丁伟成总结 java 程序员面试宝典之”葵花宝典”,要练此功，必须苦其心志，劳其筋骨.

JavaSE 部分：

1.什么是面向对象？面向对象有哪些特征？

以事物为驱动的编程思想。

我理解的面向对象是必须有一个具体的事物，

这个事物可以是一个具体的人，一个具体的物，

这个事物有自己的特征（属性），有自己的行为（方法），

那么对这个事物特征的描述，和对行为的操作称为面向对象的。

继承：

抽取出同类事物中的共同特性，最大限度的体现出类的重用性

例如：父子类之间 对于父类而言，他的一些东西（属性和方法）是可以共享的，

对于子类而言，它可以共享到父类的一些东西（属性和方法），那么

我们把父类与子类的这种关系称为继承。

封装：

是指对象的属性和操作结合在一起，组成一个独立的对象；

其内部信息对外是隐蔽的。对于外部而言，只能知道这个对象能干什么，

但不知道他是怎么做的。

多态：

行为和引用。

一个类中（行为）。

例如: 打 ， 打人，打车，打麻将 ，也叫方法重载

父子类之间（引用）

儿子可以引用父亲的行为,也叫方法的重写。

*2.*说一下什么是 javaBean 规范：

1.要有包--类必须写在包中

2.属性都是私有--private

3.要有无参构造方法

4.写有效的 setXXX getXXX 方法

5.实现序列化接口

*3.*简述一下 java 基本数据类型及所占位数, java 基本数据类型：4 类 8 种

逻辑型：boolean false/true 1byte 8bit

字符型：char 2byte 16bit 运算时按照 int 类型运算

整数型：

byte(1byte 8bit) 运算时按照 int 类型运算

short(2byte 16bit) 运算时按照 int 类型运算

int(4byte 32bit)

long(8byte 64bit) 后缀为 L/l

注：整数类型的默认类型是 int 类型 也就是说 整数类型的字面量都是 int 类型

浮点数型：
float(4byte 32bit)　后缀为 F/f
double(8byte 64bit) 后缀为 D/d
注：浮点类型的默认类型是 double 类型

除了基本数据类型，其他类型都是引用类型。
引用类型首字母大写的类型：String，Integer 等

Java 中的字面量：true，false，null，18，1.8
Java 中的特殊关键字：goto　const

Int a1=5+4;//正确 编译期优化
byte b1=a1+5;//错误　a1 是变量　遵循类型运算规则

short s1=2 short s2=0
s2=s1+s2;//错误 a1 是变量　遵循类型运算规则
s2+=s1;//正确　+=是复合运算类型 直接在 s2 值的基础追加一个 s1 的值

4. 说出 9 个的启动时异常:

RunTimeException
|---NullPointerException
|——ArrayIndexOutOfBoundsException
|——ArithmeticException
|——ClassCastException
|——NumberFormatException
|——SystemException
|——ArrayStoreException
|——EmptyStackException
|——BufferOverflowException

注：异常 Exception 分为两大类：
编译时异常：编译期间要检查的
运行时异常：编译期间不检查的

*5.*说出 5 个常用的 java-api 包
java.lang
java.util
java.sql
java.text
java.io

*6.*说出 String 类中常用的 5 个方法
charAt() 返回指定索引处的 char 值
concat() 将指定字符串连接到此字符串的结尾
length()返回此字符串的长度

split()根据给定正则表达式的匹配拆分此字符串
trim()返回字符串的副本，忽略前导空白和尾部空白.

7. HashMap 和 HashTable 的区别：

1.HashMap 允许空键值对，HashTable 不允许
2.HashMap 不是线程安全的，HashTable 是
3.HashMap 直接实现 Map 接口，HashTable 直接继承 Dictionary 类

8. ArrayList，Vector，LinkedList 存储性能和特性

它们都实现 List 接口
ArrayList 和 Vector 都是基于数组实现的
LinkedList 基于双向循环链表（查找效率低，添加删除容易）
ArrayList 不是线程安全的而 Vector 是线程安全的，所以速度上 ArrayList 高于 Vector

*9.*Collection 和 Collections 的区别。

Collection 是集合类的上级接口，继承与他的接口主要有 Set 和 List.
Collections 是针对集合类的一个帮助类，他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。

*10.*StringBuffer 和 StringBuilder 的区别?

StringBuffer 是线程安全的 速度慢 旧
StringBuilder 是非线程安全的 速度快些 新

*11.*List、Map、Set 三个接口，存取元素时，各有什么特点？

List 以特定次序来持有元素，可有重复元素。
Set 无法拥有重复元素,内部排序。
Map 保存 key-value 值，value 可多值。

12. final, finally, finalize 的区别

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

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

方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。
重写 Overriding 是父类与子类之间多态性的一种表现,方法名，参数列表，
返回值类型都得与父类的方法一致.
重载 Overloading 是一个类中多态性的一种表现。重载的方法是可以改变返回值的类型。

*14.*写出选择，冒泡，插入排序的代码

选择排序：(每一轮比较选择一个最小的或最大的元素排在前面)

```
for(int i=0;i<ary.length-1;i++){
    for(int j=i+1;j<ary.length;j++){
        if(ary[i]>ary[j]){
            int t = ary[i];
            ary[i]=ary[j];
            ary[j]=t;
        }
```

```
        }
    }
```

冒泡排序：(依次比较相邻的两个数，将小数放在前面，大数放在后面.)

```
    for(int i=0;i<ary.length-1;i++){
        for(int j=0;j<ary.length-i-1;j++){
            if(ary[j]>ary[j+1]){
                int t = ary[j];
                ary[j]=ary[j+1];
                ary[j+1]=t;
            }
        }
    }
```

插入排序：

```
    for (int i = 1; i < ary.length; i++) {
        int temp = ary[i];
        int j;
        for (j = i - 1; j >= 0 && temp < ary[j]; j--) {
            ary[j + 1] = ary[j];
        }
        ary[j + 1] = temp;
    }
```

