

## JAVASE 部分

1、Choose the three valid identifiers from those listed below. (Choose three)?

- A. IDoLikeTheLongNameClass
- B. \$byte
- C. const
- D. \_ok
- E. 3\_case

答: ABD

2、Which of the following lines of code will compile without error (Choose two)?

A.

```
int i=0;
if (i) {
    System.out.println( "Hi" );
}
```

B.

```
boolean b=true;
boolean b2=true;
if(b==b2) {
    System.out.println( "So true" );
}
```

C.

```
int i=1;
int j=2;
if(i==1|| j==2)
    System.out.println( "OK" );
```

D.

```
int i=1;
int j=2;
if (i==1 &| j==2)
    System.out.println( "OK" );
```

答: BC

3、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;
}
```

答: DE

4、What will happen when you attempt to compile and run the following code?

```
public class Static{
    static {
        int x = 5;
    }
    static int x,y;
    public static void main(String args[]){
        x--;
        myMethod();
        System.out.println(x + y + ++x);
    }
    public static void myMethod(){
        y = x+++++x;
    }
}
```

- A. compiletime error
- B. prints: 1
- C. prints: 2
- D. prints: 3
- E. prints: 7
- F. prints: 8

答: D

5、What is the correct ordering for the import, class and package declarations when found in a single file?

- A. package, import, class
- B. class, import, package
- C. import, package, class
- D. package, class, import

答: A

6、What will happen when you attempt to compile and run the following code.

```
public class Pvf{
    static boolean Paddy;
    public static void main(String argv[]){
        System.out.println(Paddy);
    }
}
```

- A. Compile time error                      B. compilation and output of false  
 C. compilation and output of true            D. compilation and output of null  
 答: B

7、 Given the following 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;

答: ABD

8、 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) {...}

答: AD

9、 给出如下声明:

```
String s = "Example";
```

合法的代码有哪些?

- A) s>>>=3    B) s[3]= "X"    C) int i = s.length()    D) s = s + 10

答: CD

10、 如下哪些不是 java 的关键字?

- A) const    B) NULL    C) false    D) this    E) native

答: B

11、 关于垃圾收集的哪些叙述是对的

- A) 程序开发者必须自己创建一个线程进行内存释放的工作  
 B) 垃圾收集将检查并释放不在使用的内存  
 C) 垃圾收集允许程序开发者明确指定并立即释放该内存  
 D) 垃圾收集能够在期望的时间释放被 java 对象使用的内存

答: B

12、 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

答: D

13、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。

答: D

14、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可运行状态。

答: BCD

15、已知如下代码:

```
public class Test
{
    long a[] = new long [10];
    public static void main(String arg[] ){
        System.out.print(a[6]);
    }
}
```

请问哪个语句是正确的?

- A) Output is null.
- B) Output is 0
- C) When compile, some error will occur.
- D) When running, some error will occur.

答: C

16、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

答: AD

17、已知如下代码:

```
switch (m)
{
case 0: System.out.print( "Condition 0" );
case 1: System.out.print( "Condition 1" );
case 2: System.out.print( "Condition 2" );
case 3: System.out.print( "Condition 3" );
default: System.out.print( "Other Condition " );
}
```

当m的值为什么时能输出“Condition 2”

A) 0    B) 1    C) 2    D) 3    E) 4    F)None

答: ABC

18、下面的哪些声明是合法的?

A) long l = 4990    B) int i = 4L    C) float f =1.1    D) double d = 34.4

答: AD

19、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

答: C

20、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()

答: C

21、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;

答: DE

22、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);

答: CD

23、给出下面代码:

```

public class Person{
    static int arr[ ] =new int[10];
    public static void main (String a [ ] ) {
        System.out.println (arr[ 1 ]);
    }
}

```

哪个选项是正确的?

- A. 编译时将产生错误;
- B. 编译时正确, 运行时将产生错误;
- C. 输出 0;
- D. 输出空。

答: C

24、关于运算符>>和>>>描述正确的是

- A. >>执行移动
- B. >>执行翻转
- C. >>执行有符号右移, >>>执行无符号右移
- D. >>执行无符号右移, >>>执行有符号右移

答: C

25、下列关于栈的叙述正确的是

- A 栈是非线性结构
- B 栈是一种树状结构
- C 栈具有先进先出的特征
- D 栈具有后进先出的特征

答: D

26、选出 Java 语言中的关键词 (多选)

- A. NULL
- B. sizeof
- C. implements
- D. extends

答: CD

27、现有下述代码, 选择出能够打印出"Test2"的数字 (多选)

```
switch(x){
    case 1:
        System.out.println("Test1");
    case 2:
    case 3:
        System.out.println("Test2");
        break;
}
System.out.println("Test3");
```

- A.0
- B.1
- C.2
- D. 3
- E. 4

答: CD

28、字符(char)的整型表示范围为

- A. 0 ... 32767
- B. 0 ... 65535
- C. -256 ... 255
- D. -32768 ... 32767

答: B

29、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: ADE

30、从下列选项中选择正确的 Java 表达式 (多选)

- A. `int k=new String("aa");`
- B. `String str=String("bb");`
- C. `char c=74;`
- D. `long j=8888;`

答: CD

31、下述代码的执行结果为

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

- A.代码能够编译, 打印出“s=”
- B.代码能够编译, 打印出“s=null”
- C.代码不能够编译, 因为 `String s` 没有初始化
- D.代码能够编译, 但是当调用 `toString()`时抛出 `NullPointerException` 异常

答: B

32、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.println(sooper.getLength()+ “,” + sub.getLength() );
    }
}
```

- A. 4, 4
- B. 4, 5
- C. 5, 4
- D. 5, 5
- E. 代码不能被编译

答: E

33、下述程序的执行结果为

```
int index = 1;
String [] test = new String[3];
String foo = test[index];
```

- A. `foo` 的值为 “”
- B. `foo` 的值为 `null`
- C. 抛出一个异常



D. 程序不能够被编译

答: B

34、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

答: AD

35、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

答: B

36、int[][] myArray=new int[][]{new int[]{5,6,2},new int[]{6,9,7,8,3},new int[]{3,2}};  
myArray[2][2]的值是 ( )

A. 9

B. 2

C. 6

D. 越界

答: D

37、下列描述中, 那些符合 Java 语言的特征 (多选)

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

答: ABD

38、关于异常(Exception)，下列描述正确的是（多选）

- 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

答：ABC

39、(单选)声明一个委托 public int myCallBack(int x); 则用该委托产生的回调方法的原型应该是

- A. void myCallBack(int x)
- B. int receive(int num)
- C. String receive(int x)
- D. 不确定的

答：B

40、(单选)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

答：C

41、class Class1

```
{
public static int Count = 0;

public Class1()
{
Count++;
}
}

public static void main(String args[]){
```

```
Class1 o1 = new Class1();  
Class1 o2 = new Class1();  
}
```

请问，Class1.Count 的值是多少? ( )

- A. 1
- B. 2
- C. 3
- D. 4

答: B

42、三种字符串判空串方法:

- 1: bool isEmpty = (str.length() == 0);
- 2: bool isEmpty = (str == String.Empty);
- 3: bool isEmpty = (str == "");

哪种方法正确?

- A. 1
- B. 2
- C. 3

答: A

43、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.

答: CD

```
44、 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;

答: AD

45、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。

答: D

46、给出下面代码段 (多选)

- 1、switch (m)
- 2、{ case 0 : System.out.println("case 0");
- 1. case 1: System.out.println("case 1"); break;
- 2. case 2:
- 3. default : System.out.println("default");
- 4. }

下列 m 的哪些值将引起 “default ” 的输出?

- A. 0    B. 1    C. 2    D. 3

答: CD

47、下列哪些说法是正确的?

- A. Java 语言只允许单一继承
- B. Java 语言只允许实现一个接口
- C. Java 语言不允许同时继承一个类并实现一个接口
- D. Java 语言的单一继承使得代码更加可靠.

答: AD

48、main 方法的返回类型是:

- A. int
- B. void
- C. boolean
- D. static
- E. public

答: B

49、给定下列代码:

```
public void test() {
    try {
        oneMethod();
        System.out.println( "condition 1");
    } catch (ArrayIndexOutOfBoundsException e){
        System.out.println("condition 2");
    }catch(Exception e){
        System.out.println("condition 3");
    }finally{
        System.out.println("finally");
    }
}
```

```

    }
    oneMethod 正常运行, 将显示?
    A. condition 1
    B. condition 2
    C. condition 3
    D. finally

```

答: AD

50、下列代码哪几行会出错

```

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

```

A. line 4      B. line 6      C. line 7      D. line 8

答: C

51、对于下列代码

```

public class Parent {
    public int addValue (int a ,int b ){
        int s;
        s = a + b;
        return s;
    }
}
class Child extends Parent {
}

```

下述哪些方法可以加入类 Child?

- A. int addValue(int a ,int b){//do something...}
- B. public void addValue(){//do something...}
- C. public int addValue(int a ){//do something...}
- D. public int addValue( int a, int b )throws MyException {//do something...}

答: BC

52、请选出下面两个类间关系描述错误的项

- a. public interface A {}  
public class B extends A {}
- b. public interface A {}  
public interface B extends A {}

