数据库：

1 delete、drop、truncate 的区别？delete:删除某条记录；drop：删除表或者某列；truncate 清空表的内容。

delete：会发起事务，可以有 where 过滤条件(即删除指定的数据，列名不变),属于 dml 语句，删除效率比 truncate 低
drop：属于 DDL 语句，删除表，删除列
truncate：不会发起事务，不能有 where 过滤条件(即默认删除表中所有的数据，列名不变),属于 ddl 语句,删除效率比 delete 高，但有危险性

2 什么是视图，有什么特点，表和视图的区别

答：视图指的是为了方便调用而记录的一条 select 语句。他只是记录了一条 select 语句。
特点：占内存小，只有执行时才有数据，不能增删改，只能查询。
区别：1 表一直有数据，视图只有执行才有数据
2 表可以进行增删查改，而视图只能查询

3 普通索引和唯一索引有什么区别，索引有什么作用

区别：唯一索引，不允许具有索引值相同的行。
索引的作用：提升查询数据的效率，相当于表的一个附表，通常有一个索引列。

4 外连接有哪些？有什么特点

答：外连接分为左连接和右连接，左连接指左表为主表，显示左表的每一行，右连接反之。

5 什么是事务？

通俗理解就是做一件事情的过程，事务封装了一条 dml、或者多条 dml 语句。
这个过程有两种结果：要么全部成功、要么全部失败。
事务的四个特点：原子性，一致性，持久性，隔离性。

6 plsql 是什么？

plsql 是专用于 Oracle 服务器，在 SQL 基础之上，添加了一些过程化控制语句，叫 PLSQL。plsql 强调过程，主要有判断，循环语句。

7 定义一个存储过程，比如传入某个值 n，计算从 1 到 n 的和。

```
create or replace procedure qiuhe(n number) as
  i number(20) :=0;
  j number(20) :=0;
begin
  loop
    exit when i>n;
     j:=j+i;
     i:=i+1;
  end loop;

  dbms_output.put_line('1 到'||n||'的和:'||j);

end;

--调用方式 1

declare
 m number;
begin

  m:=&q;--&符号表示从键盘取值，后面的变量任意。

  qiuhe(m);
end;
```

8 有学生各科成绩表：table1

	XINGMING	KECHENG	FENSHU
1	张三	语文	80
2	张三	数学	75
3	李四	语文	76
4	李四	数学	90
5	王五	语文	81
6	王五	数学	100
7	王五	英语	90

8.1 写出对 table1 表中的记录进行增删改查的 sql，where 条件可自定义：

新增：

```
insert into table1 values ('张三','英语',100);
```

删除：

```
delete from table1 where xingming='张三';
```

修改：

```
update table1 set fenshu=100 where xingming='张三' and
  kecheng='数学';
```

查询：

```
select * from table1 where  xingming='张三';
```

8.2 查询分数都大于或者等于 80 的学生

```
select t.xingming
from table1 t
group by t.xingming
having min(t.fenshu)>=80;
```

或

```
select t.xingming
from table1 t
where t.fenshu>=80
group by t.xingming
having sum(t.fenshu)>=240;
```

8.3 查询平均分大于 80 的学生

```
select t.xingming
```

```
from table1 t

group by t.xingming

having avg(t.fenshu)>80;
```

8.4 假如 80 分及格，请写出一条 sql 展示下图的结果：

	XINGMING	KECHENG	FENSHU	是否及格
1	张三	语文	80	及格
2	张三	数学	75	不及格
3	李四	语文	76	不及格
4	李四	数学	90	及格
5	王五	语文	81	及格
6	王五	数学	100	及格
7	王五	英语	90	及格

```
select t.*,decode(sign(t.fenshu-80),0,'及格',1,'及格',-1,'
  不及格') 是否及格 from  table1 t;
```

或者

```
select t.*,
case
  when t.fenshu>=80 then '及格'
  else '不及格'
end 是否及格
from  table1 t;
```

8.5 表二结构如下，请写一条 sql 把表 1 的数据拷贝到表 2

XINGMING	KECHENG	是否及格

```
insert into table2
(select t.xingming,t.kecheng,
decode(sign(t.fenshu-80),0,'及格',1,'及格',-1,'不及格') 是否及格
```

```
from table1 t);
```

9 把这样的图 1 学生表 xs 用一条 SQL 语句查询得到图 2 的效果。

（分别用 decode 和 case when then 做出来）

图 1: 学生表 xs

	XNO	NAME	COURSE	SCORE
1	01	张三	数学	80.00
2	01	张三	语文	81.00
3	01	张三	英语	88.00
4	02	王五	数学	70.00
5	02	王五	语文	90.00
6	02	王五	英语	77.00