15.写出二分查找的代码：

注：二分法从数组查询元素，必须保证数组内部元素是有顺序的。

意思就是先对数组进行排序。

```
public class BinarySearch {
 static int idx=1;
 public static void find(int leftIndex,int rightIndex,int val,int[] arr){
  int midIndex=(rightIndex+leftIndex)/2;
  int midVal=arr[midIndex];//找到中间的数
  if(rightIndex>=leftIndex){
  //如果要找的数比midVal大
   if(midVal>val){
        idx++;
    //在arr左边数中找
    find(leftIndex,midIndex-1,val,arr);
   }else if(midVal<val){
    //在arr的右边去查找
        idx++;
        find(midIndex+1,rightIndex,val,arr);
   }else if(midVal==val){
    System.out.println("找到下标"+midIndex+"共查找了"+idx+"次");
   }
```

```
        }else{
          System.out.println("没有该数!");
        }
       }

       public static void main(String[] args) {
        int[] arr=new int[]{100,300,500,800,1000,2000,3000};
        find(0,arr.length-1,1000,arr);
       }
}
```

16.实现线程安全的两种方式

1）synchronized 方法：通过在方法声明中加入 synchronized 关键字来声明 synchronized 方法。

2）synchronized 块：通过 synchronized 关键字来声明 synchronized 块。

17.实现多线程的两种方式

1)继承 Thread 类

2)实现 Runnable 接口

优先选择实现 Runnable 接口 因为比较灵活。

18. 简述如下几个概念。

程序：电脑上面的可运行文件.

进程：正在运行的程序，是程序动态的执行过程。

线程：一个程序内部的顺序控制流.

并发: 进程是并发运行的, 操作系统(OS)将时间划分为很多时间片段, 尽可能均匀分配给正在运行的程序, 微观上进程走走停停, 宏观上都在运行

这种都运行的现象叫: 并发。

19. 说一下 “==”和 equals()方法在字串变量操作中的不同?

”==”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。

20. sleep()和 wait()有什么区别？

sleep 是线程类（Thread）的方法，导致此线程暂停，然后执行给定时间，让出 cpu 给其他线程，但不会释放对象锁，时间到了自动恢复.

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。

21.&与&&的区别?

&位运算符,非短路逻辑运算符，它会把所有条件执行完毕之后，才会返回结果

&&逻辑运算符（and）：短路运算符，遇到不符合条件，立即终止程序的执行

22. error 和 exception 区别

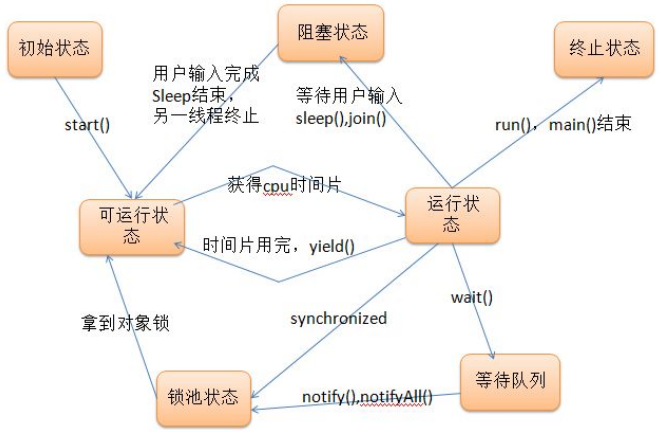
error:表示恢复不是不可能的一种严重的问题,比如：内存溢出，不指望程序处理

exception 程序运行时的异常，如果程序设计合理从不会出现的情况

23.请说出你所知道的线程同步的方法。

join():合并当前线程，相当于方法调用
yield():让出 cpu
wait():使一个线程处于等待状态，并且释放所持有的对象的 lock；
sleep():使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉 InterruptedException 异常；
notify():唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，而是由 JVM 确定唤醒哪个线程，而且不是按优先级；
notityAll():唤醒所有处入等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。
suspended() –挂起，该方法已过时: 在临时停止或中断线程的执行时，线程就处于挂起状态。
resume()—恢复，该方法已过时 ：在挂起的线程被恢复执行时，可以说它已被恢复。
stop()--暴力结束当前正在执行的线程，该方法已过时

24.线程的生命周期：



25.IO

字节流：
FileInputStream 和 FileOutputStream 对文件基本的输入输出操作
BufferedInputStream 和 BufferedOutputStream 加缓冲区的操作
DataInputStream 和 DataOutputStream 对基本操作的扩展，
可以一次读一种基本数据类型的数据.
字符流:
InputStreamReader 和 OutputStreamWriter 基本字符流 可以处理字符编码
BuffereReader 和 PrintWriter 可以一次读取一行数据

26.什么是反射？
在运行过程中:

1.对于任意一个类,可以知道这个类的属性和方法.
2.对于任意一个对象,可以调用这个对象的任意方法.
对于这种动态获取的信息,以及动态调用对象方法的功能称为反射机制.

java 反射机制提供的功能:
1.运行时判断任意对象的所属类;
2.运行时构造任意类的对象;
3.运行时判断和调用对象的成员变量和方法;
4.生成动态代理.

27.什么是回调函数?

某个程序 S(Student.main) 调用 服务程序 A(Arrays) 中的某个方法(sort), 服务程序 A 的 sort 方法在某个时候反过来调用 S 的某个方法(compareTo),这种情况下,compareTo 叫做 S 的回调方法。

例如：

```
public class Student implements Comparable{
    private int id;
    private String name;
    private int age;
    private int score;
    //构造器
    //getter / setter 方法
    //回调方法
    public int compareTo(Object obj){
        return
        this.id - ((Student)obj).id;
    }
}
Student s1 = new Student(1,"a",18,89);
Student s2 = new Student(2,"x",22,94);
Student s3 = new Student(3,"w",19,78);
Student [] arrs = {s1,s2,s3};
Arrays.sort(arrs);
```

28. 遍历文件夹下所有.java 的文件?

```
public void listFiles(String path){
   File dir = new File(path);
   Files files [] = dir.listFiles(new FileFilter(){
       public boolean accept(File f){
           return f.getName().endWith(".java");
       }
   });
   for(File file : files){
      System.out.println(file.getName());
   }
```



```
}
```

29.描述一下 JVM 加载 class 文件的原理机制?
JVM 中类的装载是由 ClassLoader 和它的子类来实现的
Java ClassLoader 是一个重要的 Java 运行时系统组件。
它负责在运行时查找和装入类文件的类。

30.打印三位数中所有的水仙花数(一个数的每一位数的 3 次方相加等于它本身):

```
for(int i=100;i<1000;i++){
        int a=i/100;
        int b=i/10%10;
        int c=i%10;
        if(a*a*a+b*b*b+c*c*c==i){
            System.out.println(i);
        }
    }
```