```
c. public interface A {}
public class B implements A {}
d. public class A {}
public class B { protected A a;}
```

答: A

53、下面正确的是

```
A) float f = 3.14;
B) byte i = 225;
C) long k = 33;
D) int p[ ][ ];
```

答: CD

54、下列创建数组的方法正确的是

```
A) int two Dim[ ][ ] = new int[4][ ];
B) int two Dim[ ][ ] = new int[ ][4];
C) int two Dim[ ][ ] = new int[4][5];
D) char a[ ] = { 'A' , 'B' };
E) char c[ ] = "ABC" ;
```

答: ACD

55、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

答: C

56、public class X extends Thread implements Runnable{

```
    public void run(){
        System.out.println("this is run()");
    }
    public static void main(String args[]) {
        Thread t=new Thread(new X());
        t.start();
    }
}
```

- A 第一行会产生编译错误
- B 第六行会产生编译错误
- C 第六行会产生运行错误
- D 程序会运行和启动

答: D

57、根据线程安全的相关知识，分析以下代码，当调用 test 方法时  $i > 10$  时是否会引起死锁？

```
public void test(int i)
{
    lock(this) {
        if (i > 10) {
            i--;
            test(i);
        }
    }
}
```

- A. 会锁死
- B. 不会锁死

答：B

58、欲构造 ArrayList 类得一个实例，此类实现了 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();

答：B

59、Which of the following are methods of the Runnable interface.

- A. run
- B. start
- C. yield
- D. stop

答：A

60、Java 网络程序设计中,下列正确的描述是（多选）

- 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 及其上层协议
- C. Java 网络接口只支持 UDP 及其上层协议
- D. Java 网络接口支持 IP 以上的所有高层协议

答：AD

61、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

答：A

62、选出对于 XML 的评论正确的选项\_\_\_\_\_

- A. XML 是一种语言无关平台无关的自描述语言。
- B. 标签实际上包含了元素和属性两部分。用元素来描述有规律的数据，用属性来描述系统数据。
- C. XML 不可跨平台，不具备扩展性。
- D. XML 文档中的每一个元素都处在一个名字空间中；如果没有指定的名字空间，缺省的名字空间就是和该元素相关联的名字空间。

答：AD

63、(单选)What compiler switch creates an xml file from the xml comments in the files in an assembly?

- A. /text
- B. /doc
- C. /xml
- D. /help

答：C

64、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);    }
    }
}
```

- 答：
1. fr.readLine() 没有这个方法
  2. Sleep(1000) 需要用 Thread.sleep(1000);
  3. 没有关闭 FileReader
  4. 死循环，没有退出的可能

65、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);
```

答：错误, 5 的默认类型为 int。

66、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;
}
```

答：正确



67、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();
    }
}
```

答：错误，返回值不是 long 类型

68、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 ();
    }
}
```

答：abxyxy

69、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);
            }
    }
}
```

Copyright Tarena Corporation, 2009. All rights reserved

```

    }
}

```

答: Value for i=1Value for j=1  
Value for i=2Value for j=1

70、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. }

```

答: 删除第 7 行，将第 5 行改为 `String s3 = ""`;

71、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.println("myStr =" + s2 + myStr);
    }
}

```

答: `java MyClass 1 2 3 4`

72、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);
    }
}

```

答: AB:B

73、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. }
```

答：第 7 行

74、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));
    }
}
```

答：Hello:3.14:10

75、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);
    }
}
```

答：getConnection()

76、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);
    }
}
```

答：26

77、下列程序运行的结果是

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

答：编译出错

78、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);
    }
}

```

答：03040

79、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.println("Invalid URL") ;
        return;
    } catch (Exception e) {
        System.out.println("Exception");
    }
}

```

答：打印 Exception

80、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();`

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`

81、现有  $1 \sim 100$  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);

```

82、写一段程序用冒泡算法对数组 `int[] array` 内的元素有小到大重新排列  
答:

```

package com.tarena;
class BubbleSort{
    public static void main(String[] args)
    {
        int[] arr = {8,4,2,3,5};
        arr = sort(arr);
        for(int a : arr)
        {
            System.out.print(a+"\t");
        }
    }
    public static int[] sort(int[] array) {
        int cnt;
        int n = array.length;
        do{
            cnt = 0;
            for( int i=1; i<n; i++ )
            {
                if( array[i]<array[i-1] )
                {
                    int temp = array[i];
                    array[i] = array[i-1];
                    array[i-1] = temp;
                    cnt++;
                }
            }
            n--;
        }while( cnt!=0 );
        return array;
    }
}

```

83、设计线程一个类 `WorkerThread`，其构造函数接受一个 `message` 字符串作为参数，将该字符串打印到 `console` 上，同时，在 `WorkThread` 的 `main` 函数中启动该线程

答:

```
package com.tarena;
```

```

public class WorkerThread extends Thread {
    private String message;
    public WorkerThread(String message) {
        this.message = message;
    }
    public void run(){
        System.out.println(message);
    }
    public static void main(String[] args) {
        new WorkerThread("hello world!").start();
    }
}

```

84、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?

答: `public static String translate(String str) {`  
`String tempStr = "";`  
`try {`  
`tempStr = new String(str.getBytes("GBK"), "ISO-8859-1");`  
`tempStr = tempStr.trim();`  
`} catch (Exception e) {`  
`System.err.println(e.getMessage());`  
`}`  
`return tempStr;`  
`}`

85、写一个单例

答: `public class Singleton {`  
`private Singleton sin = new Singleton();`  
  
`private Singleton() {`  
`}`  
  
`public static Singleton init() {`  
`return sin;`  
`}`  
`}`

86、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 boolean hasBlank(String str) {
    if(str.endsWith(" ") || str.startsWith(" ")) {
        return false;
    }
}

```



```

    }
    else{
        String[] str=str.split( " " );
        if(strs.length()==1){
            return false;
        }
    }
    return true;
}

@Test
public void testFun() {
    System.out.println(TestString.nullString("test"));
}

```

### 87、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()>=templnt.intValue())
            {
                list.add(j,templnt);
                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();
}
}

```

JAVA 类实现序列化方法是实现 `java.io.Serializable` 接口

Collection 框架中实现比较要实现 `Comparable` 接口和 `Comparator` 接口

88、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

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;
    }
}

```

89、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;
class SplitString
{
    public static String split(String str,int num)
    {
        byte[] strs = str.getBytes();
        if(strs[num-1]<0)
        {
            num=num-1;
        }
        byte[] news = new byte[num];
        System.arraycopy(strs,0,news,0,num);
        return new String(news);
    }
    public static void main(String[] args)
    {
        String str = split("我 ABC", 4);
        System.out.println(str);
        String str2 = split("我 ABC 走 DEF", 6);
        System.out.println(str2);
    }
}

```

90、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:a. 对 value 随机赋值;b. 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。

答：

```

package com.tarena;

```

```

import java.util.Random;
public class TestArray {
    final static char[] digits = { '0', '1', '2', '3', '4', '5', '6', '7', '8',
        '9', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',
        'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y',
        'z' };
    public static void main(String[] args) {
        Random r = new Random();
        int value = r.nextInt();
        System.out.println(value);
        char[] cs = TestArray.test(value, 1);
        int[] is = new int[32];
        for (int i = 0; i < cs.length; i++) {
            is[i]=((int)cs[i])==0? cs[i]:cs[i]-48;
        }
        for (int i = 0; i < is.length; i++) {
            System.out.print(is[i]);
        }
    }

    public static char[] test(int i, int shift) {
        char[] buf = new char[32];
        int charPos = 32;
        int radix = 1 << shift;
        int mask = radix - 1;
        do {
            buf[--charPos] = digits[i & mask];
            i >>= shift;
        } while (i != 0);
        return buf;
    }
}

```

91、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]);
}
}
}

