是同一个概念。

71.Struts1 中 actionform 和 action 属于 MVC 哪一层，为什么？

解答：actionform 和 action 属于 MVC 的 Model 层，Action 用来处理业务逻辑，actionform 保存用户表单数据以便于在不同页面间传递。而 MVC 中的 model 层就是业务逻辑层，该层用于实现具体的业务逻辑、状态维护及管理。

73.error 和 exception 有什么区别?

解答：

error 表示系统级的错误和程序不必处理的异常，是恢复不是不可能但很困难的情况下的一种严重问题；比如内存溢出，不可能指望程序能处理这样的情况；

exception 表示需要捕捉或者需要程序进行处理的异常，是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。

74.Log4J 是 Apache 组织的开源一个开源项目，通过 Log4J，可以指定日志信息输出的目的地，如 console、file 等。Log4J 采用日志级别机制，请按照输出级别由低到高的顺序写出日志输出级别。

解答：Log4J 分 为 OFF、FATAL、ERROR、WARN、INFO、DEBUG、ALL 或者您定义的级别。Log4j 建议只使用四个级别，优先级从高到低分别是 ERROR、WARN、INFO、DEBUG。通过在这里定义的级别，您可以控制到应用程序中相应级别的日志信息的开关。比如在这里定义了 INFO 级别， 则应用程序中所有 DEBUG 级别的日志信息将不被打印出来。

75.说出几个与 spring 同类型的开源框架，说出几个与 hibernate 同类型的开源框架,说出几个与 struts 同类型的开源框架

解答：

1）与 spring 同类型的开源框架：JUIDE、EJB3.0、picoContainer

2）与 hibernate 同类型的开源框架 ：ibatis，jdo，JPA

3）几个与 struts 同类型的开源框架：webwork，tapestry，JSF

76、Struts2 包含哪些标签？

解答：

A：

<s:a href=""></s:a>-----超链接，类似于 html 里的<a>

<s:action name=""></s:action>-----执行一个 view 里面的一个 action

<s:actionerror/>-----如果 action 的 errors 有值那么显示出来

<s:actionmessage/>-----如果 action 的 message 有值那么显示出来

<s:append></s:append>-----添加一个值到 list，类似于 list.add();

<s:autocompleter></s:autocompleter>-----自动完成<s:combobox>标签的内容，这个是 ajax

B：

<s:bean name=""></s:bean>-----类似于 struts1.x 中的，JavaBean 的值

C：

<s:checkbox></s:checkbox>-----复选框

<s:checkboxlist list=""></s:checkboxlist>-----多选框

<s:combobox list=""></s:combobox>-----下拉框

<s:component></s:component>-----图像符号

D：

<s:date/>-----获取日期格式

<s:datetimepicker></s:datetimepicker>-----日期输入框

<s:debug></s:debug>-----显示错误信息

<s:div></s:div>-----表示一个块，类似于 html 的

<s:doubleselect list="" doubleName="" doubleList=""></s:doubleselect>-----双下拉框

E：

<s:if test=""></s:if>

<s:elseif test=""></s:elseif>

<s:else></s:else>-----这 3 个标签一起使用，表示条件判断

F：

<s:fielderror></s:fielderror>-----显示文件错误信息

<s:file></s:file>-----文件上传

<s:form action=""></s:form>-----获取相应 form 的值

G：

<s:generator separator="" val=""></s:generator>----和<s:iterator>标签一起使用

H：

<s:head/>-----在<head></head>里使用，表示头文件结束

<s:hidden></s:hidden>-----隐藏值

I：

<s:i18n name=""></s:i18n>-----加载资源包到值堆栈

<s:include value=""></s:include>-----包含一个输出，servlet 或 jsp 页面

<s:inputtransferselect list=""></s:inputtransferselect>-----获取 form 的一个输入

<s:iterator></s:iterator>-----用于遍历集合

L：

<s:label></s:label>-----只读的标签

M：

<s:merge></s:merge>-----合并遍历集合出来的值

O：

<s:optgroup></s:optgroup>-----获取标签组

<s:optiontransferselect doubleList="" list="" doubleName=""></s:optiontransferselect>-----左右选择框

P：

<s:param></s:param>-----为其他标签提供参数

<s:password></s:password>-----密码输入框

<s:property/>-----得到'value'的属性

<s:push value=""></s:push>-----value 的值 push 到栈中,从而使 property 标签的能够获取 value 的属性

R：

<s:radio list=""></s:radio>-----单选按钮

<s:reset></s:reset>-----重置按钮

S：

<s:select list=""></s:select>-----单选框

<s:set name=""></s:set>-----赋予变量一个特定范围内的值

<s:sort comparator=""></s:sort>-----通过属性给 list 分类

<s:submit></s:submit>-----提交按钮

<s:subset></s:subset>-----为遍历集合输出子集

T：

<s:tabbedPanel id=""></s:tabbedPanel>-----表格框

<s:table></s:table>-----表格

<s:text name=""></s:text>-----I18n 文本信息

<s:textarea></s:textarea>-----文本域输入框

<s:textfield></s:textfield>-----文本输入框

<s:token></s:token>-----拦截器

<s:tree></s:tree>-----树

<s:treenode label=""></s:treenode>-----树的结构

U：

<s:updownselect list=""></s:updownselect>-----多选择框

<s:url></s:url>-----创建 url

77、struts2 中，OGNL 访问值栈的时候查找的顺序是什么？请排序：模型对象、临时对象、固定名称的对象、Action 对象

解答：struts2 的值栈排列顺序为：1）.临时对象；2）.模型对象；3）.Action 对象；4）.固定名称对象（如#application, #session, #request 等对象）。

78、struts2 中，Action 通过什么方式获得用户从页面输入的数据，又是通过什么方式把其自身的数据传给视图的？

解答：

1）可以直接通过与表单元素相同名称的数据成员（需要存在符合命名规范 set 和 get 方法）获取页面表单数据。

2）会把处理好的数据成员放入值栈中，到页面可以使用 struts2 标签取值就可以了。

79.常用的设计模式有哪些？说明工厂模式。

解答：Java 中的 23 种设计模式：

Factory（工厂模式）， Builder（建造模式）, Factory Method（工厂方法模式），

Prototype（原始模型模式）, Singleton（单例模式）,Facade（门面模式），

Adapter（适配器模式）, Bridge（桥梁模式）,Composite（合成模式），

Decorator（装饰模式）, Flyweight（享元模式）, Proxy（代理模式），

Command（命令模式）, Interpreter（解释器模式）, Visitor（访问者模式），

Iterator（迭代子模式）, Mediator（调停者模式）, Memento（备忘录模式），

Observer（观察者模式）, State（状态模式）, Strategy（策略模式），

Template Method（模板方法模式）， Chain Of Responsibleity（责任链模式）

工厂模式：工厂模式是一种经常被使用到的模式，根据工厂模式实现的类可以根据提供的数据生成一

组类中某一个类的实例，通常这一组类有一个公共的抽象父类并且实现了相同的方法，但是这些方法针对不同的数据进行了不同的操作。首先需要定义一个基类，该类的子类通过不同的方法实现了基类中的方法。然后需要定义一个工厂类，工厂类可以根据条件生成不同的子类实例。当得到子类的实例后，开发人员可以调用基类中的方法而不必考虑到底返回的是哪一个子类的实例。

80.什么是数据库的参照完整性？

解答：数据库的参照完整性是指表与表之间的一种对应关系，通常情况下可以通过设置两表之间的主键、外键关系，或者编写两表的触发器来实现。

有对应参照完整性的两张表格，在对他们进行数据插入、更新、删除的过程中，系统都会将被修改表格与另一张对应表格进行对照，从而阻止一些不正确的数据的操作。

81.如何优化数据库，如何提高数据库的性能?

解答：

1） 硬件调整性能

最有可能影响性能的是磁盘和网络吞吐量,解决办法扩大虚拟内存，并保证有足够可以扩充的空间；把数据库服务器上的不必要服务关闭掉；把数据库服务器和主域服务器分开；把 SQL 数据库服务器的吞吐量调为最大；在具有一个以上处理器的机器上运行 SQL。

2）调整数据库

若对该表的查询频率比较高,则建立索引;建立索引时,想尽对该表的所有查询搜索操作， 按照 where 选择条件建立索引，尽量为整型键建立为有且只有一个簇集索引，数据在物理上按顺序在数据页上，缩短查找范围，为在查询经常使用的全部列建立非簇集索引，能最大地覆盖查询；但是索引不可太多，执行 UPDATE DELETE INSERT 语句需要用于维护这些索引的开销量急剧增加；避免在索引中有太多的索引键；避免使用大型数据类型的列为索引；保证每个索引键值有少数行。

3）使用存储过程

应用程序的实现过程中，能够采用存储过程实现的对数据库的操作尽量通过存储过程来实现，因为存储过程是存放在数据库服务器上的一次性被设计、编码、测试，并被再次使用，需要执行该任务的应用可以简单地执行存储过程，并且只返回结果集或者数值，这样不仅可以使程序模块化，同时提高响应速度，减少网络流量，并且通过输入参数接受输入，使得在应用中完成逻辑的一致性实现。

4）应用程序结构和算法

建立查询条件索引仅仅是提高速度的前提条件，响应速度的提高还依赖于对索引的使用。因为人们在使用 SQL 时往往会陷入一个误区，即太关注于所得的结果是否正确，特别是对数据量不是特别大的数据库操作时，是否建立索引和使用索引的好坏对程序的响应速度并不大，因此程序员在书写程序时就

忽略了不同的实现方法之间可能存在的性能差异，这种性能差异在数据量特别大时或者大型的或是复杂的数据库环境中（如联机事务处理 OLTP 或决策支持系统 DSS）中表现得尤为明显。在工作实践中发现，不良的 SQL 往往来自于不恰当的索引设计、不充份的连接条件和不可优化的 where 子句。在对它们进行适当的优化后，其运行速度有了明显地提高！

82.JS 中的三种弹出式消息提醒(警告窗口、确认窗口、信息输入窗口)的命令是什么？

解答：alert　confirm　prompt

83.描述 JSP 和 Servlet 的区别、共同点、各自应用的范围

解答：JSP 在本质上就是 SERVLET,但是两者的创建方式不一样.Servlet 完全是 JAVA 程序代码构成，擅长于流程控制和事务处理，通过 Servlet 来生成动态网页很不直观.JSP 由 HTML 代码和 JSP 标签构成，可以方便地编写动态网页.因此在实际应用中采用 Servlet 来控制业务流程，而采用 JSP 来生成动态网页.

84.在 java 语言中 int 和 Integer 有什么区别

解答：int 是基本数据类型，Integer 是 int 的包装类，属于引用类型

85.计算下列结果：

25&3 =

25|3=

解答：25 的二进制是 11001 ， 3 的二进制是 00011， 按位与的结果是：00001 ，按位或的结果是：11010，因此转成 10 进制分别为：1 和 27

85.如何获得 This is first layer 中的值？

解答：

```
<script>
   var div1=Document.getElementById(“div1”);
   alert(div1.innerHTML)
</script>
```

86.JDK1.5 中支持的 for 循环的语法

解答：

```
for(type element : array) {
  System.out.println(element)
}
```

type　　集合(不仅仅指 Collection，也包含普通的数组)中元素的数据类型

element　　遍历到的元素

array　　集合对象本身(当然不只是 Collection)

87.简述 synchronized 和 java.util.concurrent.locks.Lock 的异同?

解答：

主要相同点：Lock 能完成 synchronized 所实现的所有功能

主要不同点：Lock 有比 synchronized 更精确的线程语义和更好的性能。synchronized 会自动释放锁，而 Lock 一定要求程序员手工释放，并且必须在 finally 从句中释放。

88.如何格式化日期

解答：

Date now=new Date();

SimpleDateFormat sdf=new SimpleDateFormat(“yyyy-MM-dd hh:mm:ss”);

String formatNow=sdf.format(now);

变量 formatNow 就是格式化好的日期。

89.将字符“12345”转换成 long 型

解答：　String s="12345";

　　long num=Long.valueOf(s).longValue();

90.struts 中如何实现国际化，涉及哪些文件?

解答：“国际化”是指一个应用程序在运行时能够根据客户端请求所来自的国家/地区、语言的不同而显示不同的用户界面。Struts 框架通过使用<bean:message>标记，以及使用 java.util 数据包中定义的 Locale 和 ResourceBundle 类来支持国际化。java.text.MessageFormat 类定义的技术可以支持消息的格式。利用此功能，开发人员不需了解这些类的细节就可进行国际化和设置消息的格式。会涉及到资源文件，不需了解这些类的细节就可进行国际化和设置消息的格式。会涉及到资源文件，struts-config.xml 配置文件,web.xml 配置文件。

91.例举在诊断 Oracle 性能问题时，常用的工具、方法

解答：

1）简单一点的可以用 toad 及 dbartisan 这样的工具。

2）纯做性能监测，比较出色的有 spolight 和 emc 的 I3，这两个软件都比较贵。

3）一些网管系统这方面也不错，如 hp 的 openview。不过定制起来不太容易，且很贵。

4）不用花钱又好用的就是 Oracle 的 statpack 了。

5）再有就是自己写脚本了，大多数有经验的 DBA 都喜欢这样的方式。优点是实用灵活。缺点是如果

想出一个性能变化的曲线图等图表，且做的美观就需要些 delphi,c++builder 或是 pb 的开发功底了。

92.Oracle 启动中，startup nomount、 startup mount 有什么差别?

解答： startup nomount：启动实例，读取参数文件，分配内存空间，启动后台进程，打开跟踪文件和报警文件。startup mount：装载数据库，打开控制文件。nomount 方式下还没有读取控制文件，该选项用于在数据库的控制文件全部损坏，需要重新创建数据库控制文件或创建一个新的数据库时使用。mount 选项下并没有打开数据文件，该选项可以用来修改数据库的运行模式或进行数据库恢复。

93.Oracle 启动中，spfile.ora、init<SID>.ora、spfile<SID>.ora 这三个文件正确的先后顺序是什么?

解答：启动数据库，使用 startup 命令，Oralce 将会按照以下顺序在缺省目录中搜索参数文件：

spfile<SID>.ora , spfile.ora ,init<SID>.ora a

94.说明 Oracle 数据库逻辑备份和物理备份的方式。

解答：Oracle 备份包括逻辑备份和物理备份。

1). 逻辑备份

数据库的逻辑备份包含读一个数据库记录集和将记录集写入文件。

a.输出（Export）输出可以是整个数据库、指定用户或指定表。

b.输入（Import）输入将输出建立的二进制转储文件读入并执行其命令。

2). 物理备份

物理备份包含拷贝构成数据库的文件而不管其逻辑内容。

Oracle 支持两种不同类型的物理文件备份：脱机备份(offline backup)和联机备份(online backup)。

a.脱机备份: 脱机备份用在当数据库已正常关闭，数据库处于"offline"时，要备份下列文件：

所有数据文件

所有控制文件

所有联机日志

init.ora（可选的）

b 联机备份:联机备份可用来备份任何运作在 ARCHIVELOG 方式下的数据库。在这种方式下，联机日志被归档，在数据库内部建立一个所有作业的完整记录。联机备份过程具备强有力的功能。第一，提供了完全的时间点（point-in-time）恢复。第二，在文件系统备份时允许数据库保持打开状态。

95.有 2 个类 Cat 及 WhiteCat，代码如下：

```
public class Cat {
    protected static String color = "random";
```

```
    public Cat() {
    }
    public void showCatColor() {
        System.out.println("Cat:" + color);
    }
    public static void  showColor() {
        System.out.println("Cat:" + color);
    }
}
public class WhiteCat extends Cat {
    protected static String color = "white";
    public WhiteCat() {
        super();
    }
    public void showCatColor() {
        System.out.println("WhiteCat:" + color);
    }
    public static void showColor() {
        System.out.println("WhiteCat:" + color);
    }
}
```

请分析下面各段程序的运行结果

```
A. WhiteCat whiteCat = new WhiteCat();
Cat cat = whiteCat;
cat.showColor();
cat.showCatColor();
B. Cat cat = new Cat();
WhiteCat whiteCat = (WhiteCat) cat;
cat.showColor();
cat.showCatColor();
```

C. Cat cat = new WhiteCat();

WhiteCat whiteCat = (WhiteCat) cat;

cat.showColor();

cat.showCatColor();

解答：A 段执行的结果是：

Cat:random

WhiteCat:white

B 段执行的结果是：

会抛出 java.lang.ClassCastException 异常

C 段执行的结果是：

Cat:random

WhiteCat:white

96、说说下面语句是否有错误，或可能出现的缺陷，并指出错误，或缺陷在哪里？

```
   public class MyFile implements Runnable{
         public void run(){
             while (true){
             try{
                 FileReader fr=new FileReader(new File("a.txt")) ;
                 String line=fr.readLine();
                 System.out.println(line);
             }catch(IOException err) {
             }
         Sleep(1000);    }
}
```