31.不用临时变量，交换两个数的位置：

```
int a=-4;
int c=-3;
```

方式一：a=a^c;c=a^c;a=a^c;
方式二：a=a*c;c=a/c;a=a/c;

32.不用循环的方式，求出 1+2..+100=?
利用递归：

```
public static int f(int n){
    if(n==1){
        return 1;
    }else{
        return n+f(n-1);
    }
}
```

33.数字倒置：求任意一个整数的倒过来的数(不准用字符串)

```
Scanner sc=new Scanner(System.in);
    System.out.println("请输入:");
    int num=sc.nextInt();
    int renum=0;
    while(num!=0){
        renum=renum*10+num%10;
        num=num/10;
    }
    System.out.println(renum);
```

34. 求一个数是否是素数?

```
Scanner sc=new Scanner(System.in);
```

```
        System.out.println("请输入一个数");
        int n=sc.nextInt();
        int i=0;
        for( i=2;i<n;i++){
            if(n%i==0){
                break;
            }else{
                continue;
            }
        }
        if(i>n-1){
            System.out.println("是素数");
        }else{
            System.out.println("不是素数");
        }
```

35.百元买百鸡？（公鸡 3 元一只母鸡 2 元一只，小鸡一元 3 只）

```
    for(int i = 1; i <= 33 ;i++){
            for(int j = 1; j <= 49;j++){
                int k = 100 - i - j;
                if(k%3==0 && (i*3+j*2+k/3)==100){
                    System.out.println("公:"+i+"母:"+j+"小鸡:"+k);

                }
            }
        }
```

36.打印杨辉三角问题？

```
        int[][] a = new int[10][10];
       for(int i = 0 ; i < 10 ; i++){
           for(int j = 0 ; j <= i;j++){
               a[i][j] = 1; }
       }
       /*从第三行第二列开始  不包含最后一列*/
       for(int i = 2; i < 10; i++){
           for(int j = 1; j < i ; j++){
               /*这个元素就等于它上一个一维数组中的该位置元素+前一个元素*/
               a[i][j] = a[i-1][j]+a[i-1][j-1];
           }
       }
       for(int i = 0 ;i < 10 ; i++){
           for(int j = 0 ; j <= i; j++){
               System.out.print(a[i][j]+" "); }
           System.out.println(); }
```

Oracle 部分：

1.薪水排序后薪水排名在第 3--5 的员工

1)select * from(select ename,sal,rownum rn from
(select ename,sal from emp_44 where sal is not null order by sal desc) where rownum<6)where rn>2;

2)select * from(select ename,sal,rownum rn from
(select ename,sal from emp_44 where sal is not null order by sal desc))where rn between 3 and 5;

2. 删除一张表中所有数据的方式？

1.truncate table 命令将快速删除数据表中的所有记录

2.delete 产生 rollback，如果删除在数据量的表速度会很慢，同时会占用很多的 rollback,segments.truncate 是 DLL 操作，不产生 rollback，速度会快一些。

3. 用一条 sql 语句取出所有姓名有重复的学员姓名和重复的记录数.

```
select name, count(*) from student group by name
having count(*) > 1 order by count(*) desc;
```

4. 去除 oracle 数据库 表中重复数据应有如下两种方法：

方法一：创建新表删除旧表法

1）create table student1 as select distinct id, name, score from student;

2）drop table student;

3）rename student1 to student;

方法二：使用 rowid(地址)伪列

删除伪列地址除了最大地址值以外的记录

```
delete from temp where rowid not in (select max(rowid) from temp group by id);
```

删除伪列地址除了最大地址值以外的记录

```
delete from temp where rowid not in (select min(rowid) from temp group by id);
```

5. 按工资进行排名，排名从 1 开始，工资相同排名相同（如果两人并列第一则没有第二名，从第三名继续排）

方法一：

```
select empno,ename,sal, (select count(*)+1 from iv_emp where sal>e.sal) rank
  from iv_emp e
```


order by rank asc;

方法二：select e.empno,e.ename,e.salary,rank() over(order by
salary desc) mingci from emp_ding e;

6. 按工资进行排名，排名从 1 开始，工资相同排名相同。

select e.empno,e.ename,e.salary,dense_rank() over(order by
salary desc) mingci from emp_ding e;

7. 查询每个部门工资最高的前 2 名员工

方法一:

```
select *
from( select empno,ename,salary, deptno,(select count(*)+1 from emp_ding where salary>e.salary and deptno = e.deptno ) rank
        from emp_ding e) ee
where ee.rank<=2
order by deptno,rank asc;
```

方法二：

```
select * from ( select e.empno,e.deptno,e.ename,e.salary, dense_rank() over (partition by deptno order by salary desc) mingci from emp_ding e ) x where x.mingci<=2;
```

8. 查询所有科目成绩都大于 80 分的人

```
select distinct name from student where name not in (select name from student

where score<80);
```

PL/SQL 部分：

解释如下概念

1.触发器(trigger)：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(inserting)、删除(deleting)和修改(updating)。需要注意的是：触发器没有参数，没有返回值，不能够显示调用。

```
create or replace trigger user_log     –user_log 该触发器的名字
before delete or update or insert
on t_user
for each row
begin
if deleting then
insert into t_user_log values
(myseq.nextval,'用户'||:old.username||'被删除了'); --:old 特殊的变量表示操作的原数据
```



```
elsif inserting then
insert into t_user_log values
(myseq.nextval,'用户'||:new.username||'被添加了');--new,特殊的变量表示操做的新数据
elsif updating then
insert into t_user_log values
(myseq.nextval,'用户'||:old.username||'被修改了'||:new.username);
else
null;
end if;
end;
```

2.存储过程(procedure)：我的理解就是一堆 sql 的集合，可以建立非常复杂的查询，编译运行运行一次后，以后再运行速度比单独执行 SQL 快很多.

1、具有删除功能的存储过程

```
create or replace procedure
delete_stu(v_stuid number)
is
begin
delete from t_student where
sid=v_stuid;
end;
```

调用存储过程：

```
exec delete_stu(1001);
```

3.游标(cursor)：游标(cursor)，是一种控制结构，可以帮助我们处理多条记录。游标不是一种数据类型。

对于游标的使用：

1、声明游标 cursor cursor_name is select....