```

91、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*。

92、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[0];
            tail--;
            return tmp;
        } else {
            System.err.println("队列为空！");
            throw new RuntimeException();
        }
    }
}
```

### 93、简述 JAVA 代码安全性

答：类加载器（class loader）加载程序运行所需要的所有类，它通过区分本机文件系统的类和网络系统导入的类增加安全性，这可以限制任何的特洛伊木马程序，因为本机类总是先被加载，一旦所有的类被加载完，执行文件的内存划分就固定了，在这个时候特定的内存地址被分配给对应的符号引用，查找表（lookuo table）也被建立，由于内存划分发生在运行时，解释器在受限制的代码区增加

保护防止未授权的访问；然后字节码校验器 (byte code verifier) 进行校验，主要执行下面的检查：类符合 JVM 规范的类文件格式，没有违反访问限制，代码没有造成堆栈的上溢或者下溢，所有操作代码的参数类型都是正确的，没有非法的数据类型转换（例如将整型数转换成对象类型）发生；校验通过的字节码被解释器 (interpreter) 执行，解释器在必要时通过运行时系统执行对底层硬件的合适调用。

#### 94、简述 java 中的继承

答：Java 中是单继承关系，也就是说一个类只能有一个父类，但是可以同时实现多个接口；对于接口本身而言，可以继承多个接口。这样做的目的就是为了阻止因为多继承而造成的业务对象本身的可读性的降低。

#### 95、简述 java 中的垃圾回收

答：GC 是垃圾收集的意思 (Garbage Collection)，内存处理是编程人员容易出现问题的地方，忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃，Java 提供的 GC 功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的，Java 语言没有提供释放已分配内存的显示操作方法。Java 程序员不用担心内存管理，因为垃圾收集器会自动进行管理。要请求垃圾收集，可以调用下面的方法：`System.gc()` 或 `Runtime.getRuntime().gc()`。

#### 5、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()`；。

#### 96、`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。

#### 97、abstract class 和 interface 有什么区别？

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为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。接口 (interface) 是抽象类的变体。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，而接口表示的是多继承的关系；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。

98、String, StringBuffer 与 StringBuilder 的区别。

答：String 是不可变的；StringBuffer 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，并且它是线程安全的；StringBuilder 是从 JDK 5 开始提供的类，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。

99、谈谈 final, finally, finalize 的区别

答：**final**—修饰符（关键字）如果一个类被声明为 **final**，意味着它不能再派生出新的子类，不能作为父类被继承。因此一个类不能既被声明为 **abstract** 的，又被声明为 **final** 的。将变量或方法声明为 **final**，可以保证它们在使用中不被改变。被声明为 **final** 的变量必须在声明时给定初值，而在以后的引用中只能读取，不可修改。被声明为 **final** 的方法也同样只能使用，不能重载

**finally**—再异常处理时提供 **finally** 块来执行任何清除操作。如果抛出一个异常，那么相匹配的 **catch** 子句就会执行，然后控制就会进入 **finally** 块（如果有的话）

**finalize**—方法名。Java 技术允许使用 **finalize()** 方法在垃圾收集器将对象从内存中清除出去之前做必要的清理工作。这个方法是由垃圾收集器在确定这个对象没有被引用时对这个对象调用的。它是在 **Object** 类中定义的，因此所有的类都继承了它。子类覆盖 **finalize()** 方法以整理系统资源或者执行其他清理工作。**finalize()** 方法是在垃圾收集器删除对象之前对这个对象调用的。

100、屋子里面有椅子，椅子有四只腿及靠背。屋子、椅子什么关系，椅子、腿、靠背什么关系 (HAS-A, IS-A)?

答：屋子 HAS-A 椅子

椅子 HAS-A 四只腿及靠背

101、你认为 java 与其他（你所了解的）语言相比，有什么优点和缺点？

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 struct 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。

102、Java 中包含哪些基本数据类型? String 是最基本的数据类型吗? 是否可以继承 String 类?

答: 基本类型包括 byte、int、short、long、float、double、boolean

String 不是基本数据类型, 因为 String 类是 final 的, 所以不能够被继承。

103、Math.round(11.5) 等于多少? Math.round(-11.5) 等于多少?

答: Math.round(11.5)==12 Math.round(-11.5)==-11 round 方法返回与参数最接近的长整数, 参数加 1/2 后求其 floor。

104、Bit 和 Byte 是什么意思? 它们之间有什么关系?

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

105、用最有效的方法算出 2 乘已 8 对于几?

答:  $2 \ll 3$ 。

106、String s=new Stirng("xyz"); 创建了几个 String Object?

答: 创建两个, 一个在堆内存 (通过 new 得到), 另一个在字符串常量池 ("xyz") 对象。

107、swtich 是否能作用在 byte 上? 是否能作用在 long 上? 是否能作用在 char 上? 是否能作用在 short 上? 是否能作用在 String 上?

答: 可以用在 byte、short、int、char 和枚举类型。

108、char 型变量中能不能存贮一个中文汉字? 为什么?

答: 能够定义成为一个中文的, 因为 java 中以 unicode 编码, 一个 char 占 16 个字节, 所以放一个中文是没问题的。

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

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。

110、说一下 “==” 和 equals() 方法在字符串变量操作中的不同?

答: “==” 比较的是两个字符串对象的地址, equals() 是比较的两个字符串的具体值。

111、异常机制中 try {} 后一定要跟 catch 和 finally 吗?

答: 一定要有 catch 或者 finally, catch 子句紧跟在 try 块后面, 用来指定你想要捕捉的“异常”的类型。如果有 catch 的话, 可以没有 finally。

112、能用 foreach 遍历访问的对象需要实现?

答: 需要实现 Iterable 接口

113、说出一些 JDK 常用的类, 包, 接口, 请各举 5 个

类:

HashSet、Thread、String、FileInputStream、Class

包:

java.lang、java.io、java.util、java.text、java.sql

接口:

Collection、Set、List、Comparable、Serializable

114、Anonymous 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)?

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 method...}
});

```

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。

115、Java 的接口和 C++的虚类的相同和不同处。

答：由于 Java 不支持多继承，而有可能某个类或对象要使用分别在几个类或对象里面的方法或属性，现有的单继承机制就不能满足要求。与继承相比，接口有更高的灵活性，因为接口中没有任何实现代码。当一个类实现了接口以后，该类要实现接口里面所有的方法和属性，并且接口里面的属性在默认状态下面都是 `public static`，所有方法默认情况下是 `public`。一个类可以实现多个接口。

116、JAVA 中数组有没有 `length()`这个方法？String 有没有 `length()`这个方法？

答：数组是没有 `length()`方法的，数组是使用数组对象 `.length` 来得到该数组的长度；String 是有 `length()`方法的。

117、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？

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。

118、java 中用 `char` 型变量中能不能存放一个中文汉字？

答：是能够定义成为一个中文的，因为 java 中以 `unicode` 编码，一个 `char` 占 16 个字节，所以放一个中文字符是没问题的。

119、面向对象的特征有哪些方面？

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。

120、什么是 java 序列化，如何实现 java 序列化？

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。

121、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？

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。

122、什么是 JAVA 的反射？

答：程序集包含模块，而模块包含类型，类型又包含成员。反射则提供了封装程序集、模块和类型的对象。可以使用反射动态地创建类型的实例，将类型绑定到现有对象，或从现有对象中获取类型。然后，可以调用类型的方法或访问其字段和属性。

123、构造器 Constructor 是否可以被继承？是否可以被 Override？

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

124、UDP 和 TCP 连接有和异同？

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。

125、程序注释的用途有哪些？带注释的程序有什么缺点？

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。

126、在程序中退出两重循环可以用哪些语句？

答：可以使用标签：

```
loop1: for (;;) {
    for (;;) {
        break loop1;
    }
}
```

127、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？

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。

**128、Error 和 Exception 有什么区别？**列出你见过的 Exception 并简要说明。

**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。

常见异常有：

**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。

**129、java 中会存在内存泄露吗？**请简单描述。

**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。

**130、Java AWT 或 Swing 中的常用控件包括：**

**答：**JTable, BorderLayout, GridLayout, FlowLayout, JFrame, JLabel。

**86、多线程有几种实现方法,都是什么?同步有几种实现方法,都是什么?**

**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){...}

**131、运行时异常与一般异常有何异同?**

**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。

132、sleep() 和 wait() 有什么区别？

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。

133、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

答：如果数据将在线程间共享。例如正在写的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。

134、进程和线程分别该怎么理解？

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。

135、死锁的必要条件？怎么克服？

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，在未使用完之前，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

### 136、谈谈对 ClassLoader 的理解？

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。

### 137、详细阐述一下 Collection 接口所包含的内容？

答：List:有序允许重复存放

ArrayList----按顺序存放数据的数组

LinkedList-----按顺序存放数据的链表

Vector-----线程安全的按顺序存放数据的数组

Set:无序不允许重复存放

HashSet---根据 hashCode() 和 equals() 方法来判断是否有重复

SortedSet: 有序

TreeSet-----通过实现 Comparable 接口和 Comparator 接口而具有排序功能的集合

### 138、请说出集合类中 List、Map、Set 的区别

答：List 和 Set 继承了 Collection 接口，而 map 不是；List 中存放元素有顺序并且可以重复；set 中存放的元素是无序并且是不可能重复的；Map 中存放是键值对。

### 139、请说出你所知道的线程同步的方法。

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;

notifyAll():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。

### 140、Collection 和 Collections 的区别？

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。

### 141、HashMap 与 TreeMap 的区别？

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)。

#### 142、ArrayList 和 Vector 的区别？

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 默认增长为原来一倍，而 ArrayList 却是原来的一半。

#### 143、HashMap 和 Hashtable 的区别？

答：HashMap 是 Hashtable 的轻量级实现（非线程安全的实现），他们都完成了 Map 接口，主要区别在于 HashMap 允许空（null）键值（key），由于非线程安全，效率上可能高于 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，Hashtable 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 就必须为之提供外同步。

#### 144、OOAD 中的 Association、Aggregation、Composition 的区别？

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。

#### 145、j2ee 常用的设计模式？说明工厂模式？

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（组合模式），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（迭代器模式），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 Responsibility (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。

#### 146、请说出 ArrayList, Vector, LinkedList 的存储性能和特性

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。

#### 147、JAVA 中常用的 XML 解析技术有哪些? 区别是什么?

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 的树形结构所造成的, 该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。

#### 148、XML 文档定义有几种形式? 有何本质区别?

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 Schem 允许把文件类型定义为整数, 浮点数, 字符串, 布尔值或其他各数据类型, 而无须重新定义。

## JAVAAEE 部分

1、O/R 可以实现\_\_\_\_\_

- A. 类到数据表的映射
- B. 数据查询
- C. 数据恢复
- D. 数据集成

答：A

2、(单选)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

答：A

3、(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

答：A

4、(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

答：A

5、(单选)单元测试一般在什么时候进行？

- A. 编码完成后
- B. 系统测试前
- C. 测试用例编写完成后
- D. 集成测试后

答：A

6、(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

答: ABCD

7、Data Sources 与 DriverManager 都可获得 Connection 对象, 有何区别\_\_\_\_?