解答： 1.fr.readLine()没有这个方法

2.Sleep(1000)需要用 Thread.sleep(1000);

97、判断下列语句是否正确，如果有错误，请指出错误所在？

List<Short> a = new ArrayList<Short>();

a.add(5);

解答：错误,默认封装 int 类型。

98、判断下列语句是否正确，如果有错误，请指出错误所在？

```
void foo(final int []arg){
    if (arg.length > 1)
      arg[0] = 5;
}
```

解答：正确

99、判断下列语句是否正确，如果有错误，请指出错误所在？

```
interface A{
   int add(final A a);
}
class B implements A{
    long add(final A a){
       return this.hashCode() + a.hashCode();
    }
}
```

解答：返回值不是 long 类型

100、指出下面程序的运行结果：

```
  class A{
        static{
            System.out.print("a");
        }
        public A (){
            System.out.print("x");
        }
}
class B extends A{
static{
        System.out.print("b");
        }
        public B (){
```

```
        System.out.print("y");
}
}
public class Test{
        public static void main(String[] args){
            A ab = new B ();
            ab = new B ();
        }
}
```

解答：abxyxy

101、下列代码的输出结果是什么？

```
public class MyFor {
    public static void main (String argv[]){
        int i; int j;
        outer:for(i=1;i<3;i++)
        inner:for(j=1;j<3;j++){
            if (j==2) continue outer;
                System.out .println("Value for i="+i+" Value for j=" +j);
        }
    }
}
```

解答：Value for i=1 Value for j=1

Value for i=2 Value for j=1

102、查看下面的代码，写出可以使程序正常执行的修改方法

```
1.public class MyClass {
2.static String s1;
3.  String s2;
4.  public static void main(String args[]) {
5.      String s3;
6.      System.out.println("s1 =" + s1);
```

```
7.        System.out.println("s2 =" + s2);
8.        System.out.println("s3 =" + s3);
9.    }
10.}
```

解答：删除第 8 行或者将第 6 行改为 String s3 = "";

103、为了显示 myStr = 23 这样的结果，写出在控制台输入的命令

```
 public class MyClass {
    public static void main(String args[]) {
        String s1 = args[0];
        String s2 = args[1];
        String myStr = args[2];
        System.out.printin("myStr =" + s2 + myStr);
    }
}
```

解答：java MyClass 1 2 3 4

104、写出下面代码的执行结果

```
public class MyClass {
    static void aMethod(StringBuffer sf1, StringBuffer sf2) {
        sf1.append(sf2);
        sf2 = sf1;
    }
    public static void main(String[] args){
        StringBuffer sf1 = new StringBuffer("A");
        StringBuffer sf2 = new StringBuffer("B");
        aMethod(sf1,sf2);
        System.out .println(sf1+ ":"+sf2);
    }
}
```

解答：AB:B

105、第 3 行中生成的 object 在第几行执行后成为 garbage collection 的对象？

```
1.public class MyClass {
2.  public StringBuffer aMethod() {
3.       StringBuffer sf = new StringBuffer("Hello");
4.       StringBuffer[] sf_arr = new StringBuffer[1];
5.       sf_arr[0] = sf;
6.       sf = null;
7.       sf_arr[0] = null;
8.       return sf;
9.  }
10.}
```

解答：第 7 行

106、写出执行下面的代码后的结果

```
  public class MyClass {
    public static void main(String args[]) {
        java.util.Vector v1 = new java.util.Vector();
        v1.addElement("Hello");
        v1.addElement(new Float(3.14f));
        v1.addElement(10);
        System.out.println(v1.elementAt(0) + ":" + v1.elementAt(1) + ":"+ v1.elementAt(2));
    }
}
```

解答：Hello : 3.14 : 10

107、写出执行下面代码后的正确结果

```
interface MyDB {
public void getConnection();
}
class MyDBDriver implements MyDB {
        public void getConnection() {
            System.out.println("getConnection()");
        }
```

```
}
public class MyClass {
        public static void aMethod(MyDB db) {
            db.getConnection();
        }
        public static void main(String args[]) {
            MyDBDriver db_driver = new MyDBDriver();
            aMethod(db_driver);
        }
}
```

解答：getConnection()

108、下列程序运行的结果是

```
  class A {
    class Dog {
        private String name;
        private int age;
        private int step;
        Dog(String s, int a) {
            name = s;
            age = a;
            step = 0;
        }
        public void run(Dog fast) {
            fast.step++;
        }
    }
    public static void main(String args[]) {
        A a = new A();
        Dog d = a.new Dog("Tom", 3);
        d.step = 25;
```

```
        d.run(d);
        System.out.println(d.step);
    }
}
```

解答：26

109、请看下列程序，运行结果是

```
class Super{
int i=10;
Super(){
print();
i=20;
}
void print(){
System.out.print(i);
}
}
public class Sub extends Super{
int j=30;
Sub(){
print();
j=40;
}
void print(){
System.out.print(j);
}
public static void main(String[] args){
System.out.print(new Sub().j);
}
}
```

解答：03040

110、getSomething ()执行时发生 IllegalArgumentException 会出现什么样的结果？

```
void makeConnection(String url)  {
    try {
        getSomething();
    }catch(NullPointerException e) {
        System.out.printin("Invalid URL") ;
        return;
    }catch(Exception e) {
        System.out.println("Exception");
    }
}
```

解答：Exception

111.Tomcat 服务器的默认端口是多少？怎样修改 tomcat 的端口？

解答：默认端口为 8080，可以通过 service.xml 的 Connector 元素的 port 属性来修改端口。

112.多线程有几种实现方法,都是什么?同步的方法有几种，都是什么？

解答：多线程有两种实现方法：继承 Thread 类或者实现 Runnable 接口。

实现同步也有两种方法：一种是同步方法，另一种是同步代码块。

同步方法是在方法返回类型前面加上 synchronized 关键字

同步代码块是 synchronized (这里写需要同步的对象){...}

113.谈一下聚簇索引和非聚簇索引的区别以及各自的优缺点。

解答：

聚集索引,表中存储的数据按照索引的顺序存储,检索效率比普通索引高,但对数据新增/修改/删除的影响比较大

非聚集索引,不影响表中的数据存储顺序,检索效率比聚集索引低,对数据新增/修改/删除的影响很小

114. 死锁的必要条件？怎么克服？

解答：产生死锁的四个必要条件：

互斥条件：一个资源每次只能被一个进程使用。

请求与保持条件：一个进程因请求资源而阻塞时，对已获得的资源保持不放。

不剥夺条件:进程已获得的资源，在末使用完之前，不能强行剥夺。

循环等待条件:若干进程之间形成一种头尾相接的循环等待资源关系。

这四个条件是死锁的必要条件，只要系统发生死锁，这些条件必然成立，而只要上述条件之一不满足，就不会发生死锁。

死锁的解决方法:

a 撤消陷于死锁的全部进程；

b 逐个撤消陷于死锁的进程，直到死锁不存在；

c 从陷于死锁的进程中逐个强迫放弃所占用的资源，直至死锁消失。

d 从另外一些进程那里强行剥夺足够数量的资源分配给死锁进程，以解除死锁状态

115.描述重做与回滚的认识;

解答：重做日志生成日志文件,是为将来恢复数据库使用的.

回滚段保存未提交数据,是为支持事务而起作用的。

116.索引组织表,聚簇表的用途;

解答：

索引组织表：数据按主码存储和排序，同索引结构一样，不过数据直接存储于主码后面。适用于信息.检索、空间和 OLAP 程序。索引组织表的适用情况：

a.代码查找表。

b.经常通过主码访问的表。

c.构建自己的索引结构。

d.加强数据的共同定位，要数据按特定顺序物理存储。

e.经常用 between…and…对主码或唯一码进行查询。数据物理上分类查询。如一张订单表，按日期装载数据，想查单个客户不同时期的订货和统计情况。

索引聚簇表：索引聚簇表是表相关的表共享同一数据块中的相同列，并把相关数据存储中同一个数据块上。创建索引聚簇表中最重要的是对 SIZE 参数有很好的估量，否则聚簇将会降低空间利用，降低效率。

使用索引聚簇表的注意点：

a 如果表中数据有大量 DML 操作的话，那么聚簇将不适用，因为会消极地影响到 DML 性能。

b 聚簇中，全表扫描将受到影响。这是因为将扫描聚簇中不同表的数据，额外增加很多无用的数据。

c 如果经常 TRUNCATE 表和装载表的话，聚簇将不适用。聚簇中的表无法被 TRUNCATE 的，这是因为每个块中不只是存储一张表的数据。

```
SQL> truncate table emp;
truncate table emp
```

*

ERROR at line 1:

ORA-03292: Table to be truncated is part of a cluster

d 如果大部分是读取操作，且通过聚簇码索引或聚簇表中其他索引来读取的话，聚簇将会比较适用。

117.消耗资源的 SQL 的定位方法;

解答：select sql_text

from v$sql

where disk_reads > 1000 or (executions > 0 and buffer_gets/executions > 30000);

SELECT * FROM (

SELECT sql_text,buffer_gets,disk_reads FROM v$sql

ORDER BY buffer_gets,disk_reads DESC)

WHERE ROWNUM<=10;

118.对触发器的认识;

解答：触发器是表上的程序,主要提供数据添加、修改与删除后的程序处理方法，可以用来检查数据及进行数据更新，也可以分担一些前端应用程序撰写的逻辑规则。用场景: 触发器可以查询其他表，而且可以包含复杂的 SQL 语句。它们主要用于强制复杂的业务规则或要求。

触发器的主要应用场合概括起来讲有以下几种：

1).当向一张表中添加或删除记录时，需要在相关表中进行同步操作。比如，当为应用系统添加一个系统用户时，需要同时向权限表中添加该用户的缺省权限，此时就编写系统用户表的触发器在添加记录动作时触发。

2).当表上某列数据的值与其他表中的数据有联系时。比如，当某客户进行欠款消费，可以在生成订单时通过设计触发器判断该客户的累计欠款是否超出了最大限度。

3).当需要对某张表进行跟踪时。比如，当人事表中有人离职时，第一时间通知或更改相关表的值。

119.对 ORA-01555 错误的认识;

解答：ORA-01555 错误产生的原因：一致性读(Consistent Get)和延迟块清除(Delayed Block Cleanout)。

120.将 ORACLE 数据库更改为归档模式；写出步骤

解答：具体步骤如下：

1)，以 exp 方式在线备份数据库到指定位置;

2)，观察当前数据库是以服务器参数文件(spfile)方式启动还是以参数文件(pfile)方式启动：

SQL> show parameter spfile;

NAME	TYPE	VALUE
spfile	string	/home/db/oracle/10g/dbs/spfile XXXX.ora

value 后有内容，说明数据库以服务器参数文件方式启动，这里的 spfile 文件对应的裸设备为 /dev/vgdata/rspfile（通过查看/home/db/oracle/10g/dbs/initSID.ora 文件内容获得）；

3)，关闭所有实例（shutdown immediate）；

4)，任意选取一个实例，创建参数文件到指定路径：

SQL>create pfile='/home/db/oracle/pfile.ora' from spfile;

5)，修改 pfile.ora 文件，添加参数 cluster_database=false；

6)，以修改之后的参数文件按 nomount 方式启动数据库：

SQL>startup nomount pfile='/home/db/oracle/pfile.ora';

7)，使数据库以 exclusive 方式启动：

SQL>alter database mount exclusive;

8)，改变归档模式：

SQL>alter database archivelog;

9)，将 pfile 参数中的 cluster_database 重新更改为“true”；

10)，更新服务器参数文件：

SQL>create spfile from pfile='/home/db/oracle/pfile.ora';

11)，关闭数据库实例；

SQL>shutdowm immediate;

12)，分别在两个节点上启动数据库：

SQL>startup;

13)，在两个节点上分别检查归档模式是否更改成功：

SQL>archive log list;

Database log mode	Archive Mode
Automatic archival	Enabled
Archive destination	/home/db/oracle/10g/dbs/arch

Oldest online log sequence 489

Next log sequence to archive 491

Current log sequence 491

完成。

121.把表 A 从表空间 TSP1 更改成表空间 TSP2 的方法

解答：alter table A move tablespace TSP2

122.删除表的列；

解答：alert table 表名 drop 列名

123.删除表空间的数据文件？

解答：用 SQLPLUS 命令: drop tablespace tablename including contents

就可以把所有相关的数据删除

124.如何用 ALTER 命令把表数据加到缓存表里，清除呢？

解答：添加：alert table 表名 cache

清除：alert table 表名 nocache

125. 数据的复制实现办法？

解答：数据复制，就是将数据库中的数据拷贝到另外一个或多个不同的物理站点上，从而保持源数据库与目标数据库中指定数据的一致性。

数据复制的实现方法：在具体的实现之前，首先要做好设计与规划。这就需要细致分析具体的业务情况，设计出一套能够满足业务需要的方案。通常在设计过程中，需要确定出要建立的数据库站点，各站点的类型，需要复制的数据对象，以及同步方式、冲突解决方案等内容。

数据复制的实现主要包括以下几步：(1)创建复制站点。(2)创建组对象。(3)配置冲突解决方案。

126.建立 ORACLE 数据库后，系统自带的两个用户是什么，相应的密码是什么？他们的身份是什么？

解答： 用户 密码 身份

System manager SYSDBA

Sys change_on_instal SYSDPER

127.在建立 ORACLE 数据库时，选择“事务处理”模板与选用“数据仓库”模板的

区别是什么？

解答：事物处理型:经常反馈给客户信息,处理大容量或超大容量的数据.

数据仓库型:主要频繁处理小型数据库,只是进行一些查询等操作.

128、在使用 OMS 之前需要建立资料档案库。在建立资料档案库的时候，在“为资料档案库选择数据

库”选项卡中输入的服务名称应该如何写。(比如你要连接 192.168.1.2 机器上的 SIST 数据库)。

解答：192.168.1.2:1521:SIST

129、登陆 OMS 所使用的用户名和密码分别为什么？如果在登陆时，系统提示找不到服务，你应该如何处理？

解答：用户名：SYSMAN,

密码：OEM_TEMP

问题出现在服务器没有启动，只能通过手动的方法在 Windows 管理工具下的策略中将服务器启动。

130、请说出两种以上扩大数据库的方式

解答 1) 修改现有表空间的大小；

2)向表空间插入一个新的数据文件.

131 在 ORACLE 中的物理文件包含哪四种？

解答：1)数据文件 扩展名为*.dbf

2)控制文件 扩展名为*.rtl

3)配置文件 扩展名为*.ora

4)日志文件 扩展名为*.dbf

132. 日志文件（记录文件）有哪几种，分别介绍他们的工作模式。

解答: 归档日志：当日志写满，完成一次循环之前建立一个副本。这样数据库就可以在损坏中得到恢复。恢复的过程相当于把建库的所有动作重新做一次。最安全的数据库工作方式，占用空间也最大

非归档日志：日志写满后，直接覆盖，它只是部分地记录数据库操作，所以恢复能力有限。

133、简要写出在 system 方案中建立序列 xl 的步骤。

解答：

```
create squence system.xl
start whith 1
increment by 1
minvalue 1
nomaxvalue
nocycle
nocache
order;
```

134、写出 SQL 语句，向表中插入一条记录，其中 ID 字段的值来自序列 XL。

解答：

```
Insert into system.table1 values (xl.nextval,’ tom’,21,’ 男’,2000);
```

135、 写出一个匿名的 SQL 程序块，完成如下任务：向表中插入 3000 条记录，在 salary 字段中有 500 条记录的值为 1000，500 条记录的值为 1200，1000 条记录的值为 1500，1000 条记录的值为 1800

Id 字段的值来自序列 xl,其他字段的值任意.