2、打开游标 open cursor_name

3、从游标中抓取数据 fetch cursor_name into v_row_data

4、关闭游标 close cursor_name

静态游标的使用：

```
declare
v_data t_student%rowtype;
cursor mycursor
is
select * from t_student;
begin
open mycursor;
fetch mycursor into v_data;
dbms_output.put_line(v_data.stuname);
```



```
close mycursor;
end;
```

动态游标的使用：

```
declare
type cur_type is ref cursor;
v_data t_student%rowtype;
v_mycursor cur_type;
begin
open v_mycursor
for
select * from t_student;
fetch v_mycursor into v_data;
while v_mycursor%found loop
dbms_output.put_line(v_data.stuname);
fetch v_mycursor into v_data;
end loop;
close v_mycursor;
end;
```

4.索引（index）：相当于书签，能够提高数据的访问效率

主键列自动创建索引

其他列必须手动创建索引。

一般插入数据之后再创建索引。不然每插入一条数据，数据库会更新一次索引，这样大大降低了效率.

5.视图(view):虚拟的存在于数据库中。

用到视图的时候，系统会自动搜索数据库中的表，数据都是从表中取到的，并不是从视图中直接拿到的。

6.函数(function):必须有返回值，一般用于查询。

```
create or replace function emp_income
(v_empno emp_lc.empno%type)
return number
is
v_income number(10,2);
begin
select sal*12+nvl(comm,0)*12
into v_income
from emp_lc where empno=v_empno;
return v_income;
```


end;

7.序列(sequence):oracle 数据库没有提供主键自动增长，为了方便操作，
oracle 数据库提供了序列，它的作用就是能够间接的实现主键自增长。

JDBC&XML 部分：

1. 说说 jdbc 连接数据库的步骤
   1. 注册驱动 Class.forName(“”);
   2. 获得连接 Connection conn=DriverManager.getConnection(“”);
   3. 创建声明
   4. 执行 sql 语句,获得结果集，进行结果集的处理
   5. 关闭结果集
   6. 关闭声明
   7. 关闭连接，释放资源

2. statement 和 preparedstatement 的区别

   1.statement 是 preparedstatemnet 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,这样会导致恶意注入 sql 语句的问题出现.prepatedstatement 会将 sql 进行预编译，当 sql 语句要重复执行时，数据库会调用以前编译好的 sql，所以 preparedstatement 在性能方面会更好

   3.preparedstatement 在执行 sql 时，对传入的参数进行强制类型转换，以保证数据格式与底层数据库格式一致。

   4.PreparedStatement 相对于 Statemnet 要安全一些，可以有效防止 sql 注入.

   5.PreparedStatement 能够执行批处理，而 Statement 不行.

   另： CallableStatement 是用来调用存储过程的

3.数据库数据如何提高查询速度？

   1. jdbc setFetchSize（）；
   2. 数据库：索引
   3. 把大表可以拆分成多个小表
   4. dba 对表进行分区

4. 解析一个 XML 文档有哪些方式？

   解析有：dom 和 sax 两种

   dom：把整个 XML 文档放入内存，适合 XML 随机访问,占用内存资源大

   sax：事件驱动型的 XML 解析方式，顺序读取，不用一次装载整个文件，遇到标签会触发一个事件，适和对 XML 的顺序访问，占用内存资源稍小

5. XML 文档定义有几种形式？

   a：两种形式dtd schema

b: 本质区别:schema 本身是xml 的，可以被XML 解析器解析(这也是从 DTD 上发展 schema 的根本目的)，
c:有 DOM,SAX,STAX 等

6.Java 连接数据库有几种模式(方式)?
1.直接连接 JDBC（瘦连接）
2.通过客户端（胖连接）
3.使用桥连接 ODBC(桥接)

7.分页？
1. 缓存分页（假分页）
特点: 第一次取全部数据时比较慢.(可能非常慢)以后每次都从缓存中取数据,比较快.
只访问数据库一次.内存压力大.如果需求是一般只查看前几页,浪费内存资源.

2. 数据库分页（查询分页）
特点:每次只向数据库查询并返回一页的数据频繁的数据库访问.
每次查询的时间都差不多.没有缓存压力.

两者比较：
基于缓存的方式,适合小表,全部查看
基于查询的方式,适合大表,只查询部分数据

HTML/CSS/JavaScript 部分：

一个完整的网页由结构、表现、行为组成。

结构指 HTML、XHTML、XML 等
表现指 CSS 样式
行为指脚本(JS)
JavaScript 是 Netscape(网景)推出客户端运行的解释性脚本语言。
注：Javascript 是弱类型语言，变量的声明只能用 var

Html 基本结构：<html> <head><title></title></head><body></body></html>
设置隐藏文本框：<input type="hidden" name="id">
设置只读文本框：<input type="text" readonly="readonly" name="username">

== 等于 比较的是值
=== 绝对等于 不仅比较值 还要比较类型

Undefined:未定义
1.声明但没有赋值
2.没有声明的变量

获得对象：document.getElementById(“id”);
获得对象的值：document.getElementById(“id”).value

创建数组：
1）var 变量名称 = new Array()
2）var 变量名称 = new Array(长度)
3）var 变量名称 = new Array(值,....)
4）var 变量名称 = [值,...]
属性
length --> 返回数组成员的数目
方法
push() --> 向数组的末尾添加一个或多个成员,返回为新数组的长度
unshift() --> 向数组的开头添加一个或多个成员,返回为新数组的长度
shift() -> 删除数组的第一个成员，并且返回该成员。
pop() --> 删除数组的最后一个成员，并且返回该成员。
splice() --> 插入/删除数组成员，并且用新成员取代原有成员。
slice(start[,end]) --> 返回数组的一部分
join() --> 使用分隔符将数据成员连接在一起

客户端框架部分：

1、什么是 ajax？asynchronous javascript and xml:
异步的 javascript 和 xml。

为了解决传统的 web 应用当中“请求-处理-等待-响应”的弊端而创建的技术，其实质是：使用 javascript 调用浏览器内置的一个对象（XmlHttpRequest)异步向服务器发送请求，服务器返回 xml 或者 text 给 XmlHttpRequest，然后，javascript 使用服务器返回的数据更新页面。在整个过程当中，页面没有任何的刷新。

2.使用 ajax 发送异步请求的流程：
1.创建异步对象 XMLHttpRequest
2.与服务器建立连接请求 open();
3.发送异步请求 send();
4.注册监听事件，通过回调函数监听客户端与服务器交互的整个过程

通过属性 readyState(4)，得知服务器返回数据的状态，通过属性 status(200)
得知服务器是否正常返回数据。

readyState:

0 (未初始化) 对象已建立，但是尚未初始化（尚未调用 open 方法）
1 (初始化) 对象已建立，尚未调用 send 方法
2 (发送数据) send 方法已调用，正在发送数据.
3 (数据传送中) 已接收部分数据,但未完全返回.
4(响应结束) 数据完全返回.

status:

200:服务器正常返回。
404:请求未正确发送。
500:服务器报错。

返回结果的两种形式：

responseText:获得服务器返回的处理结果,该结果以文本的形式返回。
responseXml:获得服务器返回的处理结果,该结果以符合 dom 规范的对象返回。

什么是 JQuery?