- A. Data Sources 是同步的, DriverManager 是异步的。
- B. Data Sources 是异步的, DriverManager 是同步的。
- C. Data Sources 必须从 Context 对象查找 JNDI 名获得 Connection 对象, DriverManager 只需提供链接字符串, 用户名, 登录密码即可得到 Connection 对象。
- D. Data Sources 与 DriverManager 没有不同, 只是一个东西的两种叫法。

答: C

8、以下是关于 HttpServlet 类中 doGet()、doPost()、doDelete()、doPut()、service() 方法的描述。

- A. HttpServlet 中没有 doDelete() 这个方法。
- B. doGet()、doPost()、service() 这些方法都不是线程安全的。
- C. 客户端来的 get 请求将调用 doGet(), 客户端来的 post 请求将调用 doPost() 方法。客户端来的所有请求都调用 service() 方法。
- D. 以上说法都不对。

答: BC

9、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。

答: ABC

10、在以下哪些情况下 session 会话将结束生命周期

- A. 客户端关闭浏览器。
- B. 会话超时。
- C. 服务器调用了 HttpSession 的 invalidate() 方法。
- D. 客户端没有访问本应用而连接到别的应用上。

答: ABC

11、由 a.jsp 传递一个参数: "我是竹子" 到 b.jsp。a.jsp 编码为 UTF-8, b.jsp 编码为 gb2312 会出现何种情况\_\_\_\_?

- A. 正常传递, 无任何异常情况。
- B. 无法传递中文信息。报错。
- C. 可以传递。但 b.jsp 接到的参数为乱码。
- D. 编译不通过, 提示非法参数或语法错误。

答: C

12、以下是用户请求访问 Servlet 的过程的描述, 哪些是错误的:

- A. servlet 容器将创建一个 ServletRequest 对象和 ServletResponse 对象。

B. 在 `ServletResponse` 对象中封装了用户请求信息，然后 `ServletServletResponse` 对象和 `ServletRequest` 对象传给用户所请求的 `Servlet`。

C. `Servlet` 把响应结果写到 `ServletRequest` 中，然后由 `Servlet` 容器把响应结果传给用户。

D. 以上都没错

答: BCD

13、What should be inserted on line 3 to allow compilation to succeed, and allow the output of "Set it to: 94"?

1. `<html>`

2. `<body>`

3.

4. `<% x = x + 50; %>`

5. `Set it to: <%= x %>`

6. `</body>`

7. `</html>`

A. `<% int x = 44 %>`      B. `<%@ int x = 44 %>`      C. `<%! int x = 44; %>`

D. `<%! int x = 44; !%>`      E. `<%@ int x = 44; @%>`

答: C

14、Given the `HttpServlet` method:

`doGet(HttpServletRequest, HttpServletResponse)`

Which statement is true?

A. It returns servlet version information.

B. It returns servlet context information.

C. It services an HTTP GET request.

D. It services an HTTP OPTIONS request.

答: C

15、关于 `Cookie` 和 `HttpSession` 的说法错误的是:

A. 缺省的 `Session` 实现机制采用的是 `Cookie` 技术

B. 通常以一个 `PC` 机作为同一个 `Session` 的衡量标志

C. 如果浏览器的安全级别设置的非常高, `Cookie` 可能就用不了了

D. 当 `Cookie` 不能用时, `Session` 就肯定用不了了

答: D

16、下列 `EJB2.0` 中描述正确的有:

A. 远程接口继承自 `EJBObject`, 远程接口只是规定了客户机与 `EJB` 间的约定。

B. 本地接口继承自 `EJBHome`, 本地接口提供创建、查找、删除 `EJB` 实例。

C. 远程接口与本地接口均是方法定义没有方法实现。

D. 远程接口与本地接口均有方法实现。

答: BC

17、实现 `JTA` 说法错误的有:

- A. 需要实现 Transaction 接口。
- B. 需要 Context 对象 lookup "javax.transaction.UserTransaction"。
- C. 需要 Context 对象 lookup "javax.transaction.Transaction"。
- D. 必须有相应方法开始事务和终止事务。

答: C

18、J2EE 规范只认可的事务性资源类型有:

- A. JDBC 数据库
- B. JMS 消息队列服务器
- C. 其它通过 JCA 访问的事务性服务
- D. JCX 事务

答: ABC

19、关于 EJB 的 Home 接口和 Remote 接口, 描述错误的是:

- A. Remote 接口定义了业务方法, 用于 EJB 客户端调用业务方法
- B. Home 接口是 EJB 工厂, 用于创建、移除、查找 EJB 实例
- C. JNDI 树上绑定的是 Remote 接口
- D. JNDI 树上绑定的是 Home 接口

答: D

20、关于 EJB 的描述错误的是:

- A. SessionBean 分为 Stateful 和 Stateless 两种
- B. EntityBean 分为 BMP 和 CMP 两种
- C. Message-Driven Bean 相当于 JMS 应用中的消息生产者
- D. 假设 EJB 由容器管理事务, 在 EJB 方法中需要会滚的话执行如下代码:  
ctx.getUserTransaction().rollback(); (假设 ctx 为该 EJB 的 Context)

答: CD

21、在 JSP 页面中定义一个方法, 如下所示, 选择一个正确的答:

```
...
<%!
void doPrint() {
    out.println( "hello jsp" );
}
%>
...
```

- A. 编译没有问题
- B. 编译和调用都没有问题
- C. 编译出错
- D. 运行出错

答: C

22、找出下面程序中可能引起异常的代码行, 并说明原因。

```
// 登录编号, 长度在 0-38 间
String admin_id = session.getAttribute( "admin_id" ).toString();
```

```

// 登录者权限
String admin_right = request.getParameter( "admin_right" );
// 登录者号码
String admin_no = request.getParameter( "admin_no" );
if (admin_right.equals( "" ))
    out.println( "对不起，没有授权" );
// 登录者特别号
int admin_no = Integer.parseInt(admin_no);
// 所在组
String admin_group = admin_id.substring(38);

```

请将上段程序按照您的习惯重新写一下。

答：

- 在 `if (admin_right.equals( "" ))` 这行可能引起空指针异常，如果登录者权限的值为空时，会导致 `request.getParameter( "admin_right" )` 得到是一个空值，如果再使用这个变量时，就有可能抛出空指针异常。
- 当登录号码的值为空时，`Integer.parseInt(admin_no)` 也可能会抛出空指针异常；
- 当登录编号的值为空值时，`String admin_group=admin_id.substring(38);` 也会抛出空指针异常，
- 在 `int admin_no = Integer.parseInt(admin_no);` 这行也定义了一个重名的变量 `admin_no`

重写后：

```

// 登录编号, 长度在 0-38 间
String admin_id = session.getAttribute( "admin_id" ).toString();
// 登录者权限
String admin_right = request.getParameter( "admin_right" );
// 登录者号码
String admin_no = request.getParameter( "admin_no" );
// 登录者特别号
int admin_num = 0;
// 所在组
String admin_group = null;

if ( "" .equals(admin_right)){
    out.println( "对不起，没有授权" );
}
if(admin_no!=null){
    admin_num = Integer.parseInt(admin_no);
}
if(admin_id!=null){
    admin_group = admin_id.substring(38);
}

```

23、写出下面代码的输出结果（alert 的输出）

```
function funTouch() {
var a = 10;
var b = 20;
var c = 10;
alert(a = b);
alert(a == b);
alert(a == c);
}
```

答: 20 true false

24、检查身份证号码正确性长度 15 或 18 位

答: var regx=/^[0-9]{15,18}\$/;

25、前 14 位一般为数字末位可以为字母

答: var regx=/^\d{14}(\d|X)\$/;

26、身份证号码中带有出生日期，请编写取出出生日期的函数，格式为 YYYY-MM-DD。

```
答: function getBirthday()
{
var str = "110101195403297654";//提取 19540329
var dest = str.substring(6,10) + "-" + str.substring(10,12) + "-" +
str.substring(12,14);
return dest;
}
```

27、完成以下操作

(1) 写出打开和关闭窗口方法

(2) 页面有一个下拉菜单和一个输入框，写一方法使下拉菜单值变动时输入框显示对应的值

(3) 写一方法校验多选框有多少被勾选

答:

1. window.open 和 window.close

2.

```
<html>
<head>
<script type="text/javascript">
function show(f) {
document.getElementById('ss').value=f;
}
</script>
</head>
<body>
<select id="s" onchange="show(this.value);">
<option id="one" value="one">
```

```

        one
    </option>
    <option id="two" value="two">
        two
    </option>
    <option id="three" value="three">
        three
    </option>
</select>
<input type="text" id="ss" >
</body>
</html>

```

3.

```

function isChecked(v) {
    var checkboxid=document.getElementById("checkboxid");
    var count=0;
    for(var i=0;i<checkboxid.length;i++) {
        if(checkboxid[i].checked==true) {
            count++;
        }
    }
    return count;
}

```

28、对于 Jscript, 对下面片断, 解释每一行的用处

```

1. var news=getnews();
2. arr=news.split("@");
3. len=arr.length;

```

答:

1. 把 getnews() 方法得到的返回值赋给变量 news
2. 把 news 这个字符串按@拆分的得到的数组赋给 arr
3. 把数组 arr 的长度赋给 len 变量

29、使用 html 实现如下的表格:

编码	姓名	时间
001	张三	2009-1-1
002	李四	2009-2-1

答:

```

<table border="1">
<tr>
<td>编码</td>
<td>姓名</td>
<td>时间</td>
</tr>
<tr>

```



```

<td>001</td>
<td>张三</td>
<td>2009-1-1</td>
</tr>
<tr>
<td>002</td>
<td>李四</td>
<td>2009-2-1</td>
</tr>
</table>

```

30、javascript 常用的方面:

答: 常用于数据输入校验、页面特殊效果、AJAX 等。

31、Statement, PreparedStatement, CallableStatement 的区别

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

32、运行 ant 需要设置的环境变量有?