解答：

```
declare
x number:=0;
begin
for x in 1..3000 loop
If(x<=500) then
Insert into system.test values(x1.nextval,'jim',24,'m',1000);
elsif((x>500)and (x<1001)) then
Insert into system.test values(system.xl1.nextval,'jim',24,'m',1200);
Elsif((x>1000)and (x<2001)) then
Insert into system.test values(system.xl1.nextval,'jim',24,'m',1500);
Else
Insert into system.test values(system.xl1.nextval,'jim',24,'m',1800);
end if;
end loop;
end;
```

136、写出一个存储过程，这个存储过程的作用是修改特定 id 编号的记录，将该条记录的 salary 字段的值加上 500；

解答：

```
Create or replace procedure system.update_age
(vid in number) is
Begin
Update table1 set salary=salary+500 where id=vid;
End;
```

137、用 SQL 语句创建个一个视图，这个视图用来显示 ID>1000 的记录；

解答：create view st as select * from table1 where id>1000;

138、创建一个 system 方案中的函数 fn1，函数作用为：将指定 ID 号的记录中的 salary 字段值乘以 1.05。

解答：

```
Create or replace function system.fn1(salary1 system.table1 salary &type)
Return number as
V1 numbre:=1.05;
V2 numbre;
Begin
V2=v1*salary1;
Return v2;
End ;
```

139.解释冷备份和热备份的不同点以及各自的优点

解答：热备份针对归档模式的数据库，在数据库仍旧处于工作状态时进行备份。而冷备份指在数据库关闭后，进行备份，适用于所有模式的数据库。热备份的优点在于当备份时，数据库仍旧可以被使用并且可以将数据库恢复到任意一个时间点。冷备份的优点在于它的备份和恢复操作相当简单，并且由于冷备份的数据库可以工作在非归档模式下,数据库性能会比归档模式稍好。（因为不必将 archive log 写入硬盘）

140. 你必须利用备份恢复数据库，但是你没有控制文件，该如何解决问题呢？

解答：重建控制文件，用带 backup control file 子句的 recover 命令恢复数据库。

141. 如何转换 init.ora 到 spfile?

解答：使用 create spfile from pfile 命令

142. 解释 data block , extent 和 segment 的区别（这里建议用英文术语）

解答：data block 是数据库中最小的逻辑存储单元。当数据库的对象需要更多的物理存储空间时，连续的 data block 就组成了 extent . 一个数据库对象拥有的所有 extents 被称为该对象的 segment.

143. 给出两个检查表结构的方法

解答：1、DESCRIBE 命令

2、DBMS_METADATA.GET_DDL 包

144. 怎样查看数据库引擎的报错

解答：alert log.

145. 使用索引的理由

解答：快速访问表中的 data block

146. 给出在 STAR SCHEMA 中的两种表及它们分别含有的数据

解答：Fact tables 和 dimension tables. fact table 包含大量的主要的信息而 dimension tables 存放对 fact table 某些属性描述的信息

147. FACT Table 上需要建立何种索引？

解答：位图索引（bitmap index）

148. 给出两种相关约束?

解答：主键和外键

149. 如何在不影响子表的前提下，重建一个母表

解答：子表的外键强制失效，重建母表，激活外键

150. 如何建立一个备份控制文件？

解答：Alter database backup control file to trace.

151. 给出数据库正常启动所经历的几种状态 ？

解答：

STARTUP NOMOUNT - 数据库实例启动

STARTUP MOUNT - 数据库装载

STARTUP OPEN - 数据库打开

152. 哪个 column 可以用来区别 V$视图和 GV$视图?

解答： INST_ID 指明集群环境中具体的某个 instance 。

153. 如何生成 explain plan?

解答：

运行 utlxplan.sql. 建立 plan 表针对特定 SQL 语句,使用 explain plan set statement_id = 'tst1' into plan_table 运行 utlxplp.sql 或 utlxpls.sql 察看 explain plan

154. 如何增加 buffer cache 的命中率？

解答：在数据库较繁忙时，适用 buffer cache advisory 工具，查询 v$db_cache_advice . 如果有必要更改，可以使用 alter system set db_cache_size 命令

155. 解释$ORACLE_HOME 和$ORACLE_BASE 的区别？

解答：ORACLE_BASE 是 oracle 的根目录，ORACLE_HOME 是 oracle 产品的目录

156. 如何判断数据库的时区？

解答：SELECT DBTIMEZONE FROM DUAL

157. 解释 GLOBAL_NAMES 设为 TRUE 的用途

解答：GLOBAL_NAMES 指明连接数据库的方式。如果这个参数设置为 TRUE,在建立数据库链接时就必须用相同的名字连结远程数据库

158.如何加密 PL/SQL 程序？

解答：WRAP

159. 解释 TABLE Function 的用途

解答：TABLE Function 是通过 PL/SQL 逻辑返回一组纪录，用于普通的表/视图。他们也用于 pipeline 和 ETL(ETL，Extraction-Transformation-Loading 的缩写，中文名称为数据提取、转换和加载)过程。

160. 举出 3 种可以收集 three advisory statistics

解答：Buffer Cache Advice, Segment Level Statistics, Timed Statistics

161. Audit trace 存放在哪个 oracle 目录结构中?

解答：unix $ORACLE_HOME/rdbms/audit

Windows the event viewer

162. 解释 materialized views 的作用

解答：Materialized views 用于减少那些汇总，集合和分组的信息的集合数量。它们通常适合于数据仓库和 DSS 系统

163. 当用户进程出错，哪个后台进程负责清理它

解答： PMON

164. 哪个后台进程刷新 materialized views?

解答：The Job Queue Processes.

165. 如何判断哪个 session 正在连结以及它们等待的资源？

解答：V$SESSION / V$SESSION_WAIT

166. 描述什么是 redo logs

解答：Redo Logs 是用于存放数据库数据改动状况的物理和逻辑结构。可以用来修复数据库

167. 如何进行强制 LOG SWITCH?

解答：ALTER SYSTEM SWITCH LOGFILE;

168. 举出两个判断 DDL 改动的方法？

解答：你可以使用 Logminer 或 Streams

169. Coalescing 做了什么？

解答：Coalescing 针对于字典管理的 tablespace 进行碎片整理，将临近的小 extents 合并成单个的大 extent.

170. TEMPORARY tablespace 和 PERMANENT tablespace 的区别是？

解答：temporary tablespace 用于临时对象例如排序结构而 permanent tablespaces 用来存储那些'真实'的对象(例如表，回滚段等)

171. 创建数据库时自动建立的 tablespace 名称？

解答：SYSTEM tablespace.

172. 创建用户时，需要赋予新用户什么权限才能使它联上数据库。

解答：CONNECT

173. 如何在 tablespace 里增加数据文件？

解答：ALTER TABLESPACE <tablespace_name> ADD DATAFILE <datafile_name> SIZE <size>

174. 如何变动数据文件的大小？

解答：ALTER DATABASE DATAFILE <datafile_name> RESIZE <new_size>;

175. 哪个 VIEW 用来检查数据文件的大小？

解答： DBA_DATA_FILES

176. 哪个 VIEW 用来判断 tablespace 的剩余空间

解答：DBA_FREE_SPACE

177. 如何判断谁往表里增加了一条纪录？

解答：auditing

178. 如何重构索引？

解答： ALTER INDEX <index_name> REBUILD;

179. 解释什么是 Partitioning（分区）以及它的优点。

解答：Partition 将大表和索引分割成更小，易于管理的分区。

180. 你刚刚编译了一个 PL/SQL Package 但是有错误报道，如何显示出错信息？

解答：SHOW ERRORS

181. 如何搜集表的各种状态数据？

解答： ANALYZE

The ANALYZE command.

182. 如何启动 SESSION 级别的 TRACE

解答: DBMS_SESSION.SET_SQL_TRACE

ALTER SESSION SET SQL_TRACE = TRUE;

183.IMPORT 和 SQL*LOADER 这 2 个工具的不同点

解答：这两个 ORACLE 工具都是用来将数据导入数据库的。

区别是：IMPORT 工具只能处理由另一个 ORACLE 工具 EXPORT 生成

的数据。而 SQL*LOADER 可以导入不同的 ASCII 格式的数据源

184.用于网络连接的 2 个文件？

解答： TNSNAMES.ORA and SQLNET.ORA

185. 说出 Servlet 的生命周期，并说出 Servlet 和 CGI 的区别？

解答：Servlet 被服务器实例化后，容器运行其 init 方法，请求到达时运行其 service 方法，service 方法自动派遣运行与请求对应的 doXXX 方法（doGet，doPost）等，当服务器决定将实例销毁的时候调用其 destroy 方法。

与 cgi 的区别在于 servlet 处于服务器进程中，它通过多线程方式运行其 service 方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，而 CGI 对每个请求都产生新的进程，服务完成后就销毁，所以效率上低于 servlet。

186、JAVA 中常用的 XML 解析技术有哪些？区别是什么？

解答： DOM、SAX 两种方式。

DOM：处理大型文件时其性能下降的非常厉害。这个问题是由 DOM 的数结构所造成的，这结构占用的内存较多，而且 DOM 必须在解析文件之前把整个文档载入内存，适合对 XML 的随机访问

SAX：不同于 DOM，SAX 是事件驱动型的 XML 解析方法。它顺序读取 XML 文件，不需要一次全部装载整个文件。当遇到像文件开头，文档结束，或者标签开头与标签结束时，它会触发一个事件，用户通过在其回调事件中写入处理代码来处理 XML 文件，适合对 XML 的顺序访问。

187、XML 文档定义有几种形式？有何本质区别？

解答：两种形式 dtd 和 schema，区别：

a.Schema 是标准的 XML 文件,而 DTD 则使用自己的特殊语法,因此,只需要知道 XML 的语法规则就可以编写 Schema 了,不需要再学习其它语法规则。

b.Schema 利用命名空间将文件中特殊的节点与 Schema 说明相联系,一个 XML 文件可以有多个对应的 Schema；而一个 XML 文件只能有一个相对应的 DTD 文件。

c.Schema 的内容模型是开放的,可以随意扩充,而 DTD 则无法解读扩充的内容.DTD 只能把文件类型定义为一个字符串,而 XML Schema 却允许把文件类型定义为整数,浮点数,字符串,布尔值或其他各各数据类型,而无须重新定义。

188.MVC 的各个部分都有哪些技术来实现？如何实现？

解答：MVC 是 Model-View-Controller 的缩写，Model 代表的是应用的业务逻辑（通过 JavaBean，EJB 组件实现），View 是应用的表示面（由 JSP 页面产生）Controller 是通过应用的处理过程控制，（一般是一个 servert）通过这种设计模型把应用逻辑，处理过程和显示逻辑分成不同的组件实现，这些组件可以进行交互和重用。

189 什么是垃圾回收？什么时候触发垃圾回收？如何降低垃圾回收的触发频率？它能保证程序有足够的可用内存吗？

解答：垃圾回收(GC)是 Java 语言的一个重要特性，作用是释放不再被使用的内存。垃圾回收由系统进行管理。在系统认为需要的时候自动启动一个线程进行处理。

尽量减少垃圾内存，也就是新建对象的数量，可以降低垃圾回收的频率。

垃圾回收机制无法保证有足够的内存。

190.什么是混淆(obfuscate)?有什么好处？有哪些工具可以混淆 jar 文件？

解答：混淆是指通过对 class 文件中的变量名和部分方法名进行处理，来提高代码反编译的难度。

好处主要有 2 个：1、提高反编译以后代码阅读的难度 2、降低 class 文件的大小。

常见的混淆器有 Proguard 和 RetroGuard 两种。

191.什么是状态机？游戏开发中有那些地方能用到状态机？

解答：状态机(State Machine)是根据对应状态进行处理的一种机制，在游戏开发中最典型的应用是游戏人工智能(AI)等地方。

192. 请根据你的知识，对以下计算机名词进行尽量简单的描述

解答：

1)J2ME 是一种使用 Java 语言进行嵌入式设备开发的技术。

2) Python 是一种语法简单的面对对象的程序设计语言

3)Ant 是 Java 的生成工具。

4)Javac 是 Java 语言的编译程序

5)Subversion 是新一代的版本工具

6)OpenGL 是一套开发的图形界面开发库标准

193.J2EE,EJB,JDBC 是一回事吗，它们之间有什么关系？

解答：不是一回事，J2EE（Java 2 Platform,Enterprise Edition）是一套全然不同于传统应用开发的技术架构，包含许多组件，主要可简化且规范应用系统的开发与部署，进而提高可移植性、安全与再用价值。J2EE 核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共

通的标准及规格，让各种依循 J2EE 架构的不同平台之间，存在良好的兼容性。其中，EJB 和 JDBC 都属于 J2EE 的一部分。

194.什么是数据库系统？

解答：数据库系统是储存、管理、处理和维护数据的软件系统，它由数据库、数据库管理员和有关软件组成。数据库系统的结构框架由外部层（单个用户的视图）、概念层（全体用户的公共视图）和内部层（存储视图）组成。

195.试述数据库完整保护的主要任务和措施。

解答：数据库的完整性保护也就是数据库中数据正确性的维护。数据库完整性包括三个内容：实体完整性规则，参照物完整性规则以及用户定义完整性规则。

前两个是有 DBMS 自动处理。

实体完整性规则是说针对于基表中的关键字中属性值不能为空值，是数据库完整性的基本要求，主关键字和元组的唯一性对应。

参照物完整性规则是不允许引用不存在的元组：即基表中的外关键字要么为空，要么关联基表中必存在元组。

用户定义的完整性规则针对具体的数据环境由用户具体设置的规则，它反应了具体应用中的语义要求。

一个完整性规则一般由下面三部分组成:完整性约束条件设置,完整性约束条件的检查以及完整性约束条件的处理.后两部分在数据库中一般有相应的模块处理。另外触发器也可以做完整性的保护，但触发器大量用于主动性领域。

196. 请写出十种以上你知道的 java 开源软件，并用一句话说明功能。

解答：

Ibatis：持久层框架

Hibernate：持久层框架，它对 JDBC 进行了非常轻量级的对象封装

Struts：是一个基于 Sun J2EE 平台的 MVC 框架

Spring：是一个解决了许多在 J2EE 开发中常见的问题的强大框架

Tomcat：实现了 servlet、struts 框架的 web 服务器

WebWork：组件化和代码重用的拉出式 MVC 模式 J2EE Web 框架

Rose：系统分析和设计的工具

JUnit：用于单元测试

ANT：用于辅助开发

Eclipse：IDE 集成开发工具

197. 部署一个 web 应用的步骤是什么？

解答：1.将 web 应用放到 Tomcat 服务器的 Webapps 包下，

2.启动服务器

3.在地址栏中输入 http://localhost:8080/应用名/

198. 什么是 UDDI、SOAP、WSDL？

解答：UDDI 是一套基于 Web 的、分布式的、为 Web Service 提供的、信息注册中心的实现标准规范，同时也包含一组使企业能将自身提供的 Web Service 注册，以使别的企业能够发现的访问协议的实现标准。

SOAP 即简单对象访问协议(Simple Object Access Protocol)，它是用于交换 XML 编码信息的轻量级协议。

WSDL 是一种 XML 格式，用于将网络服务描述为一组端点，这些端点对包含面向文档信息或面向过程信息的消息进行操作。这种格式首先对操作和消息进行抽象描述，然后将其绑定到具体的网络协议和消息格式上以定义端点。相关的具体端点即组合成为抽象端点(服务)。

199. Java 数据库编程包含哪些类？Java 数据库编程的基本过程是什么？

解答：用到的类：Connection、ResultSet、PreparedStatement、Statement

Java 中访问数据库的步骤如下：

1）注册驱动；

2）建立连接；

3）创建 Statement；

4）执行 sql 语句；

5）处理结果集（若 sql 语句为查询语句）；

6）关闭连接。

200. 什么是 B/S 结构,C/S 结构？

解答：C/S 是 Client/Server 的缩写。服务器通常采用高性能的 PC、工作站或小型机，并采用大型数据库系统，如 Oracle、Sybase、Informix 或 SQL Server。客户端需要安装专用的客户端软件。

B/S 是 Brower/Server 的缩写，客户机上只要安装一个浏览器(Browser)，如 Netscape Navigator 或 Internet Explorer，服务器安装 Oracle、Sybase、Informix 或 SQL Server 等数据库。在这种结构下，用户界面完全通过 WWW 浏览器实现，一部分事务逻辑在前端实现，但是主要事务逻辑在服务器端实现。浏览器通过 Web Server 同数据库进行数据交互。

201. 什么是 AJAX 和 AOP?