Jquery 是一个 javascript 框架，对底层 javascript 代码进行封装，方便用户操作。
并且 jquery 提供了对 ajax 的支持，使用 jquery 也可以发送异步请求，
不需要屏蔽浏览器差异，代码书写简单。

Jquery 对 ajax 的支持：

1)、序列化元素：

serialize():将 jquery 对象包含的表单或者表单控件转换成查询字符串

2)、三个方法

（1）load(url),将服务器响应结果插入当前 div
jQuery 对象匹配的 dom 元素之内。一般用于从服务器获取静态的数据
(比如.html 文件),支持返回 javascript。

```
/* jquery 的 load 请求*/
    $(function(){
        $('#i4').click(function(){
            $("#d1").load("list.do");
        });
    });
```

（2）$.get(url,[data],[callback],[type])
$.post()格式同上。

```
/* jquery 发送 get 或者 post*/
    $(function(){
        $("#i5").click(function(){
            $.get(
                "test.do",
```

```
                {"username":$('#t1').val(),"password":$('#t2').val()},
                function(data,statusText){
                    if(statusText=="success"){
                        alert(statusText);
                        alert(data.id+","+data.username+","+data.password);
                    }else{
                        alert("系统异常，稍后再试.");
                    }
                },
                "json"
            );
        });
    });
```

（3）$.ajax({}):

```
/*jquery 发送 ajax 请求 */
    $(function(){
        $('#i6').click(function(){
            $.ajax({
                url:"test.do",
                type:"get",
                data:{"username":$('#t1').val(),"password":$('#t2').val()},
                dataType:"json",
                success:function(data,statusText){
                    alert(data.id+","+data.username+","+data.password);
                },
                error:function(xhr,statusText,errorThorwn){
                    alert(statusText);
                }
            });
        });
    });
```

什么是 DWR？

是一个支持后台调用的 ajax 框架，对于 ajax 异步请求进行了封装，它的优势在于，能够直接访问后台中的业务方法。

什么是 extJS？

是一个与后台技术无关的 ajax 框架。它主要注重页面设计方面，提供了一些优于传统 html 页面设计的效果。它里面的很多内容可以直接引入到项目中，大大 减少了公司对于页面的设计，因此受到了部分公司的青睐。

什么是 flex？

是一种富客户端技术。由于它富于表现的 UI，深受企业客户的青睐。但由于 flex 编译产生的 swf 文件过于庞大，对于带宽受限的用户是一个很严重的问题。同样的一个功能使用 html+css+javascript 开发只需要几十 kb，而 flex 则需要几百 kb，这也是需要考虑的现实问题。

但 flex 技术是未来的发展趋势.

Servlet&Jsp 部分：

1. MVC 的各个部分都有那些技术来实现?如何实现?

   MVC 是一种软件架构设计思想，不是设计模式.

   M-Model 模型

   模型的职责是负责业务逻辑。包含两部分：业务数据和业务处理逻辑。

   由 javaBean 充当,或者由容器管理的 javabean 充当(Spring,EJB)

   例如：实体类、DAO、Service 等都属于模型层

   V-View 试图

   试图的职责是负责显示界面和用户交互（收集用户信息）。

   属于视图的类是不包含业务逻辑和控制逻辑的 JSP(如果 JSP 页面中有<% %>就不能算是视图层的类，或者 JSP 中更有转发和重定向的控制逻辑也是不可以的)。

   C-Controller 控制器

   控制器是模型层 M 和视图层 V 之间的桥梁，用于控制流程。

   比如我们之前项目中写的 Struts2 的 Action 或原始的 Servlet

2.servlet 的生命周期

   web 容器加载 servlet，生命周期开始。

   通过调用 servlet 的 init()方法进行 servlet 的初始化。

   通过调用 service()方法实现，根据请求的不同调用不同的 doGet()或者 doPost()方法。

   结束服务，web 容器调用 servlet 的 destroy()方法。

3. jsp与servlet的区别及联系

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。

4. 数据库连接池的工作机制：

J2EE 服务器启动的时候，会创建一定数量的池连接，并维持不少于此数量的池连接。

程序需要时，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，池驱动会新建一批，数量由配置参数决定。

当调用池连接完成后，池驱动将此连接标记为空闲，其他调用就可以使用这个连接。

5 .jsp 有哪些内置对象?作用分别是什么?

答：JSP 共有以下 9 种基本内置组件（可与 ASP 的 6 种内部组件相对应）：

page　　jsp 网页本身

pageContext 网页的属性是在这里管理

request 客户端请求，此请求会包含来自 GET/POST 请求的参数

session 与请求有关的会话期，只要浏览器不关闭，保存在 session 里面的值都存在.
application servlet 正在执行的内容
config servlet 的构架部件
response 网页传回用户端的回应
out 用来传送回应的输出
exception 针对错误网页，未捕捉的例外

6.forward(转发) 和 redirect(重定向)的区别
答：forward 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。

7.Jsp 的四种会话范围
page 是代表与一个页面相关的对象和属性。作用域在当前页.
request 是代表与Web客户机发出的一个请求相关的对象和属性。
session 只要访问的浏览器不关闭，作用域就一直存在。
application 只要访问的服务器不关闭，作用域就一直存在。

8. 什么是 B/S 结构,C/S 结构?
C/S 是 Client/Server 的缩写。服务器通常采用高性能的 PC、工作站或小型机，并采用大型数据库系统，如 Oracle、Sybase、Informix 或 SQL Server。客户端需要安装专用的客户端软件。
B/S 是 Brower/Server 的缩写，客户机上只要安装一个浏览器(Browser)，如 Netscape Navigator 或 Internet Explorer，服务器安装 Oracle、Sybase、Informix 或 SQL Server 等数据库。在这种结构下，用户界面完全通过 WWW 浏览器实现，一部分事务逻辑在前端实现，但是主要事务逻辑在服务器端实现。浏览器通过 Web Server 同数据库进行数据交互。