答: ANT\_HOME、JAVA\_HOME、PATH。

33、解释四种会话跟踪技术?

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

34、J2EE、EJB、JDBC 是一回事吗, 它们之间有什么关系?

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。

35、J2EE 是什么？它包括哪些技术？

答：从整体上讲，J2EE 是使用 Java 技术开发企业级应用的一种事实上的工业标准，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：

- 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。

- 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。

- 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。

- 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 相当与 Microsoft 平台中的 ODBC (Open Database Connectivity)。

36、数据连接池的工作机制是什么？

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标记为忙。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标记为空闲，其他调用就可以使用这个连接。

37、什么是 ORM？

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。

38、Hibernate 有哪 5 个核心接口？

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。

39、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，默认为自动提交事务，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。

40、说出下列配置文件是何含义：

```
<bean id="testLocal"  
class="org.springframework.ejb.access.LocalStatelessSessionProxyFactoryBean">  
<property name="jndiName" value="Test"/>  
<property name="businessInterface"  
value="net.greatsoft.cms.prv.test.TestLocal"/>  
</bean>
```

答：

定义一个 bean

对应的 2 个初始化参数 String jndiName 值为 test

String businessInterface 值为 net.greatsoft.cms.prv.test.TestLocal

41、请写出一段利用 JDBC 连接 oracle 数据库并执行

答：

```

public void testJdbc() {
    Connection con = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try{
        //step1: 注册驱动;
        Class.forName("oracle.jdbc.driver.OracleDriver");
        //step 2: 获取数据库连接;
        con=DriverManager.getConnection(
            "jdbc:oracle:thin:@192.168.0.23:1521:tarena",
            "openlab","open123");
        /*****查询*****/
        //step 3: 创建 Statement;
        String sql = "SELECT id, fname, lname, age, FROM Person_Tbl";
        ps = con.prepareStatement(sql);
        //step 4 : 执行查询语句, 获取结果集;
        rs = ps.executeQuery();
        //step 5: 处理结果集—输出结果集中保存的查询结果;
        while (rs.next()) {
            System.out.print("id = " + rs.getLong("id"));
            System.out.print(", fname = " + rs.getString("fname"));
            System.out.print(", lname = " + rs.getString("lname"));
            System.out.print(", age = " + rs.getInt("age"));
        }
        /*****JDBC 修改*****/
        sql = "UPDATE Person_Tbl SET age=23 WHERE id = ?";
        ps = con.prepareStatement(sql);
        ps.setLong(1, 88);
        int rows = ps.executeUpdate();
        System.out.println(rows + " rows affected.");
    } catch (Exception e) {
        e.printStackTrace();
    } finally{
        try{
            con.close(); //关闭数据库连接, 以释放资源。
        } catch (Exception e1) {
        }
    }
}
}

```

42、用你熟悉的语言写一个连接 ORACLE 数据库的程序,能够完成修改和查询工作。

答: JDBC 示例程序如下:

```
public void testJdbc() {
    Connection con = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try{
        //step1: 注册驱动;
        Class.forName("oracle.jdbc.driver.OracleDriver");
        //step 2: 获取数据库连接;
        con=DriverManager.getConnection(
            "jdbc:oracle:thin:@192.168.0.23:1521:tarena",
            "openlab", "open123");
        /*****查询*****/
        //step 3: 创建 Statement;
        String sql = "SELECT id, fname, lname, age, FROM Person_Tbl";
        ps = con.prepareStatement(sql);
        //step 4 : 执行查询语句, 获取结果集;
        rs = ps.executeQuery();
        //step 5: 处理结果集—输出结果集中保存的查询结果;
        while (rs.next()) {
            System.out.print("id = " + rs.getLong("id"));
            System.out.print(", fname = " + rs.getString("fname"));
            System.out.print(", lname = " + rs.getString("lname"));
            System.out.print(", age = " + rs.getInt("age"));
        }
        /*****JDBC 修改*****/
        sql = "UPDATE Person_Tbl SET age=23 WHERE id = ?";
        ps = con.prepareStatement(sql);
        ps.setLong(1, 88);
        int rows = ps.executeUpdate();
        System.out.println(rows + " rows affected.");
    } catch (Exception e) {
        e.printStackTrace();
    } finally{
        try{
            con.close(); //关闭数据库连接, 以释放资源。
        } catch (Exception e1) {
        }
    }
}
```

43、从数据库中取出 100,000 条数据，每次插入数据库 1000 条，请写出 java 相关处理程序，其中 insertAll(ArrayList 或 Vector) 为公共方法，已封装好从数据库取出的 100,000 条数据；要求实现功能：批量插入数据

答：

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Iterator;
import java.util.List;

public class InsertBatch {
    private static Connection conn;

    static {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/test", "root", "1234");
    }

    public static boolean insertAll(List<?> result) {
        Iterator<?> it = result.iterator();
        int i = 0;
        try {
            PreparedStatement ps = conn
                .prepareStatement("insert into table values(?)");

            while (it.hasNext()) {
                ++i;
                ps.setString(0, it.next()+"");
                if (i % 1000 == 0) {
                    ps.execute();
                    ps.clearBatch();
                }
                ps.addBatch();
            }
            return true;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
}
```

44、Hibernate 实现中，load 和 get 方法的区别，find 和 iterate 的区别？

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。

45、Tomcat 缺省端口号是多少，如果需要修改哪个目录下的哪个文件？

答：8080，/conf/server.xml

46、介绍一下 hibernate 的工作原理、优点以及如何优化？

答：

工作原理：

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();

b. 加载 hibernate.cfg.xml 文件至该实例内存，通过 hibernate.xfg.xml 文件中的 mapping 节点配置，加载 hbm.xml 文件至该实例内存；

c. 由上面得到的 Configuration 实例构建一个 SessionFactory 实例  
SessionFactory sf = cfg.buildSessionFactory();

d. 由上面得到的 SessionFactory 实例创建连接 Session s = sf.openSession();

e. 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();

f. 通过 Session 接口提供的各种方法操作数据库的访问

g. 提交数据库的操作结果 tx.commit();

h. 关闭 Session 链接 s.close();

优点：

a. 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。

b. 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

c. hibernate 使用 Java 反射机制，而不是字节码增强程序来实现透明性。

d. 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：

a. 使用双向一对多关联，不使用单向一对多

b. 灵活使用单向一对多关联

c. 不用一对一，用多对一取代

- d. 配置对象缓存, 不使用集合缓存
- e. 一对多集合使用 Bag, 多对多集合使用 Set
- f. 继承类使用显式多态
- g. 表字段要少, 表关联不要怕多, 可以利用二级缓存

47、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)

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。

48、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?

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 是怎么知道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。



49、说出 Servlet 的生命周期，并说出 Servlet 和 CGI 的区别？

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。

50、WEB SERVICE 名词解释。JSWDL 开发包的介绍。JAXP、JAXM 的解释。SOAP、UDDI, WSDL 解释。

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；

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信息进行操作。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）；

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；

UDDI 的目的是为电子商务建立标准；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。

51、Struts 的控制器部分，视图部分包括哪些内容？

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。

52、实际应用中，servelet 和 JSP 如何使用？

答：servlet 一般用来做控制流转，调用业务层方法，响应请求。jsp 用来做页面中数据的收集和表现。

53、jsp 有哪些动作？作用分别是什么？

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

54、MVC 的各个部分都有哪些技术来实现？如何实现？

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 server 或是 struts 中的 action）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。

55、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。

56、JSP 中动态 INCLUDE 与静态 INCLUDE 的区别？Servlet 中 forward 与 redirect 的区别？

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" %>`

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，所以 session, request 参数都可以获取，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。

57、说出 JSP 的内置对象，并给出说明。

答：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。

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）。

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。

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。

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。

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

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。

page 表示从该页面产生的一个 servlet 实例。

exception 针对错误网页，未捕捉的例外

58、简述 struts 是怎样实现的 MVC 模型。

答：

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动作映射对象，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。

59、struts 中 actionform 和 action 属于 MVC 哪一层，为什么？actionform 的功能有哪些？

答：actionform 属于 MVC 当中的 V 层，因为 M 层应该是封装了业务逻辑的对象，而 ActionForm 仅仅是一个 form-bean，封装了用户提交的表单数据（物理 View），

可以在其中进行一些非业务逻辑的验证，并没有真正的映射到模型数据，因为与层的关系比较密切，所以实际上它应该算是 View 层（逻辑 View）；action 应该属于 C 层，仅仅描述“做什么”，与“如何做”（Model）关系不大，把它归于 Controller 层比较合适。

ActionForm 的功能主要是：

- 1) 表单中的信息——>对象的封装
- 2) 表单信息的缓存
- 3) 表单信息验证（服务器端验证）建议用 validator 框架

60、一个 Web 页面包含哪三大要素，都有些什么标准。要解决跨浏览器访问的问题应注意哪些方面。

答：HTML/XHTML/CSS 都是基于 W3C 标准

跨浏览器访问问题主要是因为浏览器对于脚本语言的解析和支持程度不同。要编写跨浏览器的页面，就要注意页面中的标签和脚本语言是否被支持。

61、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);
}
```