图 2：

	XNO	NAME	SHUXUE	YUWEN	YINGYU
1	02	王五	70	90	77
2	01	张三	80	81	88

解题思路：利用 decode 临时增加列名，并且对 decode 别名，同时利用子查询。

--decode 函数的高级运用的建表语句

```
--创建 xs 学生表
create  table xs(
  xno varchar2(3) ,
  name varchar2(10),
  course varchar2(10),
  score number(5,2)
)
--插入数据
insert into xs values ('01','张三','数学','80');

insert into xs values('01','张三','语文','81');

insert into xs values('01','张三','英语','88');

insert into xs values('02','王五','数学','70');
```

```
insert into xs values('02','王五','语文','90');

insert into xs values('02','王五','英语','77');

commit;
select x.* from xs x;
```

答案如下：

```
--用 decode 实现

select x1.xno,x1.name,
  sum(x1.shuxue) shuxue,
  sum(x1.yuwen) yuwen,
  sum(x1.yingyu) yingyu
from   (select x.xno,x.name,

       decode(x.course,'数学',x.score) shuxue,

       decode(x.course,'语文',x.score) yuwen,

       decode(x.course,'英语',x.score) yingyu

       from xs x) x1
group by x1.xno,x1.name;

--用 case when then 实现

select x1.xno,x1.name,
  sum(x1.shuxue) shuxue,
  sum(x1.yuwen) yuwen,
  sum(x1.yingyu) yingyu
from  (select x.xno,x.name,
        case x.course

         when '数学' then x.score

        end  shuxue,
        case x.course

          when '语文' then x.score

        end  yuwen,
        case x.course

          when '英语' then x.score

        end  yingyu
        from xs x) x1
group by x1.xno,x1.name;
```

```
--drop table xs;
```

10 有 3 个表 S，C，SC：

S（SNO，SNAME）代表（学号，姓名）

C（CNO，CNAME，CTEACHER）代表（课号，课名，教师）

SC（SNO，CNO，SCGRADE）代表（学号，课号，成绩）

问题：

1 找出没选过“黎明”老师的所有学生姓名。（分别使用嵌套和非嵌套 2 种方式）

```
select sname from s where sno not in
(select sno from sc where cno in (select cno from c where cteacher='黎明'))
```

或

```
select S.SNAME from S,C,SC
where C.CNO=SC.CNO and SC.SNO=S.SNO and C.CTEACHER !='黎明'
```

2 列出 2 门以上（含 2 门）不及格学生姓名及平均成绩。

```
select S.SNAME,avg(SC.SCGRADE) from S,SC
where S.SNO=SC.SNO and SC.SCGRADE<60 group by S.SNO,S.SNAME having count(1)>=2;
```

或者

```
select s.sname, t.pj from s inner join
(select sno, avg(scgrade) pj from sc
where scgrade < 60 group by sno having count(*) >= 2) t on s.sno = t.sno;
```

3 即学过 1 号课程又学过 2 号课所有学生的姓名。

```
select S.SNAME from S
where S.SNO in
(select a.SNO from SC a,SC b where a.SNO=b.SNO and a.CNO=1 and b.CNO=2);
```

或

```
select sname from s where
    sno in (select sno from sc where cno = 1)
and
    sno in (select sno from sc where cno = 2)
```

11 有以下三张表 S，SC，C

学生表 S（S#，SNAME，AGE，SEX），其属性表示学生的学号，姓名，年纪，性别；
成绩表 SC（S#，C#，GRADE），其属性表示学号，课程号，成绩；
课程表 C（C#，CNAME，TEACHER），其属性表示课程号，课程名，任课教师名。
1）把 SC 表的每门课程的平均成绩插入到另一个已经存在的表 SC_C（C#，CNAME，AVG_GRADE）中，AVG_GRADE 为每门课程的平均成绩。

```
insert into SC_C (select SC.C#,C.CNAME,avg(SC.GRADE) from
SC,C where SC.C#=C.C# group by SC.C#,C.CNAME);
```

2）把 SC 表的 WU 老师的女学生选课元组删除。

```
delete from sc where sc.c# in

(select c.c# from c where c.TEACHER='wu 老师') and

sc.s# in

(select s.s# from s where s.sex='女')
```

或

```
delete from SC
where SC.S# in
  (select S.S# from S,SC,C

   where S.S#=SC.S# and SC.C#=C.C# and C.TEACHER='WU' and S.SEX='女')

and
SC.C# in(select C.C# from C where C.TEACHER='WU');
```

3) 查询前 5 名的学生成绩。

```
select t.S#,t.cj
from (select SC.S#,sum(SC.GRADE) cj,rownum rr from SC group
by SC.S# order by sum(SC.GRADE) desc) t
where t.rr between 1 and 5;
```

或者（rownum 本身不属于任何一张表）

```
select t.S#,t.cj
from (select SC.S#,sum(SC.GRADE) cj from SC group by SC.S#
order by sum(SC.GRADE) desc) t
where rownum between 1 and 5;
```

12 为管理岗业务培训信息，建立 3 个表：

S(S#,SN,SD,SA):学员表(学号，学员姓名，所属单位，学员年龄)；

C(C#,CN):课程表(课程编号，课程名称)；

SC(S#,C#,G):成绩表(学号，所选修的课程编号，学习成绩)。

1 使用嵌套查询，查询选修课程不为'C5'的学员姓名和所属单位。

```
select s.sn,s.sd from s
where s.S# in(select sc.S# from sc where sc.C#!='C5')
```