9. 编码格式转换问题：
String str=new String("中国".getBytes("ISO-8859-1"),"GBK").trim();

10.URL 和 URI 的区别?
URL：统一资源定位符, 指的是 Internet 文件在网上的地址,用在客户程序和服务器上，定位客户端连接服务器所需要的信息，它不仅定位了这个信息资源，而且定义了如何找到这个资源.
URI：统一资源标识符，Web 上可用的每种资源 ：HTML 文档、图像、视频片段、程序等…是由一个通过通用资源标志符(Universal Resource Identifier, 简称"URI")进行定位,不局限于客户端服务器.
URI 一般由三部分组成:
1.访问资源的命名机制。
2.存放资源的主机名。
3.资源自身的名称，由路径表示。

URL 是 URI 的一个子集。

11.servlet 是线程安全的么？

Servlet 不是线程安全的。Servlet 的生命周期是由 web 容器负责，一般 web 容器对于同一个 servlet 只存在一份，也就是说如果 web 容器当前没有会新创建一个，如果存在就使用原来的 servlet。这样就形成了一个很大的问题，资源共享的问题，也就是不同的用户操作同一个 servlet，会导致数据不一致的现象。

如何解决这样的问题呢？

1. 让 servlet 实现 SingleThreadModel(该接口定义了如果处理访问同一个 servlet 的问题)
2. 给对应的业务操作加上同步锁块。
3. 尽量避免定义成员变量

三大框架部分：

Struts2 部分:

1. 介绍一下 struts2 的工作原理：

1 客户端初始化一个指向 Servlet 容器（例如 Tomcat）的请求

2 这个请求经过一系列的过滤器（用户定义）

3 接着过滤器(FilterDispatcher)被调用，FilterDispatcher 询问 ActionMapper 来决定这个请求是否需要调用某个 Action

4 如果 ActionMapper 决定需要调用某个 Action，FilterDispatcher 把请求的处理交给 ActionProxy

5 ActionProxy 通过 Configuration Manager 询问框架的配置文件，找到需要调用的 Action 类

6 ActionProxy 创建一个 ActionInvocation 的实例。

7 ActionInvocation 实例使用命名模式来调用，在调用 Action 的过程前后，涉及到相关拦截器（Intercepter）的调用。

8 一旦 Action 执行完毕，ActionInvocation 负责根据 struts.xml 中的配置找到对应的返回结果。返回结果是（但不总是，也可能是另外的一个 Action）一个需要被表示的 JSP 或者 FreeMarker 的模版。

2．介绍一下Struts2的工作流程？

1.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。

2.通过 struts 控制器进行处理，

3.经过一系列的拦截器处理

4.进行业务逻辑的处理

5.响应用户 JSP 将结果展现给用户。

3. 为什么struts2框架收到企业青睐？

1. servlet+jsp技术，使得项目非常紊乱.不利于维护.
2. 基于Struts开发的应用:
   不用再考虑公共问题
   专心在业务实现上
   结构统一，易于学习、维护
   新手也可写出好程序

3. struts2 框架提供了哪些辅助功能
   1 自动收集数据（成员变量）
   2 支持类型转换
   3 国际化
   4 异常处理
   5 标签

4. struts1 和 struts2 的区别
   1 struts1 控制器 servlet　　struts2 控制器 filter
   2 struts 收集数据时 ActionForm　　Struts2 成员变量
   3 struts1 execute(ActionForm,ActionMapping,Resquest,Response)
     struts2 execute()

   4 struts1 与 ServletAPI 耦合性强
     struts2 与 ServletAPI 耦合性低
   5 struts1 没有拦截器　struts2 有拦截器(可以使代码各司其职)
   6 struts1 只能使用 jstl 标签，而 struts2 提供了 OGNL 表达式

Hibernate部分：

1. hibernate工作原理
   1.读取并解析配置文件
   2.读取并解析映射信息，创建SessionFactory
   3.打开Sesssion
   4.创建事务Transation
   5.持久化操作
   6.提交事务
   7.关闭Session
   8.关闭 SesstionFactory
   为什么要用：
   1. 对JDBC访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。
   2. Hibernate是一个基于JDBC的主流持久化框架，是一个优秀的ORM实现。他很大程度的简化DAO层的编码工作
   3. hibernate 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。
   4.Hibernate 是面向对象进行操作的

2. Hibernate 对象状态有哪几种，并简单介绍一下。

1.临时状态：内存对象，并没有保存在数据库
2.持久化状态 已经保存在数据库并纳入了 session 缓存中
3.游离状态 已经保存在数据库中，但没有纳入 session 缓存中

3.什么是Hibernate延迟加载?
延迟加载机制是为了避免一些无谓的性能开销而提出来的，所谓延迟加载就是当在真正需要数据的时候，才真正执行数据加载操作。在 Hibernate 中提供了对实体对象的延迟加载以及对集合的延迟加载，另外在 Hibernate3 中还提供了对属性的延迟加载。

4．Hibernate中类之间的关联关系有几种?(如：一对多、多对多的关系)
many-to-one、one-to-many、many-to-many、 one-to-one

5. 说下Hibernate的缓存机制
一、hibernate一级缓存
（1）hibernate支持两个级别的缓存，默认只支持一级缓存；
（2）每个Session内部自带一个一级缓存；
（3）某个Session被关闭时，其对应的一级缓存自动清除；
二、hibernate二级缓存
(1) 二级缓存独立于 session，默认不开启；

6. Hibernate的查询方式
本地 SQL 查询、Criteria、Hql

MyBatis 或 Ibatis 部分:

什么是 Mybatis?

MyBatis 是支持普通 SQL 查询，存储过程和高级映射的优秀持久层框架。MyBatis 消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML 或注解用于配置和原始映射，将接口和 Java 的 POJOs（Plan Old Java Objects，普通的 Java 对象）映射成数据库中的记录。

Mybatis 框架和 hibernate 框架的区别?

1. Mybatis 是半自动化映射框架，hibernate 是全自动化映射框架.
2. Mybatis 必须在映射文件中写 sql 语句，hibernate 写的是 hql 语句。
3. Mybatis 不需要考虑缓存问题，而 hibernate 必须处理缓存问题.
4. Mybatis 适用于不更改数据库的情况下使用。Hibernate 则没有这种问题.