答：默认不是线程安全的，但是 servlet 实现了 SingleThreadModel 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。

62、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。

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，订阅。

Podcasting：个人视频/声频的发布/订阅。

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

63、介绍一下 Struts 的工作原理？

答：struts 的工作原理可分为如下 8 步。

1. 读取配置（初始化 ModuleConfig 对象）

Struts 框架总控制器（ActionServlet）是一个 Servlet，在 web.xml 中被配置成一个自动启动的 Servlet。读取配置文件 struts-config.xml 的配置信息，为不同的 Struts 模块初始化相应的 ModuleConfig 对象。

2. 用户请求

用户提交表单或调用 URL 向 WEB 应用程序服务器提交一个请求，请求的数据用 HTTP 协议上传给 WEB 服务器。

3. 填充 FormBean

（\*.do 请求）从 ActionConfig 中找出对应该请求的 Action 子类，如有对应的 Action 且这个 Action 有一个相应的 ActionForm，ActionForm 被实例化并用 HTTP 请求的数据填充其属性，并保存在 request 或 session 范围中，这样他们就可以被其它 Action 对象或 JSP 调用。如果没有对应的 Action，控制器则直接转发给 JSP 或静态页面。

4. 派发请求

控制器根据配置信息 ActionConfig 将请求派发到具体的 Action，相应的 FormBean 一并传给这个 Action 的 execute() 方法。

5. 处理业务

Action 一般只包含一个 execute 方法，它负责执行相应的业务逻辑。执行完毕后返回一个 ActionForward 对象，控制器通过该 ActionForward 对象来进行转发工作。

6. 返回响应

Action 根据业务处理的不同结果返回一个响应对象给总控制器，该目标响应对相对应一个具体的 JSP 页面或另一个 Action。

7. 查找响应

总控制器根据业务功能 Action 返回的目标响应对象找到对应的资源对象，通常是一个具体的 JSP 页面。

8. 响应用户

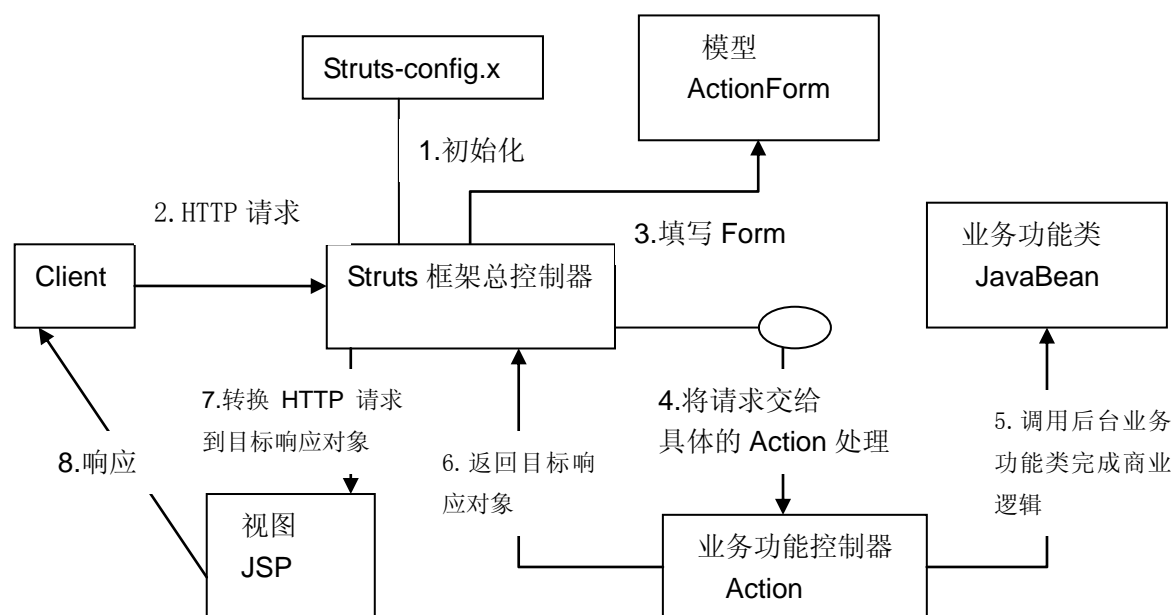
JSP 将结果展现给用户。

64、Jdo 是什么？

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 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。

65、请简单画出 struts 技术构建 mvc 的流程图

答：



#### 66、get 和 post 的区别？

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 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；

2) 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；

3) 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；

4) 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；

5) Get 是 Form 的默认方法。

#### 67、JSP 和 Servlet 有哪些相同点和不同点，他们之间的联系是什么？

答：JSP 是 Servlet 技术的扩展，本质上是 Servlet 的简易方式，更强调应用的外表表达。JSP 编译后是“类 servlet”。Servlet 和 JSP 最主要的不同点在于，Servlet 的应用逻辑是在 Java 文件中，并且完全从表示层中的 HTML 里分离开来。而 JSP 的情况是 Java 和 HTML 可以组合成一个扩展名为 .jsp 的文件。JSP 侧重于视图，Servlet 主要用于控制逻辑。

#### 68、介绍一下 spring MVC 的工作原理、用途、为什么用？

答：

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 将请求提交到目标 Controller
4. Controller 进行业务逻辑处理后, 会返回一个 ModelAndView
5. DispatcherServlet 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。

用途:

JSP、Servlet、JavaBean 技术的出现给我们构建强大的企业应用系统提供了可能。但用这些技术构建的系统非常的繁乱, 所以在此之上, 我们需要一个规则、一个把这些技术组织起来的规则, 这就是框架, Struts 便应运而生。

为什么用:

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。

69、Spring 有几种事务管理, Spring 的事务管理接口是什么?

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。

70、EJB2.0 有哪些内容? 分别用在什么场合?

答: 规范内容包括 Bean 提供者, 应用程序装配者, EJB 容器, EJB 配置工具, EJB 服务提供者, 系统管理员。这里面, EJB 容器是 EJB 之所以能够运行的核心。EJB 容器管理着 EJB 的创建, 撤消, 激活, 去活, 与数据库的连接等等重要的核心工作。

### 71、EJB 与 JAVA BEAN 的区别？

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。

### 72、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

答：

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

与 hibernate 同类型的开源框架有：ibatis、SimpleORM、OBJ、JPOX 等等可选择

与 struts 同类型的开源框架有：struts2 (webwork2)、JSF (以及 myfaces)、tapestry、EasyJWeb、Spring MVC、AppFuse 等

### 73、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

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 and 安全性[1]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。

### 74、简要说明 JVM、JSP、Servlet、Web Server、Web Browser 之间的关系。

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。

### 75、什么是 AJAX 和 AOP?



答: Ajax 的全称是: AsynchronousJavaScript+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之处, 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。

76、写出熟悉的 JSTL 标签。

答: <c:if>、<c:choose>、<c:when>、<c:otherwise>、<c:forEach>、<c:set>。

77、谈谈对 Struts、spring 和 hibernate 的理解

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 this 才能真正理解 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。

78、什么是 B/S 结构, C/S 结构?