解答： Ajax 的全称是：AsynchronousJavaScript And XML。Ajax 不是一个技术，它实际上是几种技术，每种技术都有其独特这处，合在一起就成了一个功能强大的新技术。Ajax 包括： XHTML 和 CSS 使用文档对象模型(Document Object Model)作动态显示和交互 使用 XML 和 XSLT 做数据交互和操作 使用 XMLHttpRequest 进行异步数据接收 使用 JavaScript 将它们绑定在一起 。

AOP 是 OOP 的延续，是 Aspect Oriented Programming 的缩写，意思是面向切面编程。可以通过预编译方式和运行期动态代理实现在不修改源代码的情况下给程序动态统一添加功能的一种技术。AOP 实际是 GoF 设计模式的延续，设计模式孜孜不倦追求的是调用者和被调用者之间的解耦，AOP 可以说也是这种目标的一种实现。

202. 简要说明 JVM、JSP、Servlet、Web Server、Web Browser 之间的关系。

解答：当用户在 JSP 页面上提交了需要服务器处理的数据后，通过 Web Browser 发送到服务器端，Servlet 会根据用户的请求产生必要的相应，如果需要还会通过 JVM 或 Web Server 来获取资源，最后把服务器端的相应结果返回给用户。

203. 说说对开源项目 apache 的了解，说出其中你使用过的项目并给以评价

解答：

Apache HTTP Server（简称 Apache）是 Apache 软件基金会的一个开放源码的网页服务器，可以在大多数计算机操作系统中运行，由于其多平台和安全性被广泛使用，是最流行的 Web 服务器端软件之一。它快速、可靠并且可通过简单的 API 扩展，将 Perl / Python 等解释器编译到服务器中。

Apache 起初由伊利诺伊大学香槟分校的国家超级电脑应用中心（NCSA）开发。此后，Apache Httpd 被开放源代码团体的成员不断的发展和加强。Apache Http 网站服务器拥有牢靠可信的美誉，已经在全球超过半数的网站中被使用，特别是几乎所有最热门和访问量最大的网站。

Apache 支持许多特性，大部分通过编译的模块实现。这些特性从服务器端的编程语言支持到身份认证方案。一些通用的语言接口支持 Perl，Python， Tcl， 和 PHP。流行的认证模块包括 mod_access，mod_auth 和 mod_digest。其他的例子有 SSL 和 TLS 支持（mod_ssl），代理服务器 (proxy) 模块，很有用的 URL 重写（由 mod_rewrite 实现），定制日志文件（mod_log_config），以及过滤支持（mod_include 和 mod_ext_filter）。Apache 日志可以通过网页浏览器使用免费的脚本 AWStats 或 Visitors 来进行分析。

204. EJB 与 JAVA BEAN 的区别?

解答：Java Bean 是可复用的组件，对 Java Bean 并没有严格的规范，理论上讲，任何一个 Java 类都可以是一个 Bean。但通常情况下，由于 Java Bean 是被容器所创建（如 Tomcat)的，所以 Java Bean

应具有一个无参的构造器，另外，通常 Java Bean 还要实现 Serializable 接口用于实现 Bean 的持久性。Java Bean 实际上相当于微软 COM 模型中的本地进程内 COM 组件，它是不能被跨进程访问的。Enterprise Java Bean 相当于 DCOM，即分布式组件。它是基于 Java 的远程方法调用（RMI）技术的，所以 EJB 可以被远程访问（跨进程、跨计算机）。但 EJB 必须被布署在诸如 Webspere、WebLogic 这样的容器中，EJB 客户从不直接访问真正的 EJB 组件，而是通过其容器访问。EJB 容器是 EJB 组件的代理，EJB 组件由容器所创建和管理。客户通过容器来访问真正的 EJB 组件。

205. spring 有几种事务管理，spring 的事务管理接口是什么？

解答：Spring 提供的事务管理可以分为两类：编程式的和声明式的。编程式的，比较灵活，但是代码量大，存在重复的代码比较多；而声明式的比编程式的更灵活方便。

接口：其中最重要的三个接口：TransactionDefinition、PlatformTransactionManager 、 TransactionStatus 。 在 Spring 中 ， 事 务 是 通 过 TransactionDefinition 接口来定义的。该接口包含与事务属性有关的方法，TransactionDefinition 接口中定义了五个表示隔离级别的常量、代表传播行为的常量，在 TransactionDefinition 中以 int 的值来表示超时时间， PlatformTransactionManager.getTransaction(…) 方法返回一个 TransactionStatus 对象。返回的 TransactionStatus 对象可能代表一个新的或已经存在的事务（如果在当前调用堆栈有一个符合条件的事务）。TransactionStatus 接口提供了一个简单的控制事务执行和查询事务状态的方法。

206. 介绍一下 springMVC 的工作原理、为什么用 spring?

解答：

springMVC 工作原理：

1）.spring mvc 请所有的请求都提交给 DispatcherServlet,它会委托应用系统的其他模块负责负责对请求进行真正的处理工作。

2）.DispatcherServlet 查询一个或多个 HandlerMapping,找到处理请求的 Controller.

3）.DispatcherServlet 请请求提交到目标 Controller

4）.Controller 进行业务逻辑处理后，会返回一个 ModelAndView

5).Dispathcher 查询一个或多个 ViewResolver 视图解析器,找到 ModelAndView 对象指定的视图对象

6）.视图对象负责渲染返回给客户端。

为什么用 spring：

AOP 让开发人员可以创建非行为性的关注点，称为横切关注点，并将它们插入到应用程序代码中。使用 AOP 后，公共服务 （比 如日志、持久性、事务等）就可以分解成方面并应用到域对象上，同

时不会增加域对象的对象模型的复杂性。IOC 允许创建一个可以构造对象的应用环境，然后向这些对象传递它们的协作对象。正如单词 倒置 所表明的，IOC 就像反 过来的 JNDI。没有使用一堆抽象工厂、服务定位器、单元素（singleton）和直接构造（straight construction），每一个对象都是用其协作对象构造的。因此是由容器管理协作对象（collaborator）。Spring 既是一个 AOP 框架，也是一 IOC 容器。 Spring 最好的地方是它有助于您替换对象。有了 Spring，只要用 JavaBean 属性和配置文件加入依赖性（协作对象）。然后可以很容易地在需要时替换具有类似接口的协作对象。

207. get 和 post 的区别？

解答：Form 中的 get 和 post 方法，在数据传输过程中分别对应了 HTTP 协议中的 GET 和 POST 方法。二者主要区别如下：

1）Get 是用来从服务器上获得数据，而 Post 是用来向服务器上传数据；

2）Get 将表单中数据按照 variable=value 的形式，添加到 action 所指向的 URL 后面，并且两者使用“?”连接，而各个变量之间使用“&”连接；Post 是将表单中的数据放在 form 的数据体中，按照变量和值相对应的方式，传递到 action 所指向 URL；

3）Get 是不安全的，因为在传输过程，数据被放在请求的 URL 中；Post 的所有操作对用户来说都是不可见的；

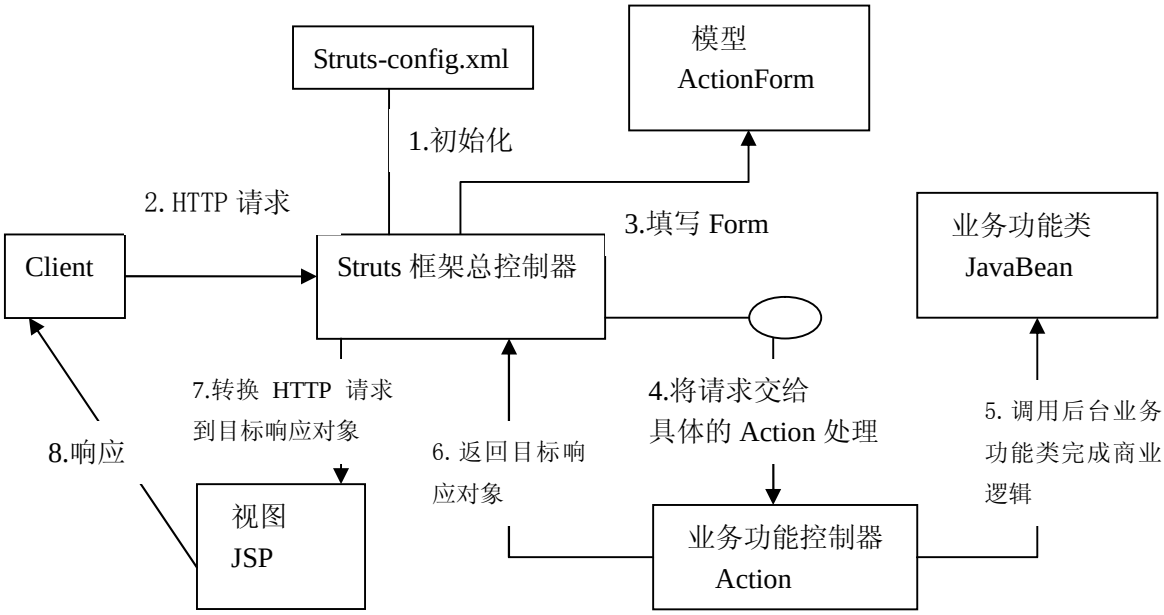
4）Get 传输的数据量小，这主要是因为受 URL 长度限制；而 Post 可以传输大量的数据，所以在上传文件只能使用 Post；

5）Get 限制 Form 表单的数据集必须为 ASCII 字符，而 Post 支持整个 ISO10646 字符集；

6）Get 是 Form 的默认方法。

208. 请简单画出 struts 技术构建 mvc 的流程图

解答



209. Jdo 是什么?

解答：JDO 是 Java 对象持久化的新的规范，为 java data object 的简称,也是一个用于存取某种数据仓库中的对象的标准化 API。JDO 提供了透明的对象存储，因此对开发人员来说，存储数据对象完全不需要额外的代码（如 JDBC API 的使用）。这些繁琐的例行工作已经转移到 JDO 产品提供商身上，使开发人员解脱出来，从而集中时间和精力在业务逻辑上。另外，JDO 很灵活，因为它可以在任何数据底层上运行。JDBC 只是面向关系数据库（RDBMS）JDO 更通用，提供到任何数据底层的存储功能，比如关系数据库、文件、XML 以及对象数据库（ODBMS）等等，使得应用可移植性更强。

210. 就 WEB2.0 相关信息做一下描述，谈谈你对这些技术概念的认识。

解答：一些 WEB2.0 产品，就可以理解以上观点。

Blog：用户织网，发表新知识，和其他用户内容链接，进而非常自然的组织这些内容。

RSS：用户产生内容自动分发，定阅。

Podcasting：个人视频/声频的发布/定阅。

SNS：blog+人和人之间的链接。

WIKI：用户共同建设一个大百科全书。

从知识生产的角度看，WEB1.0 的任务，是将以前没有放在网上的人类知识，通过商业的力量，放到网上去。

WEB2.0 的任务是，将这些知识，通过每个用户的浏览求知的力量，协作工作，把知识有机的组织起来，在这个过程中继续将知识深化，并产生新的思想火花；

从内容产生者角度看，WEB1.0 是商业公司为主体把内容往网上搬，而 WEB2.0 则是以用户为主，以简

便随意方式，通过 blog/podcasting 方式把新内容往网上搬；

从交互性看，WEB1.0 是网站对用户为主；WEB2.0 是以 P2P 为主。

从技术上看，WEB 客户端化，工作效率越来越高。

我们看到，用户在互联网上的作用越来越大；他们贡献内容，传播内容，而且提供了这些内容之间的链接关系和浏览路径。在 SNS 里面，内容是以用户为核心来组织的。WEB2.0 是以用户为核心的互联网

211. Servlet 是线程安全吗？以下代码中使用 synchronized 关键字的意义是什么？

```
Synchronized (aList){
        aList.remove (1);
    }
```

解答：默认不是线程安全的，但是 servlet 实现了 SingthreadModel 接口 就能单线程执行。此题中代码的意思是给 aList 对象加同步锁，保证 aList 对象在多线程任务环境中，每次只能够有一个线程调用 remove 方法。从而提高对 aList 对象操作的安全性和正确性。

212. JSP 中动态 INCLUDE 与静态 INCLUDE 的区别？

解答：动态 INCLUDE 用 jsp:include 动作实现 <jsp:include page="head.jsp"/>

它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数；静态 INCLUDE 用 include 伪码实现,它不会检查所含文件的变化，适用于包含静态页面<%@include file="head.htm" %>

213. Session 的基本原理是什么？

答：Session 对象的原理在于，服务器可以为客户端创建并维护一个所谓的 Session 对象，用于存放数据。在创建 Session 对象的同时，服务器将会为该 Session 对象产生一个唯一编号，这个编号称之为 SessionID，服务器以 Cookie 的方式将 SessionID 存放在客户端。当浏览器再次访问该服务器时，会将 SessionID 作为 Cookie 信息带到服务器，服务器可以通过该 SessionID 检索到以前的 Session 对象，并对其进行访问。需要注意的是，此时的 Cookie 中仅仅保存了一个 SessionID，而相对较多的会话数据保存在服务器端对应的 Session 对象中，由服务器来统一维护，这样一定程度保证了会话数据安全性，但增加了服务器端的内存开销。存放在客户端的用于保存 SessionID 的 Cookie 会在浏览器关闭时清除。我们把用户打开一个浏览器访问某个应用开始，到关闭浏览器为止交互过程称为一个“会话”。在一个“会话”过程中，可能会向同一个应用发出了多次请求，这些请求将共享一个 Session 对象，因为这些请求携带了相同的 SessionID 信息。Session 对象的正常使用要依赖于 Cookie。如果考虑到客户端浏览器可能出于安全的考虑禁用了 Cookie，应该使用 URL 重写的方式使 Session 在客户端禁用 Cookie 的情况下继续生效。

214. jsp 有哪些动作？作用分别是什么？

解答：JSP 共有以下 6 种基本动作：

jsp:include：在页面被请求的时候引入一个文件；

jsp:useBean：寻找或者实例化一个 JavaBean。；

jsp:setProperty：设置 JavaBean 的属性。；

jsp:getProperty：输出某个 JavaBean 的属性；

jsp:forward：把请求转到一个新的页面；

jsp:plugin：根据浏览器类型为 Java 插件生成 OBJECT 或 EMBED 标记

215. Struts 的控制器部分，视图部分包括哪些内容？

解答：

控制器:在 struts 中，基本的控制器组件是 ActionServlet 类中的实例 servelt，实际使用的 servlet 在配置文件中由一组映射（由 ActionMapping 类进行描述）进行定义。对于业务逻辑的操作则主要由 Action、ActionMapping、ActionForward 这几个组件协调完成的，其中 Action 扮演了真正的业务逻辑的实现者，ActionMapping 与 ActionForward 则指定了不同业务逻辑或流程的运行方向。struts-config.xml 文件配置控制器。

视图：视图主要由 JSP 建立，struts 包含扩展自定义标签库（TagLib），可以简化创建完全国际化用户界面的过程。目前的标签库包括：Bean Tags、HTML tags、Logic Tags、Nested Tags 以及 Template Tags 等。

216. WEB SERVICE 名词解释。JSWDL 开发包的介绍。JAXP、JAXM 的解释。

解答：Web Service 是基于网络的、分布式的模块化组件，它执行特定的任务，遵守具体的技术规范，这些规范使得 Web Service 能与其他兼容的组件进行互操作；

JAXP(Java API for XML Parsing)定义了在 Java 中使用 DOM, SAX, XSLT 的通用的接口，这样在你的程序中你只要使用这些通用的接口，当你需要改变具体的实现时候也不需要修改代码；

JAXM(Java API for XML Messaging)是为 SOAP 通信提供访问方法和传输机制的 API；

217. 介绍一下 hibernate 的工作原理、优点以及如何优化？

解答：

工作原理：

1)启动 Hibernate 构建 Configuration 实例，初始化该实例中的所有变量 Configuration cfg = new Configuration().configure();

2).加载 hibernate.cfg.xml 文件至该实例内存,通过 hibernate.cfg.xml 文件中的 mapping 节点配置，加载 hbm.xml 文件至该实例内存;

3).由上面得到的 Configuration 实例构建一个 SessionFactory 实例 SessionFactory sf = cfg.buildSessionFactory();

4).由上面得到的 SessionFactory 实例创建连接 Session s = sf.openSession();

5).由上面得到的 Session 实例创建事务操作接口 Transaction 的一个实例 tx Transaction tx = s.beginTransaction();

6).通过 Session 接口提供的各种方法操作数据库的访问

7).提交数据库的操作结果 tx.commit();

8).关闭 Session 链接 s.close();

优点：

1).对 JDBC 访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。

2).Hibernate 是一个基于 JDBC 的主流持久化框架，是一个优秀的 ORM 实现。他很大程度的简化 DAO 层的编码工作

3).hibernate 使用 Java 反射机制，而不是字节码增强程序来实现透明性。

4).hibernate 的性能非常好，因为它是个轻量级框架。映射的灵活性很出色。它支持各种关系数据库，从一对一到多对多的各种复杂关系。

优化：

1).使用双向一对多关联，不使用单向一对多

2).灵活使用单向一对多关联

3).不用一对一，用多对一取代

4).配置对象缓存，不使用集合缓存

5).一对多集合使用 Bag,多对多集合使用 Set

6).继承类使用显式多态

7).表字段要少，表关联不要怕多，可以利用二级缓存

218. Hibernate 实现中，load 和 get 方法的区别，find 和 iterate 的区别?

解答：load 和 get 方法的区别：hibernate 对于 load 方法认为该数据在数据库中一定存在，可以放心的使用代理来延迟加载，如果在使用过程中发现了问题，只能抛异常；而对于 get 方法，hibernate 一定要获取到真实的数据，否则返回 null。

find 和 iterate 的区别：find 方法通过一条 Select SQL 实现了查询操作，而 iterate 方法要执行多条 Select SQL.iterate 第一次查询获取所有符合条件的记录的 id，然后再根据各个 id 从库表中读取对应的记录,这是一个典型的 N+1 次的查询问题,如果符合条件记录有 10000 条,就需要执行 10001

条 Select SQL，性能会非常差。

219. JAVA 中如何进行事务的处理？

答：Connection 类中提供了 3 个事务处理方法：

setAutoCommit(Boolean autoCommit):设置是否自动提交事务，默认为自动提交，即为 true，通过设置 false 禁止自动提交事务；

commit():提交事务；

rollback():回滚事务。

220. Hibernate 有哪 5 个核心接口？

解答：Configuration 接口：配置 Hibernate，根据其启动 hibernate，创建 SessionFactory 对象；

SessionFactory 接口：初始化 Hibernate，充当数据存储源的代理，创建 session 对象，sessionFactory 是线程安全的，意味着它的同一个实例可以被应用的多个线程共享，是重量级、二级缓存；

Session 接口：负责保存、更新、删除、加载和查询对象，是线程不安全的，避免多个线程共享同一个 session，是轻量级、一级缓存；

Transaction 接口：管理事务；

Query 和 Criteria 接口：执行数据库的查询。

221. 什么是 ORM?