Spring 部分：

1.spring 的优点?
1.降低了组件之间的耦合性 ，实现了软件各层之间的解耦
2.可以使用容易提供的众多服务，如事务管理，消息服务等
3.容器提供单例模式支持
4.容器提供了 AOP 技术，利用它很容易实现如权限拦截，运行期监控等功能
5.容器提供了众多的辅助类，能加快应用的开发

6. spring 对于主流的应用框架提供了集成支持，如 hibernate，JPA，Struts 等
7. spring 属于低侵入式设计，代码的污染极低
8. 独立于各种应用服务器
9. spring 的 DI 机制降低了业务对象替换的复杂性
10. Spring 的高度开放性，并不强制应用完全依赖于 Spring，开发者可以自由选择 spring 的部分或全部

2. 什么是 DI 机制？

依赖注入（Dependecy Injection）和控制反转（Inversion of Control）是同一个概念，具体的讲：当某个角色需要另外一个角色协助的时候，在传统的程序设计过程中是由调用者来创建被调用者的实例。但在 spring 中创建被调用者的工作不再由调用者来完成，因此称为控制反转。创建被调用者的工作由 spring 容器来完成，然后注入调用者因此也称为依赖注入。

3. 什么是 AOP？

1. 面向切面编程提供声明式事务管理(利用动态代理机制)
2. spring 支持用户自定义的切面

用户可以利用自定义切面做：日志记录，权限控制，事务处理等..
可以不改变源代码的情况下为程序加一些业务逻辑进去，具有可插拔性.

4. 为什么要用spring?

Spring是一个轻量级的IOC和AOP框架。
IOC（控制反转）意味着将你设计好的类交给系统去控制，而不是在你的类内部控制。这称为控制反转
AOP（面向切面），它将那些影响多个类的行为封装到可重用的模块中，面向对象是把问题从同类事物中抽象出来，面向切面是把问题从不同类问题中抽象出来。

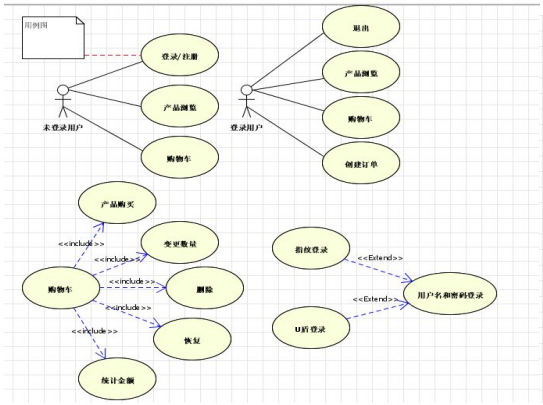
5. 应用服务器有哪些：

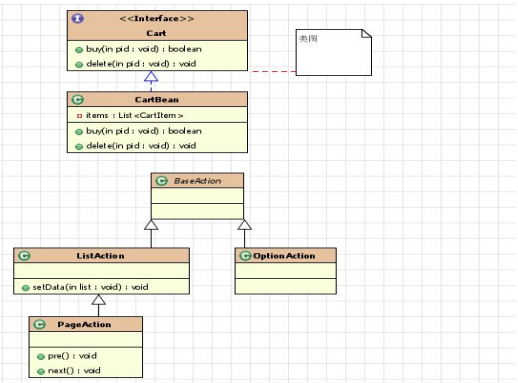
BEA WebLogic Server
IBM WebSphere Application Server
Oracle Application Server，
jBoss
Tomcat

UML 部分：

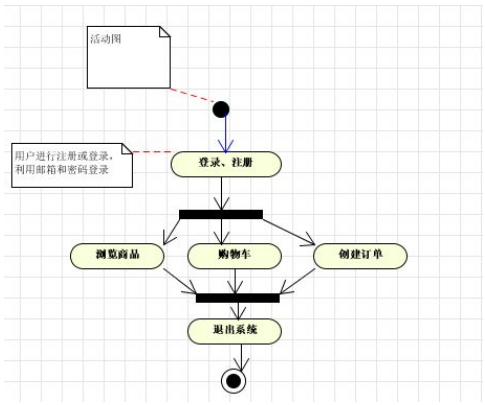
Unified Model Language 统一建模语言
该语言是利用形象化的符号描述出各种模型图，这些模型图可以在软件开发的各个阶段描述不同的内容。

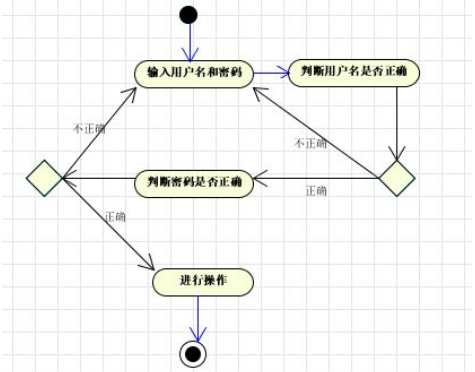
用例图: 静态图(包括类图、对象图和包图),



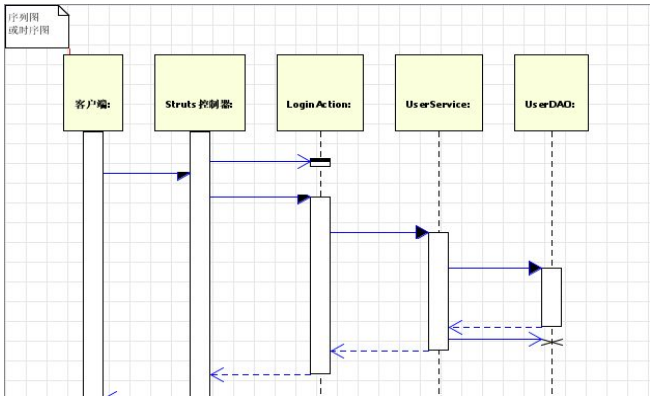


活动图,

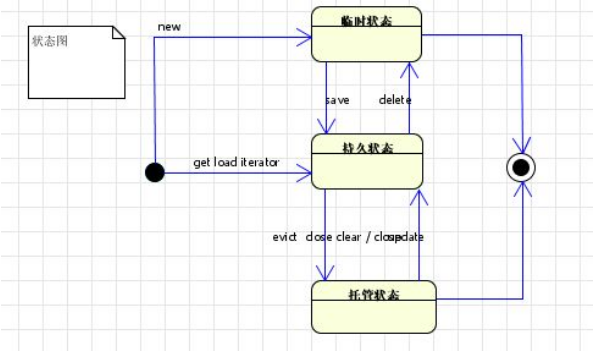




时序图也称序列图:



状态图:



设计模式部分

设计模式指的是解决问题的思路和方法。
使用设计模式可以解决一定的问题，实现代码重复利用，便于程序维护和扩展。
23 种设计模式可分为以下三大类:

1. 创建型模式--与对象创建相关的设计模式
   a) *单例模式
      特点：实现一个类只能创建一个对象实例.一般应用于工厂类的设计。
      1）饱汉式
      特点：在 getInstance 方法调用时，创建单例对象

```
public class 老婆 {
//3.静态变量，唯一性
private static 老婆 instance = null;
//1.将构造方法隐藏
private 老婆(){}
//2.提供一个静态方法，用于获取老婆对象
public static synchronized 老婆 getInstance(){
if(instance == null){instance = new 老婆();}
return instance;}
public synchronized void 做饭(){
System.out.println("做饭");}}
```

      2）饿汉式
      特点：在类加载时创建单例对象

```
public class 老婆 {
//3.静态变量，唯一性
private static 老婆 instance = new 老婆();
//1.将构造方法隐藏
private 老婆(){}
//2.提供一个静态方法，用于获取老婆对象
public static 老婆 getInstance(){
return instance;}
public void 做饭(){
System.out.println("做饭");}}
```

注意：避免使用时的线程并发问题。利用 synchronized 解决。

b) 工厂方法模式(定义一个抽象方法)
特点：利用一个工厂管理一个对象的创建
适用于对象类型频繁增减，可避免频繁修改原有工厂类。

c) 抽象工厂模式(定义多个抽象方法)
特点：利用一个工厂类管理一个系列的对象类型的创建
适用于对象类型特别多

2. 结构型模式--与对象的结构设计相关
a).适配器模式
特点：在子系统交互时，如果提供的类型和使用类型不匹配，可以采用此方法转换。封装类型转换的过程。
----Struts2 中大量采用了该模式---------
对 request,session,application 对象转换成 Map 结构的操作
对原有的 HttpServletRequestWrapper 封装成 StrutsRequestWrapper 类型

```
public class StrutsRequestWrapper
   extends HttpServletRequestWrapper{
      public StrutsRequestWrapper(
       HttpServletRequestWrapper request){
       super(request);
      }

      public Object getAttribute(String key){
       //return super.getAttribute
       //找不到去值栈获取
      }
}
HttpServletRequest request = new StrutsRequestWrapper(request);
JSP ${name}-->request.getAttribute("name")
```

3. 行为型模式--与对象交互，职责分配相关的模式
a).模版模式
特点：定义一个模版类，该模版类实现了一个处理的骨架算法，不同的实现部分放到子类具体实现.
Spring 的 Template 工具类采用了该模式设计

相关名词解释:

J2EE 或 JavaEE：J2EE 本身是一个标准，一个为企业分布式应用的开发提供的标准平台。
J2EE 也是一个框架，包括 JDBC、JNDI、RMI、JMS、EJB、JTA 等技术。

J2SE 或 javaSE： 一个 JAVA 标准开发平台，主要专注于 java 桌面程序，其中包含的技术有线程、IO、集合框架、Swing，AWT 等技术。

web 容器：给处于其中的应用程序组件（JSP，SERVLET）提供一个环境，
使JSP,SERVLET 直接更容器中的环境变量接口交互，不必关注其它系统问题。
主要有WEB 服务器来实现。例如：TOMCAT,WEBLOGIC,WEBSPHERE 等。
该容器提供的接口严格遵守J2EE 规范中的WEB APPLICATION 标准。
我们把遵守以上标准的WEB 服务器就叫做J2EE 中的WEB 容器。

EJB 容器：Enterprise java bean 容器。
更具有行业领域特色。他提供给运行在其中的组件
EJB 各种管理功能。只要满足J2EE 规范的EJB 放入该容器，
马上就会被容器进行高效率的管理。并且可以通过现成的接口
来获得系统级别的服务。例如邮件服务、事务管理。

JNDI： （Java Naming & Directory Interface）JAVA 命名目录服务。主要提供的功能是：提供一个目录系统，让其它各地的应用程序在其上面留下自己的索引，从而满足快速查找和定位分布式应用程序的功能。如：EJB、数据库驱动、JDBC 数据源及消息连接等。

JMS:（Java Message Service ,Java 消息服务）是一组 java 应用接口，提供创建，发送，接收，提取消息的服务。

JTA： （Java Transaction API）JAVA 事务服务。提供各种分布式事务服务。应用程序只需调用其提供的接口即可。

JAF： （Java Action FrameWork）JAVA 安全认证框架。提供一些安全控制方面的框架。
让开发者通过各种部署和自定义实现自己的个性安全控制策略。

RMI/IIOP: （Remote Method Invocation /internet 对象请求中介协议）
他们主要用于通过远程调用服务。
例如，远程有一台计算机上运行一个程序，它提供股票分析服务，我们可以在本地计算机上实现对其直接调用。当然这是要通过一定的规范才能在异构的系统之间进行通信。
注：RMI 是 JAVA 特有的。

JFC（java Foundation class ，java 基础类）是一组客户端图形、GUI（graphical user interfere，图形用户界面）和相关的编程任务的标准 jva API 的松散集合。由于绝大部分客户端 java 应用程序是以这些 API 为基础的，所以称其为基础类。

AWT（Abstract Windowing Toolkit，抽象窗口工具箱）是 JFC 功能的基础。

Swing： Swing 是使用纯 java 语言编写的更高级的 GUI 工具箱。它以 AWT 为基础，但提供了许多新的 GUI 组件与 GUI 相关的有用的应用程序服务。

API(Application Programming interface)应用程序编程接口

OOA（Object Oriented Analysis）面向对象的分析
OOD（Object Oriented Design）面向对象的设计
OOP（Object Oriented Programming）面向对象的编程实现
AOP（Aspect Oriented Programming）面向切面编程

JavaMail API 是一组抽象的 API，可用于构建邮件系统。

DTO（data transfer object）:数据传输对象
POJO（plain old java object）:普通 java 实体类
VO（view object）：视图对象
PO(persistence object)持久化对象
DAO（data Access object）数据访问对象