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 Browser/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。

79、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？

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。

80、什么是 UDDI、SOAP、WSDL？

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信息进行操作。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)。

81、部署一个 web 应用的步骤是什么？

- 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/>

82、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。

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、部署

Eclipse：IDE 集成开发工具

83、用 JPA 写一个用户表的实体类，表结构如下：

```
CREATE TABLE user(
    id NUMBER(8) primary key,
    userName VARCHAR2(30),          -- 用户名称
    password VARCHAR (20),         -- 密码
```

```
);  
答:  
package com.tarena;  
import java.io.*;  
import java.persistence.*;  
  
@Entity  
@Table(name="user")  
public class user implements Serializable{  
    @Id  
    @Column(name="id")  
    private Integer id;  
    @Column(name="userName")  
    private String username;  
    @Column(name="password")  
    private String password;  
  
    public long getID() {  
        return id;  
    }  
    public void setID(long id) {  
        this.id = id;  
    }  
    public String getUsername() {  
        return username;  
    }  
    public void setUsername(String username) {  
        this.username = username;  
    }  
    public String getPassword() {  
        return password;  
    }  
    public void setPassword() {  
        this.password = password;  
    }  
}
```

84、EJB3.0 有哪几种类型? 写出一个 stateless bean 的本地接口类及实现类, 接口如下 : String hello World();

答: 会话 Bean (Session Bean), 实体 Bean (Entity Bean) 和消息驱动 Bean (MessageDriven Bean)。

```
@Local  
public interface StatelessBean{  
    String helloWorld();  
}
```

```
@Stateless
public class StatelessBeanImpl implements StatelessBean{
    String helloWorld() {
        System.out.println("Hello world!!!");
    }
}
```

## SQL 部分

1、当用下面的 SQL 语句对数据库进行查询时，为什么会出错？

```
SELECT name, sal
FROM emp
WHERE sal =
(SELECT sal
FROM emp
WHERE name='John' OR deptno = 12)
```

- A. 上述查询中不允许子查询
- B. WHERE 语句中不允许出现逻辑运算符
- C. 单行运算符后面的子查询返回多行
- D. 多行运算符后的子查询返回单行

答：C

2、下面查询中哪一个查询返回所有没有下属的员工？

- A. SELECT e.ename FROM emp e WHERE e.mgr IS NOT NULL;
- B. SELECT e.ename FROM emp e WHERE e.empno IN (SELECT m.mgr FROM emp m)
- C. SELECT e.ename FROM emp e WHERE e.empno NOT IN (SELECT m.mgr FROM emp m)
- D. SELECT e.ename FROM emp e WHERE e.empno IN (SELECT m.mgr FROM emp m WHERE m.mgr IS NOT NULL)

答：D

3、在 PL/SQL 块的变量声明部分声明的 number 类型的变量，如果没有给出变量的初始值，那么在 PL/SQL 块执行时，该变量的被初始化为：

- A. 0
- B. Null
- C. 变量的初始值取决于变量定义中指定的长度和精度
- D. 由于变量没有初始化，PL/SQL 块将无法执行

答：B

4、在执行下面 SQL 时 DELETE FROM dept WHERE dept\_id = 12; 出现 integrated constraint error because the child record was found 错误,如何才能使这条 SQL 语句执行成功?

- A. 执行之前, 删除 child record
- B. 无法使该命令执行成功
- C. 在该命令上添加另外的关键字
- D. 在命令中添加 cascade constraints 选项

答：AD

5、Which statement shows the maximum salary paid in each job category of each department?\_\_\_\_\_

- A. select dept\_id, job\_cat, max(salary) from employees where salary > max(salary);

- B. select dept\_id, job\_cat, max(salary) from employees group by dept\_id, job\_cat;  
 C. select dept\_id, job\_cat, max(salary) from employees;  
 D. select dept\_id, job\_cat, max(salary) from employees group by dept\_id;  
 E. select dept\_id, job\_cat, max(salary) from employees group by dept\_id, job\_cat, salary;

答: B

6、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)  
 F. maximum(start\_date)

答: CE

7、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

答: D

8、已知关系框架 R(A, B, C, D, E) 及其上的函数相关性集合,  $F = \{A \rightarrow B, E \rightarrow A, CE \rightarrow D\}$ , 该关系框架的候选关键字是

- A . AE B. BE C. CE D. DE

答: C

9、设有关系框架 R(A, B, C, D, E) 及其上的函数相关性集合,  $F = \{B \rightarrow A, BC \rightarrow D\}$ , 该关系框架 R 最高是

- A 第一范式的 B 第二范式的 C 第三范式的 DBCDF 范式的

答: B

10、关系数据库中，主键是

- A. 为标识表中唯一的实体 B. 创建唯一的索引，允许空值  
 C. 只允许以表中第一字段建立 D. 允许有多个主键的

答: A

11、当运用 SQL 语句创建主键时，可以是

- A. create table table  
     (column1 char(13) not null primary,  
     Column2 int not) on primary ;  
 B. alter table table with notcheck add

```
constraint [PK_ table1] primary key nonclustered
(column1) on primary ;
```

C. alter table table add primary key column;

答: C

12、为数据库创建索引的目的是 (\_\_\_)

- A. 提高查询的检索功能      B. 创建唯一索引  
C. 创建主键                      D. 归类

答: A

13、在 SQL 语法中, SELECT 语句的完整语法较复杂,但至少包括的部分 (\_\_\_),使用关键字 (\_\_\_) 可以把重复行屏蔽,将多个查询结果返回一个结果集合的运算符是 (\_\_\_),如果在 SELECT 语句中使用集合函数时,一定在后面使用 (\_\_\_)。

- (1) A. SELECT, INTO      B. SELECT, FROM  
    C. SELECT, GROUP      D. 仅 SELECT  
(2) A. DISTINCT      B. UNION      C. ALL      D. TOP  
(3) A. JOIN              B. UNION      C. INTO      D. LIKE  
(4) A. GROUP BY      B. COMPUTE BY      C. HAVING      D. COMPUTE

答: BABA

14、用 SELECT 进行模糊查询时,可以使用\_\_\_关键字,但要在条件值中使用\_\_\_等通配符来配合查询。

答: like, %

15、计算机字段的累加和的函数是\_\_\_,统计结果集行数的函数是\_\_\_。

答: sum 和 count

16、什么是数据库系统?

答: 数据库系统是储存、管理、处理和维持数据的软件系统,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。

17、试述数据库完整保护的主要任务和措施。

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。

18、什么是触发器和存储过程？

答：触发器是存储在数据库中的块，这些块一旦被构造后，就可以多次执行，当触发它的事件发生时调用该触发器。触发事件是指对表中数据的操作，如插入、删除和修改。

存储过程是将常用的或很复杂的工作，预先用 SQL 语句写好并用一个指定的名称存储起来，那么以后要叫数据库提供与已定义好的存储过程的功能相同的服务时，只需调用 execute, 即可自动完成命令。

19、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。

20、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 int primary key auto_increment,
    name varchar(20)
);
```

-- 系统模块表

```
create table modules (
    id int primary key auto_increment,
    name varchar(50),
    url varchar(50)
);
```

-- 模块操作表

```
create table operations (
    id int primary key auto_increment,
    name varchar(20),
    mid int,
    constraint foreign key (mid) references modules(id)
);
```



-- 权限表

```
create table rights (
  id      int primary key auto_increment,
  name    varchar(20),
  url     varchar(50),
  operationid int references operations(id));
```

-- 角色权限设置表

```
create table rolerights (
  id      int primary key auto_increment,
  roleid  int references roles(id),
  rightid int references rights(id)
);
```

21、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;
```

## 22、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_corp
```

--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;

## 23、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;`

24、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 int primary key auto_increment,
    room_id varchar(10),
    isUsed char,
    begin timestamp,
    end timestamp
);
insert into meeting values(1,'201',1,'2003-03-10 15:00:00','2003-03-10 16:00:00');
insert into meeting values(2,'301',0,'2003-03-10 17:00:00','2003-03-10 22:00:00');
```

2)

```
package com.tarena;
import java.sql.*;

public class Test {
    public static void main(String[] args) {
        String driverName = "com.mysql.jdbc.Driver";
        String url = "jdbc:mysql://localhost:3306/tarena";
        String username = "root";
        String pwd = "1234";
        Connection con = null;
        Statement stmt = null;
        ResultSet rs = null;
        String start = "' 2003-03-10 15:00:00' ";
        String end = "' 2003-03-10 15:00:00' ";
        String room_id="201";
        try {
            Class.forName(driverName);
            con = DriverManager.getConnection(url, username, pwd);
            stmt = con.createStatement();
            String sql = "select isUsed from meeting where
begin="+start+"and end = "+end+"and room_id="+room_id;
            System.out.println(sql);
            if (stmt.execute(sql)) {
                rs = stmt.getResultSet();
            } else {
                int num = stmt.getUpdateCount();
            }
            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();
            }
        }
    }
}
```

25、下面 SQL 命令哪些是 DML？（写命令，不要写标号）哪些是 DDL？

- A、SELECT
- B、UPDATE
- C、INSERT
- D、DELETE
- E、CREATE
- F、ALTER
- G、DROP

答：DML 命令是：ABCD；DDL 命令是：EFG。

26、Oracle 数据库和 SQL Server 数据库字符串连接操作符号是什么？

答：Oracle 数据库字符串连接操作符是||, 例如 ' a' || ' b' 等价于 ' ab'  
SQL Server 数据库字符串连接操作符是" +"

27、在 Oracle 数据库中，字符串是用单引号引起来的，那么，“I' m a student.” 这个字符串中有单引号，我们用怎样的办法进行处理。

答：可以用两个单引号来表示是个单引号，例如：insert into w values ( 'I''m a student' );

28、在 sqlPlus 中调试存储过程：

```
declare
  -- Local variables here
begin
  -- Test statements here
  DBMS_OUTPUT.PUT_LINE('hello world! ');
end;
```