解答：对象关系映射（Object—Relational Mapping，简称 ORM）是一种为了解决面向对象与面向关系数据库存在的互不匹配的现象的技术；简单的说，ORM 是通过使用描述对象和数据库之间映射的元数据，将 java 程序中的对象自动持久化到关系数据库中；本质上就是将数据从一种形式转换到另外一种形式。

222. 数据连接池的工作机制是什么？

解答：J2EE 服务器启动时会建立一定数量的池连接，并一直维持不少于此数目的池连接。 客户端程序需要连接时，池驱动程序会返回一个未使用的池连接并将其表记为忙。如果当前没有空闲连接，池驱动程序就新建一定数量的连接，新建连接的数量有配置参数决定。当使用的池连接调用完成后，池驱动程序将此连接表记为空闲，其他调用就可以使用这个连接。

223. 解释四种会话跟踪技术？

解答：隐藏表单域、URL 重写,Cookie、Session。

1）.隐藏表单域：<input type="hidden">，非常适合步需要大量数据存储的会话应用。

2）.URL 重写:URL 可以在后面附加参数，和服务器的请求一起发送，这些参数为名字/值对。

3）.Cookie:一个 Cookie 是一个小的，已命名数据元素。服务器使用 SET-Cookie 头标将它作为 HTTP 响应的一部分传送到客户端，客户端被请求保存 Cookie 值，在对同一服务器的后续请求使用一个 Cookie 头标将之返回到服务器。与其它技术比较，Cookie 的一个优点是在浏览器会话结束后，甚至在客户端计算机重启后它仍可以保留其值。

4）.Session：使用 setAttribute(String str,Object obj)方法将对象捆绑到一个会话

224. Statement,PreparedStatement,CallableStatment 的区别

解答：

区别有以下几点：

1） Statement 是 PreparedStatement 和 CallableStatement 的父类；

2）Statement 是直接发送 Sql 语句到数据库，事先没有进行预编译。PreparedStatement 会将 sql 进行预编译,当 sql 语句要重复执行时,数据库会调用以前预编译好的 sql 语句,所以 PreparedStatement 在性能方面会更好；

3）PreparedStatement 在执行 sql 时，对传入的参数可以进行强制的类型转换。以保证数据格式与底层的数据库格式一致。

4）CallableStatement 适用与存储过程的查询表达语句

225. OOAD 中的 Association（联合）、Aggregation(聚合)、Composition（组合）的区别?

解答:三者从概念上来讲:Association 是一般的关联,有”use-a”的含义。Aggregation 和 Composition 都有整体和部分的关系，其中 Aggregation 中的部分脱离了整体，部分仍然有意义，有”has a”的含义，是共享式的。而 Composition 中的部分脱离了整体，部分将没有任何意义，是独占式的。

从代码实现的角度上讲：三者都是以属性出现，其中 Association 中作为属性出现时，不需要对其进行强制赋值，只要在使用是对其进行初始化即可。Aggregation 中作为属性出现时，需要在构造器中通过传递参数来对其进行初始化。Composition 中 作为属性出现时，需要在整体的构造器中创建部分的具体实例，完成对其的实例化。

从数据库的层面上来讲：Association 不需要被级联删除，Aggregation 不需要被级联删除，Composition 是需要被级联删除的。

226. 请说出你所知道的线程同步的方法。

解答：wait():使一个线程处于等待状态，并且释放所持有的对象的 lock；

sleep():使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉 InterruptedException 异常；

notify():唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某

一个等待状态的线程，而是由 JVM 确定唤醒哪个线程，而且不是按优先级；

notityAll():唤醒所有处入等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。

227. 谈谈对 ClassLoader 的理解？

解答：ClassLoader 加载类用的是全盘负责委托机制。所谓全盘负责，即是当一个 classloader 加载一个 Class 的时候，这个 Class 所依赖的和引用的所有 Class 也由这个 classloader 负责载入，除非是显式的使用另外一个 classloader 载入；委托机制则是先让 parent（父）类加载器 (而不是 super，它与 parent classloader 类不是继承关系)寻找，只有在 parent 找不到的时候才从自己的类路径中去寻找。此外类加载还采用了 cache 机制，也就是如果 cache 中保存了这个 Class 就直接返回它，如果没有才从文件中读取和转换成 Class，并存入 cache，这就是为什么我们修改了 Class 但是必须重新启动 JVM 才能生效的原因。

228. 进程和线程分别该怎么理解？

解答：进程是资源分配的基本单位。所有与该进程有关的资源，都被记录在进程控制块 PCB 中。以表示该进程拥有这些资源或正在使用它们。另外，进程也是抢占处理机的调度单位，它拥有一个完整的虚拟地址空间。

与进程相对应，线程与资源分配无关，它属于某一个进程，并与进程内的其他线程一起共享进程的资源。当进程发生调度时，不同的进程拥有不同的虚拟地址空间，而同一进程内的不同线程共享同一地址空间。

线程只由相关堆栈（系统栈或用户栈）寄存器和线程控制表 TCB 组成。寄存器可被用来存储线程内的局部变量，但不能存储其他线程的相关变量。

发生进程切换与发生线程切换时相比较，进程切换时涉及到有关资源指针的保存以及地址空间的变化等问题；线程切换时，由于同不进程内的线程共享资源和地址 空间，将不涉及资源信息的保存和地址变化问题，从而减少了操作系统的开销时间。而且，进程的调度与切换都是由操作系统内核完成，而线程则既可由操作系统内 核完成，也可由用户程序进行。

229. 同步和异步有何异同，在什么情况下分别使用他们？请举例说明

解答：如果数据将在线程间共享。例如正在写的数据以后可能被另一个线程读到，或者正在读的数据可能已经被另一个线程写过了，那么这些数据就是共享数据，必须进行同步存取。当应用程序在对象上调用了一个需要花费很长时间来执行的方法，并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多情况下采用异步途径往往更有效率。

230. sleep()和 wait()有什么区别？

解答：sleep 是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用 sleep 不会释放对象锁。

wait 是 Object 类的方法，对此对象调用 wait 方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出 notify 方法（或 notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

231. java 中会存在内存泄露吗？请简单描述。

解答：内存泄露是指系统中存在无法回收的内存，有时候会造成内存不足或系统崩溃。Java 存在内存泄露。Java 中的内存泄露当然是指：存在无用但是垃圾回收器无法回收的对象。而且即使有内存泄露问题存在，也不一定会表现出来。自己实现堆栈的数据结构时有可能会出现内存泄露。

232. 什么叫应用程序域？什么是托管代码？什么是强类型系统？什么是装箱和拆箱？什么是重载？CTS、CLS 和 CLR 分别作何解释？

解答：应用程序域：一种边界，它由公共语言运行库围绕同一应用程序范围内创建的对象建立（即，从应用程序入口点开始，沿着对象激活的序列的任何位置）。应用程序域有助于将在一个应用程序中创建的对象与在其他应用程序中创建的对象隔离，以使运行时行为可以预知。在一个单独的进程中可以存在多个应用程序域。

托管代码：由公共语言运行库环境（而不是直接由操作系统）执行的代码。托管代码应用程序可以获得公共语言运行库服务，例如自动垃圾回收、运行库类型检查和安全支持等。这些服务帮助提供独立于平台和语言的、统一的托管代码应用程序行为。

强类型系统：通过运行时类型识别（RTTI）(Run-Time Type Identification)，程序能够使用基类的指针或引用来检查这些指针或引用所指的对象的实际派生类型。

装箱、拆箱：从值类型接口转换到引用类型装箱。从引用类型转换到值类型拆箱。

重载：是方法的名称相同。参数或参数类型不同，进行多次重载以适应不同的需要。

CTS：通用语言系统。CLS：通用语言规范。CLR：公共语言运行库。

233. 程序注释的用途有哪些？带注释的程序有什么缺点？

解答：注释可以说明程序，给自己或他人在阅读程序时提供帮助，使程序更容易理解，也就是增强程序代码的可读性。过多的代码注释会使程序结构变得不清晰。

234. UDP 和 TCP 连接有和异同？

解答：TCP 协议是面向连接的，每个数据包的传输过程是：先建立链路、数据传输、然后清除链路。数据包不包含目的地址。受端和发端不但顺序一致，而且内容相同。它的可靠性高；UDP 协议是面向无连接的，每个数据包都有完整的源、目的地址及分组编号，各自在网络中独立传输，传输中不管其

顺序，数据到达收端后再进行排序组装，遇有丢失、差错和失序等情况，通过请求重发来解决。它的效率比较高

235. 构造器 Constructor 是否可以被继承？是否可以被 Override?

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

236. 什么是 java 序列化，如何实现 java 序列化？

解答：序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题；

序列化的实现：将需要被序列化的类实现 Serializable 接口，该接口没有需实现的方法，implements Serializable 只是为了标注该对象是可被序列化的，然后使用一个输出流(如 FileOutputStream)来构造一个 ObjectOutputStream(对象流)对象，接着，使用 ObjectOutputStream 对象的 writeObject(Object obj)方法就可以将参数为 obj 的对象写出(即保存其状态)，要恢复的话则用输入流。

237. 面向对象的特征有哪些方面？

解答：面向对象的特征主要有以下几个方面：

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。

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。

238. 接口是否可继承接口？抽象类是否可实现接口？抽象类是否可继承实体类?

解答：接口是可以继承接口的并且可以继承多个其它接口；抽象类可以实现接口中的方法；抽象类可

以继承实体类。

239. nonymous Inner Class (匿名内部类) 是否可以 extends(继承)其它类，是否可以 implements(实现)interface(接口)?

解答：匿名内部类是可以继承其它类，同样也可以去实现接口的，用法为：

```
JButton button = new JButton();
button.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e){//some mothod...}
});
```

这样的用法在 swing 编程中是经常使用的，就是因为它需要用到注册监听器机制,而该监听类如果只服务于一个组件,那么,将该类设置成内部类/匿名类是最方便的。

240. Bit 和 Byte 是什么意思?它们之间有什么关系?

解答：bit 中文名称是位，是用以描述电脑数据量的最小单位。

byte（字节）是计算机信息技术用于计量存储容量和传输容量的一种计量单位 1byte=8bit

241. 你认为 java 与其他（你所了解的）语言相比，有什么优点和缺点?

解答：1). Java 没有预处理指令。(如 C 中的#define , #include , #ifdef 等)。C 中的常量定义在 Java 中用 static final 来取代。

2). Java 中没有 C 中的全局变量。

3). Java 中的主类型的 size 是确定的，而 C 中主类型的 size 跟平台相关。

4). Java 中没有了指针，它使用了类似的句柄来取代指针，但是 Java 中不允许对句柄进行加减，没有取地址操作符之类的东东。

5). Java 有垃圾收集机制，不需要自己释放空间。

6). Java 没有 goto 语句。Java 在 C 提供的控制语句基础上增加了异常处理和标签 break 和 continue 语句。这些可以替代 goto 的作用。

7). C 要求一个方法或块中使用的所有局部变量的定义在该方法或块的最开始处定义，而 Java 允许这些定义在方法或块的任意地方出现。

8). Java 不要求在调用一个函数以前已经定义了该函数，可以在调用点后面定义。而 C 有这个要求。

9). Java 不支持 C 中的 strut 和 union 类型。Java 支持方法重载。

10). Java 不支持 C 中的 enum 关键字。

11). Java 不支持 C 中的 bitfields 能力。

12). Java 不支持 C 的 typedef。

13). Java 不支持 C 的方法指针。

14). Java 不支持 C 的可变参数表。

java 和.net 的都不适合做桌面程序,这两个比较适合写 WEB 的程序;

c++比较适合写桌面程序 c++/java 都是强类型的静态预编译型语言。优点是结构性强，程序可读性好，开发效率高，适合开发大型应用。就本人的经验而言，java 的开发效率优于 c++，实际上 java 大行其道的首要因素就是它够简单。java 尤其不适合开发桌面程序，GUI 的 API 一直都是 java 的弱点；

perl/python 是动态解释型语言。perl 是弱类型的而 python 是强类型的，后者的变量一旦赋值，就拥有了类型，不能再赋其他类型的值。不象 javascript/perl 可以随便定义。perl 是 unix 下的王牌工具，在缺乏 IDE 的字符界面下，很好地弥补了 unix；

shell/utility 的不足，并且部分继承了面向对象语言的灵活性。适合用来搭建大程序。

242. 一个 subclass 怎样调用 superclass 中的方法（myMethod）和构造函数?

答：用 super 关键字,子类去调用父类的方法，如：super.myMethod();子类去调用父类的构造函数，如：super();。

243. 排序都有哪几种方法？用伪代码实现一个快速排序

答：排序的方法有：插入排序（直接插入排序、希尔排序），交换排序（冒泡排序、快速排序），选择排序（直接选择排序、堆排序），归并排序，分配排序（箱排序、基数排序）

快速排序的伪代码：使用快速排序方法对 a[0:n-1]排序从 a[0:n-1]中选择一个元素作为 middle，该元素为支点;把余下的元素分割为两段 left 和 right,使得 left 中的元素都小于等于支点,而 right 中的元素都大于等于支点;

递归地使用快速排序方法对 left 进行排序；递归地使用快速排序方法对 right 进行排序；所得结果为 left + middle + right。

243. String a=null; if (a!=null && a.length()>10) {...}

上面面的代码，如果你用“&”替换“&&”将发生什么错误？

答：会抛出空指针异常；&&会短路，只要遇到 boolean 值为 false 就不会再往后执行了；而&则是会把两边的运算都执行完。

三　编程题（30）

1.编程实现：二分搜索算法

解答：

```
public class SearchTest {
    /** 被搜索数据的大小 */
    private static final int size = 5000000;
    public static void main(String[] args) {
        long[] data = new long[size];
        // 添加测试数据
        for (int k = 0; k < data.length; k++) {
            data[k] = k;
        }
        // 要查找的数据
        long target = 4970002;
        binaryFindTest(data, target);
    }
    /**
     * 二分搜索算法实现
     *
     * @param data
     *            数据集合
     * @param target
     *            搜索的数据
     * @return 返回找到的数据的位置，返回-1 表示没有找到。
     */
    public static int binaryFind(long[] data, long target) {
        int start = 0;
        int end = data.length - 1;
```

```
        while (start <= end) {
            int middleIndex = (start + end) / 2;
            if (target == data[middleIndex]) {
                return middleIndex;
            }
            if (target >= data[middleIndex]) {
                start = middleIndex + 1;
            } else {
                end = middleIndex - 1;
            }
        }
        return -1;
    }
    /**
     * 二分搜索测试
     *
     * @param data
     *            数据集合
     * @param target
     *            搜索的数据
     */
    public static void binaryFindTest(long[] data, long target) {
        long start = System.nanoTime();
        int result = binaryFind(data, target);
        long end = System.nanoTime();
        System.out.println("binary search position：" + result);
        System.out.println("binary search time：" + (end - start));
    }
}
```

2.编程实现：线程 A 向队列 Q 中不停写入数据，线程 B 从队列 Q 中不停读取数据（只要 Q 中有数据）。

解答：

接口中有两个一个是向队列中写 push 方法 一个是从队列中读。

```
public interface StackInterface
{
    public void push(int n);
    public int[] pop();
}
```

上边接口的实现类。

```
public class SafeStack implements StackInterface {
    private int top = 0;
    private int[] values = new int[10];
    private boolean dataAvailable = false;
    public void push(int n) {
        synchronized (this) {
            while (dataAvailable) // 1
            {
                try {
                    wait();
                } catch (InterruptedException e) {
                    // 忽略 //2
                }
            }
            values[top] = n;
            System.out.println("压入数字" + n + "步骤 1 完成");
            top++;
            dataAvailable = true;
            notifyAll();
            System.out.println("压入数字完成");
        }
    }
```

```
    public int[] pop() {
        synchronized (this) {
            while (!dataAvailable) // 3
            {
                try {
                    wait();
                } catch (InterruptedException e) {
                    // 忽略 //4
                }
            }
            System.out.print("弹出");
            top--;
            int[] test = { values[top], top };
            dataAvailable = false;
            // 唤醒正在等待压入数据的线程
            notifyAll();
            return test;
        }
    }
}
```

读线程

```
public class PopThread implements Runnable
{
  private StackInterface s;
  public PopThread(StackInterface s)
  {
    this.s = s;
  }
  public void run()
  {
```

```
    while(true)
    {
        System.out.println("->"+ s.pop()[0] + "<-");
        try {
                Thread.sleep(100);
        }
        catch(InterruptedException e){}
    }
  }
}
```

写线程

```
public class PushThread implements Runnable
{
  private StackInterface s;
  public PushThread(StackInterface s)
  {
    this.s = s;
  }
  public void run()
  {
    int i = 0;

    while(true)
    {
      java.util.Random r = new java.util.Random();
      i = r.nextInt(10);
      s.push(i);
      try {
        Thread.sleep(100);
      }
```

```
      catch(InterruptedException e){}
    }
  }
}
```

3.编程实现：使用 Socket 经行网络通信时，客户端和服务器端流程。

解答：

服务器，使用 ServerSocket 监听指定的端口，端口可以随意指定（由于 1024 以下的端口通常属于保留端口，在一些操作系统中不可以随意使用，所以建议使用大于 1024 的端口），等待客户连接请求，客户连接后，会话产生；在完成会话后，关闭连接。

客户端，使用 Socket 对网络上某一个服务器的某一个端口发出连接请求，一旦连接成功，打开会话；会话完成后，关闭 Socket。客户端不需要指定打开的端口，通常临时的、动态的分配一个 1024 以上的端口。

4.编写代码实现同一平面内两圆是否碰撞，其中：

第一个圆圆心坐标为(x1,y1)，半径是 r1，第二个圆圆心坐标为(x2,y2)，半径是 r2。

方法声明如下：

```
boolean collisWith(int x1,int y1,int r1,int x2,int y2,int r2){}
```

解答：

```
boolean collisWith(int x1, int y1, int r1, int x2, int y2, int r2) {
        boolean flag=false;
        int num1=(x1-x2)*(x1-x2);
        int num2=(y1-y2)*(y1-y2);
        int num3=num1+num2;
        double distance=Math.sqrt(num3);
        if(distance<= (r1+r2) ){
            flag=true;
        }
        return flag;
    }
```

5.判断一个 int 数组中的元素是否存在重复，方法声明如下：

```
boolean isRepeat(int[] m){  }
```

解答：

```
public boolean isRepeat2(int[] m){
Set h =new HashSet(m.length);
for (int i = 0; i < m.length; i++) {
h.add(new Integer(m[i]));
}
if (h.size()==m.length ){
return false;
}else {
return true;
}
}
```

6.用递归方法实现正序显示数组元素。例如 String[] s = {“a”,”b”,”c”,”d”};

方法声明如下：

```
void print(String[] s,int i){ }
```

解答：参数 i 是指打印 string 数组的起始位置，原理是正序打印 s 从第 0 个开始的所有字符串，等价于先打印第 0 个，在打印 s 中从第一个开始的所有字符串，如此递归

```
void print(String[] s, int i) {
        if ((i >= 0) && (i < s.length)) {
            System.out.print(s[i]);
            i++;
            print(s, i);
        }
}
```

7.请写出求 n! 的算法。

解答：

```
public class Factorial {
    public static void main(String[] args) {
        long n = 6;
        System.out.println(doFactorial(n));
```

```
    }
    public static long doFactorial(long n) {
        if (n < 1) {
            System.out.println("ERROR");
            return 0;
        } else if (n == 1 || n == 2) {
            return n;
        } else {
            return n * doFactorial(n - 1);
        }
    }
}
```

8.在当前的 JSP 网页里，提交用户名和密码，提交给 post . jsp, post . jsp 打印出用户名和密码并返回给浏览器。请写出 post . jsp

解答：

假设页面用户名和密码在 login.jsp 里，login.jsp 页面代码如下：

```
<form action=”post.jsp” method=”post”>
  <input type=”text” name=”userName”>
  <input type=”password” name=”pwd”>
  <input type=”submit”>
</form>
```

post.jsp 页面代码：

```
<%
String userName=request.getParameter(“userName”);
String pwd=request.getParameter(“pwd”);
out.println(“用户名：”+userName+”,密码：”+pwd);
%>
```

9.编写一个字符界面的 Java Application 程序，接受用户输入的 10 个整数，并输出这 10 个整数的最大值和最小值。

解答：采用了冒泡进行排序

```
import java.util.Scanner;
import java.util.Scanner;
public class MaxAndMin {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int[] arr = new int[10];
        for (int i = 0; i < arr.length; i++) {
            int next = scanner.nextInt();
            arr[i] = next;
        }
        int[] after=Arrays.sort(arr);
        System.out.println("最小值："+after[0]+",最大值："+after[arr.length-1]);

    }

}
```

10.写一个排序算法 1-100 随机数字 进行排序 要求效率。

解答：

```
public class Sort {
    // 选择排序方法
    public static void selectionSort(int[] number) {
        for (int i = 0; i < number.length - 1; i++) {
            int m = i;
            for (int j = i + 1; j < number.length; j++) {
                if (number[j] < number[m])
                    m = j;
            }
            if (i != m)
                swap(number, i, m);
        }
```

```
    }
    // 用于交换数组中的索引为 i、j 的元素
    private static void swap(int[] number, int i, int j) {
        int t;
        t = number[i];
        number[i] = number[j];
        number[j] = t;
    }
    public static void main(String[] args) {
        // 定义一个数组
        int[] num = new int[100];
        for(int i=0;i<num.length;i++){
            num[i]=(int)(Math.random()*100)+1;
        }
        // 排序
        selectionSort(num);
        for (int i = 0; i < num.length; i++) {
            System.out.println(num[i]);
        }
    }
}
```

11.冒泡排序:依次比较相邻的两个数,将大数放在前面,小数放在后面。第一趟结束,在最后的数必是所有数中的最小数。重复以上过程,直至最终完成排序。由于在排序过程中总是大数往前放,小数往后放,相当于气泡往上升,所以称作冒泡排序。请用 JAVA 语言编写一个完成冒泡排序算法的程序。

解答:

```
int[] bubbleSort(int before[]) {
        int t;
        for (int i = 0; i < before.length; i++) {
            for (int j = 0; j < before.length - i - 1; j++) {
                if (before[j] > before[j + 1]) {
```

```
                    t = before[j];
                    before[j] = before[j + 1];
                    before[j + 1] = t;
                }
            }
        }
        return before;
    }
```

12.写出一段 socket 通讯（客户端）的代码，功能描述如下：

a)客户端发起 socket 通讯，报文结构为报文号（3 位）+用户名（5 位）+密码（8 位）+ 结束符（固定为 END)。此处报文号为 100

b)服务端收到后返回应答报文，报文结构为报文号（3 位）+验证结果（2 位）+结束符（固定为 END)。此处报文号为 101

c)Socket 服务器 ip 为 192.168.0.2，端口号为 9999

解答：

客户端代码：

```
Socket sk = new Socket("192.168.0.2",9999);
OutputStream os = sk.getOutputStream();
PrintWriter pw = new PrintWriter(os,true);
pw.write("100stone888888END");
pw.close();
sk.close();
```

服务器端代码：

```
ServerSocket vk = new ServerSocket(9999);
Socket sk = vk.accept();
OutputStream os = sk.getOutputStream();
PrintWriter pw = new PrintWriter(os,true);
pw.write("101oldEND");
pw.close();
sk.close();
```

13.编写函数 insert(String str)，将字符串”a,123;b,456;c,789”置入 HashMap 中。

解答：

```
import java.util.HashMap;
public class HashMapDemo {
    HashMap<String,String> map=new HashMap<String,String>();
    public  void insert(String str){
        map.put("a", str);
    }
    public static void main(String[] args) {
        HashMapDemo demo=new HashMapDemo();
        demo.insert("a,123;b,456;c,789");
    }
}
```

14.有一数组 a[1000]存放了 1000 个数,这 1000 个数取自 1-999, 且只有两个相同的数,剩下的 998 个数不同, 写一个搜索算法找出相同的那个数的值（请用 C# or JAVA 编程实现，注意空间效率和时间效率尽可能优化）。

解答：

```
import java.util.Arrays;
public class SearchDemo {
    /** 被搜索数据的大小 */
    private static final int size = 1000;

    public static void main(String[] args) {
        int[] data = new int[size];

        // 添加测试数据
        for (int k = 0; k < data.length; k++) {
            data[k] = k + 1;
        }
        data[999] = 567;
```

```
    result(data);

}
/**
 * 调用分搜索算法的方法实现查找相同元素
 * @param data
 */

public static void result(int data[]){
    Arrays.sort(data);
    for (int i = 0; i < data.length; i++) {
        int target = data[i];
        data[i] = 0;
        int result = binaryFind(data, target);
        if (result != -1) {
            System.out.println("相同元素为："+data[result]);
            break;
        }
    }
}

/**
 * 二分搜索算法实现
 *
 * @param data
 *            数据集合
 * @param target
 *            搜索的数据
 * @return 返回找到的数据的位置，返回-1 表示没有找到。
 */
```

```
    public static int binaryFind(int[] data, int target) {
        int start = 0;
        int end = data.length - 1;
        while (start <= end) {
            int middleIndex = (start + end) / 2;
            if (target == data[middleIndex]) {
                return middleIndex;
            }
            if (target >= data[middleIndex]) {
                start = middleIndex + 1;
            } else {
                end = middleIndex - 1;
            }
        }
        return -1;
    }
}
```

15.下面是一个由*号组成的 4 行倒三角形图案。要求：1、输入倒三角形的行数，行数的取值 3-21 之间，对于非法的行数，要求抛出提示“非法行数!”；2、在屏幕上打印这个指定了行数的倒三角形。

```
*******
*****
***
*
```

解答：

```
import java.util.Scanner;
public class Lines {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        int lines = scanner.nextInt();
        if (lines > 3 && lines < 21) {
```

```
            for (int i = lines-1; i >= 0; i--) {
                for (int z = 0; z <= i * 2; z++) {
                    System.out.print("*");
                }
                System.out.print("\n");
            }
        }else{
            System.out.println("非法行数！");
        }
    }
}
```

16.现有一个 32 位的整型变量 value 和一个有 32 个元素的数组 a[32]，要求：1、对 value 随机赋值；2、让数组 a[n]的值等于 value“位 n”的值，0<=n<=31。举例：如果 value 的“位 0”(Bit0)=0，那么 a[0]=0；如果 value 的“位 10”(Bit10)=1，那么 a[10]=1。

解答：

```
public class Foo {
    public static void main(String[] args) {
        //产生随机数
        int random = (int) (Math.random() * Integer.MAX_VALUE + 1);
        //转成二进制字符串
        String str=Integer.toBinaryString(random);
        //转成二进制时最前面的零被省略，补上省略的 0
        if(str.length()<32){
            for(int j=0;j<=32-str.length();j++){
            str="0"+str;
            }
        }
        //给数组赋值
        int[] a=new int[32];
        for(int i=0;i<str.length();i++){
```

```
            a[i]=Integer.parseInt(String.valueOf(str.charAt(i)));
            System.out.println("a["+i+"]="+a[i]);
        }
    }
}
```

17.现有 1~100 共一百个自然数，已随机放入一个有 98 个元素的数组 a[98]。要求写出一个尽量简单的方案，找出没有被放入数组的那 2 个数，并在屏幕上打印这 2 个数。注意：程序不用实现自然数随机放入数组的过程。

答：

```
int[] b = new int[]{....存入 98 个随机的 1~100 的整数};
int[] a = new int[100];
for(int t : b)
  a[t-1]=t;
for(int t=0; t < a.length; t++)
  if(a[t]==0)
      System.out.println(t+1);
```

18. 实现函数 public String[ ] array(List list)，其中参数 list 中元素类型为字符串

解答：

```
public String[] array(List list) {
        String[] elementData = new String[list.size()];
        for(int i=0;i<list.size();i++){
            elementData[i]=(String)list.get(i);
        }
        return elementData ;
    }
```

19. 创建类 Person，其中存储的成员数据为:age(int),sex(boolean),weight(int)，至少有一个构造函数可以初始化这三个属性值，同时提供获取这三个属性值的 public 方法

解答：

```
public class Person {
    private int age;
```

```
    private boolean sex;
    private int weight;
public Person() {
    }
public Person(int age, boolean sex, int weight) {
        this.age = age;
        this.sex = sex;
        this.weight = weight;
    }
public int getAge() {
        return age;
    }
public boolean isSex() {
        return sex;
    }
public int getWeight() {
        return weight;
    }
}
```

20. 设计线程类 WorkerThread，其构造函数接受一个 message 字符串作为参数，把该字符串打印到 console 上，同时，在 WorkThread 的 main 函数中启动该线程。

解答：

```
public class WorkerThread extends Thread {
    public WorkerThread(String message) {
        System.out.println(message);
    }

    public static void main(String[] args) {
       new WorkerThread("hello world!").start();
    }
```

}

21.写一个函数去掉一个字符串中单词间多余的空格，使得相邻两个单词间有且只有一个空格。例如当输入字符串是“Hello!_ _Game_programming_ _world!”时，调用该函数后字符串变为“Hello!_Game_programming_world!”。

解答：

```
/**
     * 去除字符串中多余的空格
     *
     * @param s
     *            需要处理的字符串
     * @return 处理后的字符串
     */
    public String trimSpace(String before) {
        String temp= "" + before.charAt(0);
        for (int i = 1; i < before.length(); i++) {
            char c = before.charAt(i);
            // 如果当前字符是空格
            if (c == ' ') {
                // 判断前一个不是是空格则添加，否则不添加
                if (before.charAt(i - 1) != ' ') {
                    temp += c;
                }
            } else {
                temp += c;
            }
        }
        return temp;
    }
```

22. 编写一个程序，用来计算 1 到 100 间所有整数的和是多少？

解答：

```
public static void GetSum()
 {
  int sum = 0;
  for(int i=1;i<=100;i++)
  {
   sum+=i;
  }
  System.out.println("和为："+sum);
 }
```

23.请简单写出用 JAVA 连接 Oracle 数据库,并执行一条/SQL 语句。(只需要写关键几条语句即可,/SQL 语句：SELECT*FROM t_users WHERE users_id= ‘1111’)

解答：

```
        Class.forName("oracle.jdbc.OracleDriver");
        String url = "jdbc:oracle:thin:@127.0.0.1:1521:orcl";
        String user = "scott";
        String password = "tiger";
        Connection con = DriverManager.getConnection(url, user, password);
        Statement stm = con.createStatement();
ResultSet rs = stm
                .executeQuery("SELECT*FROM t_users WHERE users_id='1111'");
        while (rs.next()) {
            // 取值
        }
        rs.close();
        stm.close();
        con.close();
```

24.在 web 应用开发过程中经常遇到输出某种编码的字符，如从 GBK 到 iso8859-1 等，如何输出一个某种编码的字符串？

```
    public static String translate(String str) {
        String tempStr = "";
```

```
        try {
            tempStr = new String(str.getBytes("ISO-8859-1"), "GBK");
            tempStr = tempStr.trim();
        } catch (Exception e) {
            System.err.println(e.getMessage());
        }
        return tempStr;
    }
```

25. 请写出一个公用方法，输入 String 返回该串是否含有非空字符,并写出 junit 的测试用例

答：

```
public class TestString {
    public static boolean hasBlank(String str) {
        if (str.endsWith("") || str.startsWith("")) {
            return false;
        } else {
            String[] strs = str.split("");
            if (strs.length == 1) {
                return false;
            }
        }
        return true;
    }
    @Test
    public void testFun() {
        System.out.println(TestString.hasBlank("test"));
    }
}
```