但是始终没有输出“hello world!”，如果要输出该内容如何去做。

- A. sql->show all
- B. sql->serveroutput OFF
- C. sql->set serveroutput on

答：C

29、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);
```

30、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 int(5) primary key,
departmentName varchar(20)
);
create table b(
id int(5),
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;
```

31、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, sc, grade
   from (select student_id, sc, grade,
              row_number() over(partition by sc 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 sc, count(sc)
   from LEANR
   group by sc
   having count(sc)>=2
   order by count(sc) desc;

```

### 32、存储过程和函数的区别

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。

### 33、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;
```

#### 34、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%';

#### 35、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;
```

36、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 user group
by username having count(username)>1);
```

37、将表 a 中的数据查询出来, 并写入表 b, 使用 oracle 存储过程实现

答:

```
create or replace procedure Insert_Pro
as
begin
  insert into b (select * from a);
end;
```

38、向表 a 中添加字段 PolicyNo, 类型为字符串, 长度为 25 位, 请写出 SQL;

答: alter table a add (PolicyNo varchar2(25) );

39、向表 a 中的 PolicyNo 和 StartDate 字段上建立一个索引, 请写出 SQL;

答: create index a\_index on a(PolicyNo, StartDate);

40、将表 a 中 ClassCode 字段的值修改为 01, 条件是险种 RiskCode 为空的、终保日期 StartDate 大于“2007-02-01”、该保标志 UnderWriteFlag 为 1 或 3, 请写出 SQL;

答: update a set ClassCode='01'  
where StartDate>to\_date("2007-02-01", "yyyy-mm-dd")  
and UnderWriteFlag in (1,3) and RiskCode=null;

41、Sql 中左连接和右连接的作用分别是什么？

答：左连接：表示要查询的左边的表的所有记录均会被显示

右连接：表示要查询的左边的表的所有记录均会被显示

42、假设表 a 和表 b 的数据有对应关系，表 a、表 b 中数据记录之间的对应关系可能有几种？

答：一对一、一对多、多对多

43、用 SQLPLUS 的\_\_\_\_\_命令可以查看表的结构信息，包括列的名称和数据类型。

答：desc

44、ORACLE 数据库的约束类型有\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。

答：not null, unique, check, primary key, foreign key

45、ORACLE 的数据库导入命令是\_\_\_\_\_

答：imp

46、目前主流的数据库产品有哪些？你使用过的有哪几种？

答：Oracle、MySQL、DB2、SQLServer

47、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 user(id number(10) primary key,
```

```
Uname varchar2(20) not null unique,
```

```
);
```

Comments:

```
Create table comments(id number(10),
```

```
comments_id number(20) not null unique,
```

```
title varchar2(20) not null,
```

```
comments varchar2(255) not null,
```

```
foreign key(id) reference user(id)
```

```
);
```

Replay:

```
Create table replay(id number(10),
```

```
comments varchar2(255) not null,
```

```
foreign key(id) reference 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。

48、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, Upper ComCode—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，对于关系的维护和管理都会变得比较方便。

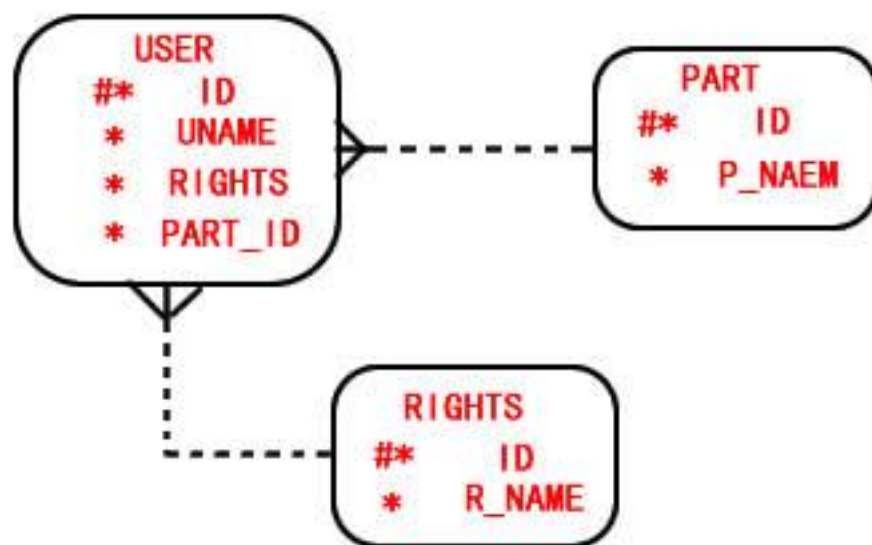
49、设计一个权限系统。类似 sqlserver 那样，包含用户，角色，权限。用户可以属于某个角色，也可以拥有指定的某个权限。设计数据库表结构，画出模型图。

答：用户：create table user(id number(10) primary key,  
Uname varchar2(20) not null unique  
Rights varchar2(10) foreign key references rights(id) unique,  
Part\_id number(10) foreign key references part(id) unique  
);

角色：create table part(id number(10) primary key,  
P\_name varchar2(20) not null unque  
);

权限：create table rights(id number(10) primary key,  
R\_name varchar2(20) not null unque  
);

模型图：



## 操作系统部分

1、Unix 中如何查看本机 MAC 地址？

答：ipconfig -all

2、请对 linux 命令 su 的使用作说明：

答：1) 作用：su 的作用是变更为其它使用者的身份，超级用户除外，需要键入该使用者的密码。

2) 格式：su [选项]... [-] [USER [ARG]...]

3) 主要参数

-f , --fast: 不必读启动文件（如 csh.cshrc 等），仅用于 csh 或 tcsh 两种 Shell。

-l , --login: 加了这个参数之后，就好像是重新登陆为该使用者一样，大部分环境变量（例如 HOME、SHELL 和 USER 等）都是以该使用者（USER）为主，并且工作目录也会改变。如果没有指定 USER，缺省情况是 root。

-m , -p , --preserve-environment: 执行 su 时不改变环境变数。

-c command: 变更账号为 USER 的使用者，并执行指令（command）后再变回原来使用者。

USER: 欲变更的使用者账号，ARG 传入新的 Shell 参数。

3、Unix 中如何删除文件

答：rm file

4、Unix 中如何对文件改名

答：mv file1 file2

5、Unix 中如何复制文件

答：cp file1 file2

6、Unix 中如何显示文件的内容

答：more、cat

7、show the path of work directory

答：pwd

8、go to previous directory

答：cd ..

9、show system date

答：date

10、verify memory

答：df -k

11、show used kernel version

答: `uname -r`

12、show details of files and directory

答: `ls -ld`

13、create two directories simultaneously

答: `mkdir dir1 dir2`

14、remove a directory called 'dir1' and contents recursively

答: `rm -r dir1`

15、create a symbolic link 'lnk1' to file 'file1'

答: `ln -s lnk1 file1`

16、outputs the mime type of the file 'file1' as text

答: `file file1`

17、lists known encodings

答: `iconv -l`

18、search files with '. bin' extension within directory '/ home/user1'

答: `find /home/user1 -name *.bin`

19、show list of partitions

答: `mounted df -h`

20、estimate space used by directory 'dir1'

答: `du -sh dir1`

21、delete user 'helen' and the home directory

答: `userdel -r user1`

22、change user owner of a directory 'directory1' and all the files and directories contained inside to 'helen'

答: `chown -R user1 directory1`

23、create a tarball compressed into gzip for directory 'dir1'

答: `tar -cvfz archive.tar.gz dir1`

24、view last six lines of a file 'file1'

答: `tail -6 file1`

25、replace "string1" with "string2" in example.txt

答: `sed 's/string1/string2/g' example.txt`



26、show configuration of an ethernet network card

答: ifconfig eth0

27、activate an interface 'eth0'

答: ifup eth0

28、show routing table

答: route -n

29、show hostname

答: hostname

30、display linux tasks using most cpu

答: top

31、force closure of the process '3518' and finish it

答: kill -9 3518

32、display kernel loaded

答: cat /proc/mounts

33、displays linux tasks

答: ps

34、Please write a script to count the number of .html files in /doc directory.

答: find . -name "\*.html" | wc -l

35、某文件的权限为: drw-r--r--, 用数值形式表示该权限, 则该八进制数为: \_\_\_\_, 该文件属性是 \_\_\_\_。

答: 八进制数为: 644; 该文件的属性是: 文件的创建者拥有读写权限, 其他人只有读权限

36、安装 Linux 系统对硬盘分区时, 必须有两种分区类型: \_\_\_\_ 和 \_\_\_\_。

答: 必须有两种分区类型: 文件系统分区 和 交换分区。

37、CD-ROM 标准的文件系统类型是 \_\_\_\_。

答: iso9660

38、Linux 使用支持 Windows 9. x/2000 长文件名的文件系统的类型是 \_\_\_\_。

答: 文件系统的类型是 vfat

39、看下实例:

```
[root@mylinux u01]# ls -l
total 16
```

```
drwxr-xr-x  6 root  root    4096 Jul  3  2005  database
```

```
drwxr-xr-x 19 oracle oinstall 4096 Jul 24 13:35  oracle
```

如果把目录 database 以及目录下的所有内容的用户和组设置成 oracle 和 oinstall, 用什么命令, 如何写?

答: `chown -R oracle:oinstall database`