26. JAVA 实现一种排序

答：用插入法进行排序代码如下

```
package com.tarena;
```

```
import java.util.*;
class InsertSort
{
ArrayList list;
public InsertSort(int num,int mod)
{
list = new ArrayList(num);
Random rand = new Random();
System.out.println("The ArrayList Sort Before:");
for (int i=0;i<num ;i++ )
{
list.add(new Integer(Math.abs(rand.nextInt()) % mod + 1));
System.out.println("list["+i+"]="+list.get(i));
}
}
public void SortIt()
{
Integer tempInt;
int MaxSize=1;
for(int i=1;i<list.size();i++)
{
tempInt = (Integer)list.remove(i);
if(tempInt.intValue()>=((Integer)list.get(MaxSize-1)).intValue())
{
list.add(MaxSize,tempInt);
MaxSize++;
System.out.println(list.toString());
}
else
{
```

```
for (int j=0;j<MaxSize ;j++ )
{
if (((Integer)list.get(j)).intValue()>=tempInt.intValue())
{
list.add(j,tempInt);
MaxSize++;
System.out.println(list.toString());
break;
}
}
}
}
System.out.println("The ArrayList Sort After:");
for(int i=0;i<list.size();i++)
{
System.out.println("list["+i+"]="+list.get(i));
}
}

public static void main(String[] args)
{
InsertSort sort = new InsertSort(10,100);
sort.SortIt();
}
}
```

27. 编写一个截取字符串的函数，输入为一个字符串和字节数，输出为按字节截取的字符串。 但是要保证汉字不被截半个，如"我 ABC"4，应该截为"我 AB"，输入"我 ABC 汉 DEF"，6，应该输出为"我 ABC"而不是"我 ABC+汉的半个"。

答：

```
package com.tarena;
```

```
public class SplitString {
    String SplitStr;
    int SplitByte;

    public SplitString(String str, int bytes) {
        SplitStr = str;
        SplitByte = bytes;
        System.out.println("The String is:" + SplitStr + ";SplitBytes="
                + SplitByte);
    }

    public void SplitIt()
    {
        int loopCount;

loopCount=(SplitStr.length()%SplitByte==0)?(SplitStr.length()/SplitByte):(SplitStr.leng
th()/SplitByte+1);
        System.out.println("Will Split into "+loopCount);

        for (int i=1;i<=loopCount ;i++ )
        {
            if (i==loopCount){
            System.out.println(SplitStr.substring((i-1)*SplitByte,SplitStr.length()));
            } else {
                System.out.println(SplitStr.substring((i-1)*SplitByte,(i*SplitByte)));
            }
        }
    }   public static void main(String[] args) {
        SplitString ss = new SplitString(
                "test 中 dd 文 dsaf 中男大 3443n 中国 43 中国人 0ewldfls=103", 4);
```

```
        ss.SplitIt();
    }
}
```

28. 编写程序将由数字及字符组成的字符串中的数字截取出来并按顺序输出，例如：“ABC137GMNQQ2049PN5FFF”输出结果应该为 01234579

答：

```
package com.tarena;
import java.util.Arrays;
public class NumberSplitChar {
    public static void main(String[] args) {
       String str="ABC137GMNQQ2049PN5FFF";
       char[] beforechars=str.toCharArray();
       char[] afterchars=new char[beforechars.length];
       int j=0;
       for(int i=0;i<beforechars.length;i++){
           if(beforechars[i]>='0' && beforechars[i]<='9'){
               afterchars[j++]=beforechars[i];
       }
     }
         Arrays.sort(afterchars);
         for(int i=(afterchars.length-j);i<afterchars.length;i++){
            System.out.print(afterchars[i]);
         }
    }
}
```

29. 请用 JAVA 实现两个类，分别实现堆栈（Stack)和队列（Queue）操作。

答：

```
public class MyStack {
    private List list;
    public MyStack(){
        list = new ArrayList();
```

```
    }
    public boolean isEmpty(){
        return list.size() == 0;
    }
    public void push(Object obj){
        list.add(obj);
    }
    public Object pop(){
        if(list.size()>0){
            Object obj = list.get(list.size()-1);
            list.remove(list.size()-1);
            return obj;
        }else{
            return null;
        }
    }
    public int getNumber(){
        return list.size();
    }
}
class IntegerQueue {
    public int[] integerQueue;// 用来当队列
    public int tail;// 队尾
    public int size;// 队的长度，也可以设置一个默认值，溢出时从新申请

    public IntegerQueue(int size) {
        integerQueue = new int[size];
        this.size = size;
        tail = 0;
    }
```

```
    public void inQueue(int i) {
        if (tail < size) {
            this.integerQueue[tail] = i;
            tail++;
        } else {
            System.err.println("溢出啦！");
        }
    }

    public int outQueue() {
        if (tail >= 0) {
            int tmp = this.integerQueue[tail];
            tail--;
            return tmp;
        } else {
            System.err.println("队列为空！");
            throw new RuntimeException();
        }
    }
}
```

30. 假定屏幕的像素宽度为 screenWidth，写一个函数计算一个字符串需要分成几行显示。

要求：

1)、每行应尽可能多地显示字符，但不能有字符部分或完全显示在屏幕外。超过部分的字符换下一行显示。

2)、每个字符的像素宽度不一样，每个字符的像素宽度不一样。用 int GetCharWidth(char c)获得每个字符的像素宽度。

```
    /**
     * 计算一个字符串可以分多少行进行显示
     *
```

```
 * @param s
 *            原始字符串
 * @param screenWidth
 *            屏幕宽度
 * @return 行数
 */
int calcLineNum(String s, int screenWidth) {
    int length = 0;
    // 行数
    int n = 0;
    // 统计长度
    for (int i = 0; i < s.length(); i++) {
        // 当前字符的宽度
        int charLen = GetCharWidth(s.charAt(i));
        // 总长度增加
        length += charLen;
        // 如果达到屏幕宽度
        if (length > screenWidth) {
            n++; // 行数+1
            length = charLen; // 重新计算长度
        }
    }
    // 最后一行处理
    if (length > 0) {
        n++;
    }
    return n;
}
```

四　数据库写 SQL 题（30）

1.按要求写 SQL 语句：根据集团成员培训业务，建立以下三张表：

S （S#,SN,SD,SA） S#,SN,SD,SA 分别代表学号、学员姓名、所属单位、学员年龄

C （C#,CN ） C#,CN 分别代表课程编号、课程名称

SC （ S#,C#,G ） S#,C#,G 分别代表学号、所选修的课程编号、学习成绩

要求如下：

1)使用标准 SQL 语句查询成员名单中所属单位叫“技术一部”的人员总数及

平均年龄；

2)使用标准的 SQL 语句更新学号为‘S#1’的姓名为“Mike”;

3)使用嵌套语句查询选修课程编号为‘C2’的学员姓名和所属单位；

4)使用嵌套语句查询不选修课程编号为‘C5’的学员姓名和所属单位；

5)查询选修课程超过 5 门的学员学号和所属单位；

解答：

1) select count(SN),avg(SA) from S where SD='技术一部';

2) update S set SN='Mike' where S#='S#1';

3) select SN,SD from S where S#=(select S# from SC where C#='C2');

4) select SN,SD from S where S# not in(select S# from SC where C#='C5');

5) select S#,SD from S where S#=

(select S# from SC group by S# having count(S#)>=5);

2.请根据以下四张表（其中 course_t 表的 teacher_id 字段是 teacher_t 表的 id 字段的外键引用），

拼写出相应的 sql 语句（oracle 语法）。(15 分)

学生表：students_t

id	name	sex
001	赵学生	Male
002	钱学生	Male
003	孙学生	Male
004	李学生	Female
005	周学生	Female

…	…	…

教师表：teacher_t

id	name	sex
001	吴老师	Male
002	郑老师	Male
003	王老师	Male
004	刘老师	Female
005	张老师	Female

课程表：course_t

id	name	credit	teacher_id
001	语文	3	001
002	数学	3	002
003	英语	4	003
004	物理	3	004
005	化学	2	005
006	政治	1	001
007	生物	1	005
008	计算机	2	005

选课表：student_course_t

id	student_id	course_id
001	001	001
002	001	002
003	001	003
004	002	001
005	002	007
…	…	…

1)统计每个学生选修的学分，并按学分降序排序

2)统计每个学生选修的所有课程和对应的任课老师；并按学生 Id 和课程 Id 排序

3)统计所有学生、所有课程和所有任课老师的对应关系；并按学生 Id 和课程 Id 排序

解答：

```
1) select sc.student_id,count(c.credit)
from students_t s, course_t c, student_course_t sc
 where s.id=sc.student_id and c.id=sc.course_id group by
sc.student_id order by count(c.credit);
2) select s.name as s_name,c.name as c_name ,t.name as t_name
from students_t s, course_t c, student_course_t sc,teacher_t t
 where s.id=sc.student_id and c.id=sc.course_id and t.id=c.teacher_id order by s.id,c.id;
```

3)与 2）相同

3.假设有以下的两个表：

Cus_A

ID*	Name	Address
…	…	…

Cus_B

ID*	Name	Address
…	…	…

*主键

表 Cus_A 和表 Cus_B 的结构完全相同，表 Cus_A 和表 Cus_B 中既存在 ID 相同的记录，也存在 ID 不同的记录。现要求将 ID 只存在于表表 Cus_A 中而不存在于表 Cus_B 中的记录全部插入到 Cus_B 表中，并用表 Cus_A 中的记录更新 Cus_B 中相同的 ID 的记录，请写出完成这一功能的存储过程。

解答：

```
create or replace procedure test
is
 cust_record cus_a%rowtype ;
 cursor cust_cursor is select id,name,address from cus_a;
```

```
Begin
  Open cust_cursor;
  LOOP
    Fetch cust_cursor into cust_record;
  EXIT WHEN cust_cursor %NOTFOUND;
        --先删除在插入
        delete from cus_b where id=cust_record.id;
        insert into cus_b values(cust_record.id, cust_record.name, cust_record.address);
  END LOOP;
end;
```

4、已有“成绩”如下表所示：

学号	课程号	分数
S1	C1	80
S1	C2	75
S2	C1	null
S2	C2	55
S3	C3	90

1） 执行 SQL 语句：

Select Count（学号）From 成绩 Where 分数〉60

后的结果是什么？

2）请写出 SQL 语句来进行查询“成绩”表中学号为 S1、课程号为 C2 的学号和分数

解答：

1）统计分数超过 60 的学生总数。

2）select 学号，分数 from 成绩 where 学号=‘S1’ and 课程号=‘C2’；

5.SAL 是 Product 表中的索引列，请优化如下 SQL 语句，并简述原因。原语句：

```
SELECT*
FROM Product
WHERE SAL * 12 〉25000;
```

解答：

Select * from product where sal>(25000/12);

理由：WHERE 子句中，如果索引列是函数的一部分. 优化器将不使用索引而使用全表扫描.

6.有一张表，字段有用户名、口令及备注，请用 SQL 选择出用户名和口令完全相同的记录（应包括用户名和数量的出现次数）

T_USER(USER_NAME,PASSWORD)

显示

USER_NAME COUNT(*)

QWE 4

WER 5

解答：select user_name,count(*) from t_user group by user_name,password;

7.有一张表，T_MONEY，字段有 ID，FEE，请用 SQL 语言选择出 FEE 值为前三条记录。

T_MONEY(ID，FEE)

显示

ID FEE

2 100

1 90

2 80

Select Id,fee from (Select id,fee from t_money order by fee desc) where rownum<=3;

8、table_name temp

Id name

1 a

2 b

3 a

4 a

结果为

Id name

1 a

2 b

写出 sql 语句。

解答：select rownum as id , name from(select distinct name from temp);

9、已知原表(t_salary)

year salary

2000 1000

2001 2000

2002 3000

2003 4000

先要实现显示结果(salary 为以前的工资和)

year salary

2000 1000

2001 3000

2002 6000

写出 sql 语句。

解答：

```
select t.year, sum(t.salary) over (order by t.year) as sum_salary from salary_t t;
```

10.有两个表 A 和 B，均有 key 和 value 两个字段，如果 B 的 key 在 A 中也有，就把 B 的 value 换为 A 中对应的 value

这道题的 SQL 语句怎么写?

解答:

```
merge into A a
using B b
on (a.key=b.key)
when matched then
update set
a.value=b.value
```

11.创建一张数据表，并插入如下数据。

购物人	商品名称	数量
A	甲	2
B	乙	4
C	丙	1

A 乙 2

B 丙 5

1)写出创建表和插入内容的 sql 语句

2)写出 sql 语句使其产生如下结果

购物人	商品甲	商品乙	商品丙
A	2	2	Null
B	Null	4	5
C	Null	Null	1

解答：

```
create table tb_order(
     customer varchar2(20),
     product_name varchar2(20),
     quantity number(2)
)
Insert into tb_order(customer,product_name,quantity)values(‘A’,’甲’,2);
Insert into tb_order(customer,product_name,quantity)values(‘B’,’乙’,4);
Insert into tb_order(customer,product_name,quantity)values(‘C’,’丙’,1);
Insert into tb_order(customer,product_name,quantity)values(‘A’,’甲’,2);
Insert into tb_order(customer,product_name,quantity)values(‘B’,’乙’,5);
2) select customer "购物人",
sum(decode(product_name,'甲',quantity,0)) "商品甲",
sum(decode(product_name,'乙',quantity,0)) "商品乙",
sum(decode(product_name,'丙',quantity,0)) "商品丙"
from tb_order
group by customer;
```

12.有如下两张表：部门表和职员表，每个职员都属于一个部门，表结构如下：

Dept 表	
Deptno	Deptname
…	…

Emp 表		
Empno	Empname	Deptno
…	…	…

请使用 SQL 语句查询每个部门有多少职员，要求查询结果包含两例（部门名称，人数）？

解答：select d.deptname,count(*) from dept d,emp e where d.deptno=e.deptno

group by d.deptno,d.deptname;

13.业务场景：存在下面的表及记录

GOODS（进货表）

GOODSID（主键）	GOODSNAME	MEMO
1	青霉素	
2	西瓜霜	
3	创可贴	
4	西洋参	

SU（进货表）

GOODSID（主键）	SUQTY
1	60
2	70

SA（销售表）

GOODSID（主键）	SAQTY
3	80
4	90

要求一：进货记录，给出 SQL 达到以下结果

GOODSID（主键）	GOODSNAME	SUQTY
1	青霉素	60
2	西瓜霜	70
3	创可贴	0
4	西洋参	0

要求二：进销对比，给出 SQL 达到以下结果

GOODSID（主键）	GOODSNAME	SUQTY	SAQTY
1	青霉素	60	0
2	西瓜霜	70	70
3	创可贴	0	80

要求三：将 GOODS.MEMO 更新为[进货数量 SU.SUQTY]

解答：

1)select g.goodsid,g.goodsname,s.suqty from goods g inner join su s on g.goodsid=s.goodsid;

2 ） select g.goodsid,g.goodsname,s.suqty,a.saqty from goods g, su s,sa a on g.goodsid=s.goodsid and g.goodsid=a.goodsid;

3)update goods set demo=(select s.suqty from su s where s.goodsId=goods.goodsId)

14.表结构:

1) 表名:apply

字段(字段名/类型/长度):

applyno varchar 8;//申请单号(关键字)

applydate bigint 8;//申请日期

state varchar 2;//申请状态

2) 表名:applydetail

字段(字段名/类型/长度):

applyno varchar 8;//申请单号(关键字)

name varchar 30;//申请人姓名

idcard varchar 18;//申请人身份证号

state varchar 2;//申请状态

其中，两个表的关联字段为申请单号。

题目:

1)查询身份证号码为 440401430103082 的申请日期

2)查询同一个身份证号码有两条以上记录的身份证号码及记录个数

3)删除 applydetail 表中所有姓李的记录

解答：

1) Select applydate from apply a join applydetail d on a.applyno=d.applyno and

Idcard='440401430103082';

2) select idcard,count(*) from applydetail group by idcard having count(*)>2;

3) delete from applydetail where name='李%';

15. 在 system 方案中建立表 table1,表中包含如下字段

字段名称 数据类型 要求

name Varchar2 非空

id Number 非空

age Number

sex Varchar2

salary Number

解答:

```
Create table system.table11 (
Id number not null,
Name varchar(8) not null,
Age number,
Sex varchar(2),
Salary number
);
```

16、某公司的机构结构为树型结构，对应的表结构为 TableCompany(ComCode—机构代码，UpperComCode—上级机构代码)，如何查询出总公司的所有下级机构？(java 或者 SQL 均可)。你觉得这种思维和设计是否合理？有什么好建议的？

答：

```
select t1.* from TableCompany t1, TableCompany t2
    Where t1.ComCode = t2.UpperComCode
```

这种设计比较容易让人理解，但是表中的数据联系过于紧密，数据量很大，会给后期维护造成不便，如果根据第三范式要求，将每一子公司独立成一张表，对于关系的维护和数据的管理都会变得比较方便。

17、一个简单的论坛系统，以数据库存储如下数据：

用户名，发帖标题，发帖内容，回复标题，回复内容。

每天论坛访问量 200 万左右，更新帖子 10 万左右。

请给出数据库表结构设计，并结合范式简要说明设计思路。

答：用户表：存储用户信息；

用户所发的帖子表：存储用户所发的帖子；

回复表：存储对帖子所做的回复。

设计：

User：

```
Create table tb_user(
id number(10) primary key,
Uname varchar2(20) not null unique
);
```

Comments:

```
Create table tb_comments(
     id number(10),
     comments_id number(20) not null unique,
     title varchar2(20) not null,
     comments varchar2(255) not null,
     foreign key(id) references  tb_user(id)
);
```

Replay:

```
Create table tb_replay(
id number(10),
comments varchar2(255) not null,
foreign key(id) references tb_comments(comments_id)
);
```

思路：因为此应用所要存储的数据量比较大，所以为了避免数据的冗余，表的设计依托于第三范式。

18、有一个数据表 userinfo，包含 userid，username 字段，其中 userid 是唯一的，username 可能重复，请写一句 sql 查询语句，把重复的记录全部取出来。

userid	username
1	老王
2	老王
3	老李

4　　老李

5　　小张

要求返回记录集

userid　username

1　　老王

2　　老王

3　　老李

4　　老李

答：

select * from userinfo where username in (select username from userinfo group by username having count(username)>1);

19、建表 Department 部门

字段名	中文名称	类型	长度	备注
depid	部门号	变长字符	10	主键
depname	部门名称	变长字符		
depcj	部门平均成绩	浮点型保留 2 位小数		

表 Employee 人员表

字段名	中文名称	类型	长度	备注
empid	员工号	变长字符	10	主键
name	姓名	变长字符	10	
depid	部门号	变长字符	10	
Cj	成绩	浮点型保留 2 位小数		
xorder	名次	整型		

实现表中的记录备下面相关题目使用

Department 表中嵌入记录

部门号	部门名称
A001	人力资源部
A002	财务部

Employee 表中嵌入记录

员工号	姓名	部门号	成绩
001	张三	A001	90
002	李四	A001	90
003	王五	A001	80
004	张飞	A002	70
005	刘备	A002	60
006	关羽	A002	50

1）写出建表以及嵌入记录语句

2）显示 A001 部门员工的姓名、成绩

3）显示所有员工的员工号、姓名、部门名称、成绩

4）将关羽的成绩修改成 52 分

5）按要求写视图 VdepEmpMax 求各部门的最高分，显示部门号、最高分成绩

6）按要求写存储过程 SP_Calc 求各部门的平均成绩，并更新到 Department 表 depcj 字段中 7）按要求写存储过程 SP_Order 求员工的名次，并更新到 Employee 表 xorder 字段中

8）按要求写视图 VdepEmp2，求各部门的前 2 名，显示部门号、员工号、成绩

排序规则如下：

员工	部门	分数	名次
张三	A001	90	1
李四	A001	90	1
张飞	A002	70	1
刘备	A002	60	2

答：

1）

```
create table Department(depid varchar2(20) primary key,
depname varchar2(20),
depcj number(10,2));
create table Employee(empid varchar2(20) primary key,
name varchar2(20),
depid varchar2(20),
```

```
cj number(10,2),
xorder number(10));

insert into Department(depid,depname) values('A001','人力资源部');
insert into Department(depid,depname) values('A002','财务部');
insert into Employee(empid, name, depid, cj) values('001','张三','A001',90);
insert into Employee(empid, name, depid, cj) values('002','李四','A001',90);
insert into Employee(empid, name, depid, cj) values('003','王五','A001',80);
insert into Employee(empid, name, depid, cj) values('004','张飞','A002',70);
insert into Employee(empid, name, depid, cj) values('005','刘备','A002',60);
insert into Employee(empid, name, depid, cj) values('006','关羽','A002',50);
```

2) `select name,cj from employee where depid='A001';`

3) `select e.empid,e.name,d.depname,e.cj from employee e, department d where e.depid=d.depid;`

4) `update employee set cj=52 where name=' 关羽';`

5)

```
create view VdepEmpMax as (select deptid,max(cj) from employee e group by deptid)
```

6)

```
create or replace procedure SP_Calc
as
begin
update department d set depcj=(
select nvl(avg(cj),0) from employee e
where e.depid(+)=d.depid);
end;
```

7)

```
create or replace procedure SP_Order
as
begin
update employee w set xorder =(select b.rn from
(select empid,rank() over (partition by depid order by cj  desc ) rn from employee) b
```

where w.empid=b.empid);

end;

8）

create or replace view VdepEmp2 as

select depid,name,cj,rn from

(select e.*,rank() over (partition by depid

order by cj desc) rn from employee e)

where rn<3;

20、数据库基础:

1)使用 SQL 语句创建学生表 students

字段: 学号:s_id 姓名:s_name 年龄:age 班级:class 辅导员:assistant (请设计各字段类型与长度)

2)查询学生表中年龄大于 20 的所有学生的学号与姓名

3)删除 0201 班的所有同学

4)查询 0302 班姓李的学生的个数

5)将班编号以’02’开头的所有班级的辅导员修改为‘李四’

答：1）create table students(s_id number(10) primary key,

s_name varchar(30) not null,

age number(3) not null,

class varchar(20) not null,

assistant varchar(30));

2）select s_id,s_name from students where age>20;

3）delete from students where class=’0201’;

4）select count(s_name) from students

where s_name like ‘李%’ and class='0302';

5）update students set assistant='李四’ where class like '02%';

21、表名：高考信息表 students_info

准考证号 科目 成绩

no subject score

2006001 语文 119

2006001　　数学　　108

2006002　　物理　　142

2006001　　化学　　136

2006001　　物理　　127

2006002　　数学　　149

2006002　　英语　　110

2006002　　语文　　105

2006001　　英语　　98

2006002　　化学　　129

写出高考总分在600以上的学生准考证号的SQL

答：

```
select no
from students_info
group by no
having sum(score)>600;
```

22、有一个表LEANR，表里有三个字段分别是学号（student_id），

课程（kc）,成绩（grade）。

1).查询每一门课程的前两名

2).查询以Grade降序排列的第31至40条记录(不需要区分课程)

3).查询表中存在课程重复4次以上的记录,显示课程和重复的次数,并且按照重复次数的降序排列

答：

```
1).select student_id,kc,grade
  from (select student_id,kc,grade,
        row_number() over(partition by kc order by grade desc)rn  from LEANR)
  where rn<=2;
2)select student_id,grade
  from (
         select lea.*,rownum rm
         from (
            select *
```

```
            from LEANR
            order by grade desc
          ) lea
    where rownum < 41
  )
  where rm between 31 and 40;
```

3). select kc,count(kc)

```
  from LEANR
  group by kc
  having count(kc)>=2
  order by count(kc) desc;
```

23、a 部门表 b 员工表

a 表字段(id --部门编号 departmentName-部门名称)

b 表字段(id--部门编号 employee- 员工名称)

问题:如何一条 sql 语句查询出每个部门共有多少人

答:

建表语句:

```
create table a(
id number primary key,
departmentName varchar(20)
);
create table b(
id number,
employee varchar(20)
);
insert into a values(1,'部门 1');
insert into a values(2,'部门 2');
insert into a values(3,'部门 3');
insert into b values(1,'emp1');
insert into b values(1,'emp2');
```

```
insert into b values(1,'emp3');
insert into b values(2,'emp4');
insert into b values(2,'emp5');
insert into b values(3,'emp6');
select departmentName,count(employee) from a,b where a.id=b.id group by departmentName;
```

24、为管理岗位业务培训信息，建立 3 个表:

S (SID,SN,SD,SA) SID,SN,SD,SA 分别代表学号、学员姓名、所属单位、学员年龄

C (CID,CN) CID,CN 分别代表课程编号、课程名称

SC (SID,CID,G) SID,CID,G 分别代表学号、所选修的课程编号、学习成绩

1. 使用标准 SQL 嵌套语句查询选修课程名称为’税收基础’的学员学号和姓名

2. 使用标准 SQL 嵌套语句查询选修课程编号为’02’的学员姓名和所属单位

3. 使用标准 SQL 嵌套语句查询不选修课程编号为’03’的学员姓名和所属单位

4. 使用标准 SQL 嵌套语句查询选修全部课程的学员姓名和所属单位

5. 查询选修课程超过 5 门的学员学号和所属单位

答:

建表 sql 语句:

```
create table s(
sid int(10) primary key,
sn varchar(20) not null,
sd varchar(20) not null,
sa int(3) not null
 );
create table c(
cid int(10) primary key,
cn varchar(20) not null
);
create table sc(
sid int(10) references s(sid),
cid int(10) references c(cid),
g int(10),
```

```
primary key(sid,cid)
);
insert into s values(1,"zhangsan","project",25);
insert into s values(2,"lisi","mis",26);
insert into s values(3,"wangwu","manager",27);
insert into s values(4,"zhaoliu","mis",26);
insert into c values(01,"税收基础");
insert into c values(02,"Core Java");
insert into c values(03,"NetWork");
insert into sc values(1,01,70);
insert into sc values(1,02,75);
insert into sc values(1,03,80);
insert into sc values(2,01,80);
insert into sc values(2,03,69);
insert into sc values(3,02,73);
1)
select s.sid,s.sn
from s,c,sc
where s.sid=sc.sid
and c.cid=sc.cid
and c.name='税收基础';
2) select a.sn,a.sd
from s a, c b
where b.cid in(select c.cid from sc c where a.sid=c.sid and b.cid=c.cid)
and b.cid=02;
3) select a.sn,a.sd
from s a, c b
where b.cid not in(select c.cid from sc c where a.sid=c.sid and b.cid=c.cid)
and b.cid=03;
4) select sn,sd
```

```
from s where sid in
(select sid from sc  group by sid having count(cid)=(select count(cid) from c));
```

5) select sn,sd from s

```
where sid in(select sid from sc group by sid  having count(distinct cid)>5);
```

25、请根据以下要求来完成题目：

会议室预定模块:某公司有多个会议室，以房间号区分。如果某部门需要预定会议室，则会提交预定请求（包含预定开始使用时间、预定结束使用，所预定会议室房间号）。

设计一个表，保存会议室预定信息。

要求采用 SQL 语句及 JAVA 代码段判断 在 2003-3-10 下午 3：00~4:00 3 号会议室是否空闲。

请写出有关 SQL 语句以及相关 JAVA 的代码段。

答：

1)Sql 语句：

```
create table meeting(
id number primary key ,
room_id varchar(10),
isUsed char,
begin timestamp,
end timestamp
);

insert into meeting values(1,'201',1,to_date('2003-03-10 15:00:00','yyyy-mm-dd hh24:mi:ss')
,to_date('2003-03-10 16:00:00','yyyy-mm-dd hh24:mi:ss'));

insert into meeting values(2,'201',1,to_date('2003-03-10 17:00:00','yyyy-mm-dd hh24:mi:ss')
,to_date('2003-03-10 22:00:00','yyyy-mm-dd hh24:mi:ss'));
```

2)

```
package com.tarena;
import java.sql.*;
```

```
public class Test {
    public static void main(String[] args) {
        String driverName = "oracle.jdbc.OracleDriver";
        String url = "jdbc:oracle:thin:@127.0.0.1:1521:orcl";
        String username = "scott";
        String pwd = "tiger";
        Connection con = null;
        Statement stmt = null;
        ResultSet rs = null;
        try {
            Class.forName(driverName);
            con = DriverManager.getConnection(url, username, pwd);
            stmt = con.createStatement();
            String sql = "select isUsed from " +
                    "meeting " +
                    "where   ((begin  between  to_date('2003-03-10  15:00:00','yyyy-mm-dd
hh24:mi:ss') and  to_date('2003-03-10 16:00:00','yyyy-mm-dd hh24:mi:ss'))  " +
                   "or(end between to_date('2003-03-10 15:00:00','yyyy-mm-dd hh24:mi:ss')
and  to_date('2003-03-10 16:00:00','yyyy-mm-dd hh24:mi:ss')))" +
                    " and room_id=201";
            if (stmt.execute(sql)) {
                rs = stmt.getResultSet();
            }
            StringBuffer sb = new StringBuffer();
            while (rs.next()) {
                sb.append("isFree:" + rs.getInt(1) + "  ");
            }
            System.out.print(sb.toString());
        } catch (Exception e) {
```

```
            e.printStackTrace();
        } finally {
            try {
                con.close();
            } catch (Exception e1) {
                e1.printStackTrace();
            }
        }
    }
}
```

26、下面是两个数据库表，分别记录员工姓名和工资

T_EMPLOYEE	
ID	NAME
2	张三
3	李四
5	王五
·····	·····

T_SALARY	
ID	SALARY
2	3400
3	4300
5	2500
·····	·····

1.查询表 T_EMPLOYEE 中 id = 3 的员工记录

2.查询表 T_EMPLOYEE 中所有员工记录

3.联合查询表 T_EMPLOYEE 和 T_SALARY 中所有员工的姓名和工资记录，并按照薪水从高到低排列

答：

1).select * from t_employee where id = 3;

2).select * from t_employee;

3).select e.name,s.salary

from t_employee e,t_salary s

where e.id=s.id

order by s.salary;

27、有三张表，学生表 S，课程表 C，学生课程表 SC，学生可以选修多门课程，一门课程可能被多个学生选修，通过 SC 表关联。

1)写出建表以及插入语句；

2)写出 SQL 语句，查询选修了所有选修课程的学生；

3)写出 SQL 语句，查询选修了至少 2 门以上的课程的学生。

答：

1)

```
create table student (id number(10) primary key,name varchar2(20));
create table course (id number(10) primary key,name varchar2(20));
create table sc(sid number(10) references student(id),cid number(10) references course(id),grade number(4,2));
insert into student values(1,'feifei');
insert into student values(2,'jingjing');
insert into student values(3,'nannan');
insert into student values(4,'yuanyuan');
insert into student values(5,'jiejie');
insert into course values(1,'corejava');
insert into course values(2,'c++');
insert into course values(3,'jdbc');
insert into course values(4,'hibernate');
insert into sc values(1,1,98);
insert into sc values(2,1,97);
insert into sc values(3,1,94);
insert into sc values(4,1,92);
insert into sc values(5,1,93);
```

```
insert into sc values(1,2,94);
insert into sc values(2,2,92);
insert into sc values(3,2,95);
insert into sc values(5,2,97);
insert into sc values(1,3,92);
insert into sc values(2,3,92);
insert into sc values(4,3,91);
insert into sc values(1,4,99);
insert into sc values(3,4,89);
```

2）select sid,count(*) from sc group by sid having count(*)=(select count(*) from course);

3）select sid,count(*) from sc group by sid having count(*)>=2;

28、SQL 题

```
--操作员表
select pkid,name,sys_corp_id '单位主键' from base_operator
--角色表
select pkid,sys_corp_id '单位主键',name from base_role
--角色与操作员的对应关系表
select pkid,base_role_id '角色主键',base_operator_id '操作员主键' from base_role_operator
--单位表
select pkid,name from sys_corps
--问题:
--1.显示出'开发'公司所拥有的操作员
--2.显示出'开发'公司每个角色所对应的操作员信息
--3.显示出'开发'公司每个角色所对应的操作员的个数
```

答：1).Select name from base_operator;

2).Select op.pkid,op.name,op.sys_corp_id
From base_operator op,base_role_operator ro ,base_role br
Where ro.base_operator_id =op.pkid
And br.pkid=ro.base_role_id;

3).Select max(br.name) ,count(*)

```
      From base_role_operator ro,base_role br
      Where ro.base_role_id=br.pkid
Group by br.base_role_id;
```

29、说明在一个系统中权限管理中应该有哪些表、表间关系、各表哪些功能？

答：

角色和权限表是 m：n 的关系

操作表和权限表是 1：m 的关系

模块表和操作表是 1：m 的关系

表的大体设计如下：

```
-- 角色表
create table roles (
    id  number primary key,
    name    varchar2(20)
);
-- 系统模块表
create table modules (
    id  number primary key,
    name    varchar2(50),
    url     varchar2(50)
);
-- 模块操作表
create table operations (
    id  number primary key,
    name    varchar2(20),
    mid     number,
    constraint foreign key (mid) references modules(id)
);
-- 权限表
create table rights (
    id      number primary key,
```

```
    name        varchar2(20),
    url         varchar2(50),
    operationid int references operations(id)
);
-- 角色权限设置表
create table rolerights (
    id      number primary key,
    roleid  number references roles(id),
    rightid number references rights(id)
);
```

30、说出下面语句的作用：

```
Select  rownum,last_name,salary
From  (select last_name,salary from s_emp order by salary desc)
Where  rownum<=10;
```

答：选出 s_emp 表中工资前 10 名员工的姓名和工资。