或

```
select s.sn,s.sd from s where s.s# not in(select sc.s# from
sc where sc.c#='C5')
```

2 使用嵌套查询，查询选修全部课程的学员姓名和所属单位

思路：在 SC 成绩表中按照学号分组，统计每组的行数，一行对应一个课程，行数就是课程数。

```
select s.sn,s.sd from s where s.s# in(select sc.s#  from sc
group by sc.s# having count(1)= (select count(1) from c))
```

3 查询选修了课程的学员人数。

```
select count(1) from (select distinct sc.s# from sc)
```

或

```
select count(distinct sc.s#) from sc;
```

4 查询选修课程数量超过 5 门的学员学号和所属单位。（使用嵌套和关联 2 种方式）

```
select s.s#,s.sd from s where s.s# in(select sc.s# from sc
group by sc.s# having count(1)>5)
```

或者

```
select s.s#,s.sd from sc,s where sc.s#=s.s# group by sc.s#,s.sd
having count(1)>5;
```

计算机网络：

1 分别写出 OSI 7 层模型和 TCP/IP 5 层模型

OSI 7 层模型：应用层、表示层、会话层、传输层、网络层、数据链路层、物理层
TCP/IP 5 层模型：应用层、传输层、网络层、数据链路层、物理层

2 http 协议属于哪层？tcp、udp 属于 tcp/ip 协议的哪层？

http 协议属于应用层。
tcp、udp 属于 tcp/ip 协议的传输层，tcp 是面向连接的服务，udp 是不面向连接的服务。

3 写出常见状态码

4 http 与 https 的区别

5 提交表单 get 和 post 方式的区别

Linux:

1 Linux 基础测试题 1

1、 linux 如何创建一个文件?

touch 文件名

2、 查看端口号 10000 的使用情况？

netstat –an|grep 10000

3、 如何查看磁盘的使用情况？

df -h

4、 如何显示 a.txt 文件的当前工作路径？

pwd a.txt

5、 如何强制删除一个 a.txt 的文件？

rm –rf a.txt

6、 如何强制结束一个名字为 monitor 的进程？

ps -ef | grep monitor

Kill -9<进程号>

7、 如何改变文件名字 a.txt 为 b.txt？

mv a.txt b.txt

8、 如何复制当前目录下的 a.txt 到/home/test 目录下？

cp ./a.txt /home/test

9、 如何压缩 file 文件为 file.tar.gz？

tar -zcvf file.tar.gz file

10、如何查看 run.log 文件的最后 10 行？

tail –10 run.log

11、如何查看一个服务器的 ip 地址

ifconfig

12、linux 如何查找一个目录下大于 50M 的文件

find 目录 –size +50M

13、 如何查看日志文件 xx.log，写出常用的日志查看两个命令

less xx.log view xx.log

14、还未打开文件情况下，查找字符串 hello 的命令

cat 文件名 | grep hello

15、已经打开文件的情况下，查找字符串 hello 的命令

往文件内容下面查找 /hello

往文件内容上面查找 ?hello

2 Linux 基础测试题 2

1、删除一个文件

rm –rf 文件名 / rmdir 空文件

2、创建一个空文件

touch 文件名

3、Linux 下的文本编辑工具是哪个?

vi 或者 vim

4、同时创建两个文件夹

mkdir 文件夹 1 文件夹 2

5、如何删除一个空文件夹？

rmdir 文件夹名

6、查看文件的方式？有什么不同

more ： 按回车一行，空格一页。不能上下键翻行。

less ： 按回车一行，空格一页。可以通过上下键上下翻行。按 q 就退出。

head -n ： 查看文件的前 n 行， n 表示你要看的行数。（光标默认跳出文件）

tail -n ： 查看文件的后 n 行，可能会引起循环（文件时刻会更新的情况下）（光标默认跳出文件）

cat ： 查看文件的所有内容（默认返回文件尾，并且跳出文件）

cat -n：查看文件的所有内容，并显示行数（光标默认返回文件尾，并且跳出文件）

查看日志常用 less ， view（空格就走一格）

less、view：？查找光标的上面，/查找光标的下面

7、当前目录下有 d101、d102，如何把 d101 目录下的 d1011 目录拷贝到 d102 目录下？剪切呢？

cp -R d101/d1011 d102

mv d101/d1011 d102

8、当前目录下有 d101、f101，将他们打包到当前目录下的 t101.tar 中？查看 t101.tar 中的内容？打包到/opt/d102/t102.tar 中

tar -cvf t101.tar d101 f101

tar –tvf t101.tar

tar -cvf /opt/d102/t102.tar d101 f101

10、将当前目录下的所有内容打成压缩包，有哪几种方式？

1:tar -cvf a.tar d01

gzip -9 a.tar

2:tar -zcvf a.tar.gz d01

11、如何查看命令的帮助？

man 命令;

或者

命令 --help;

12、如何释放一个压缩文件?

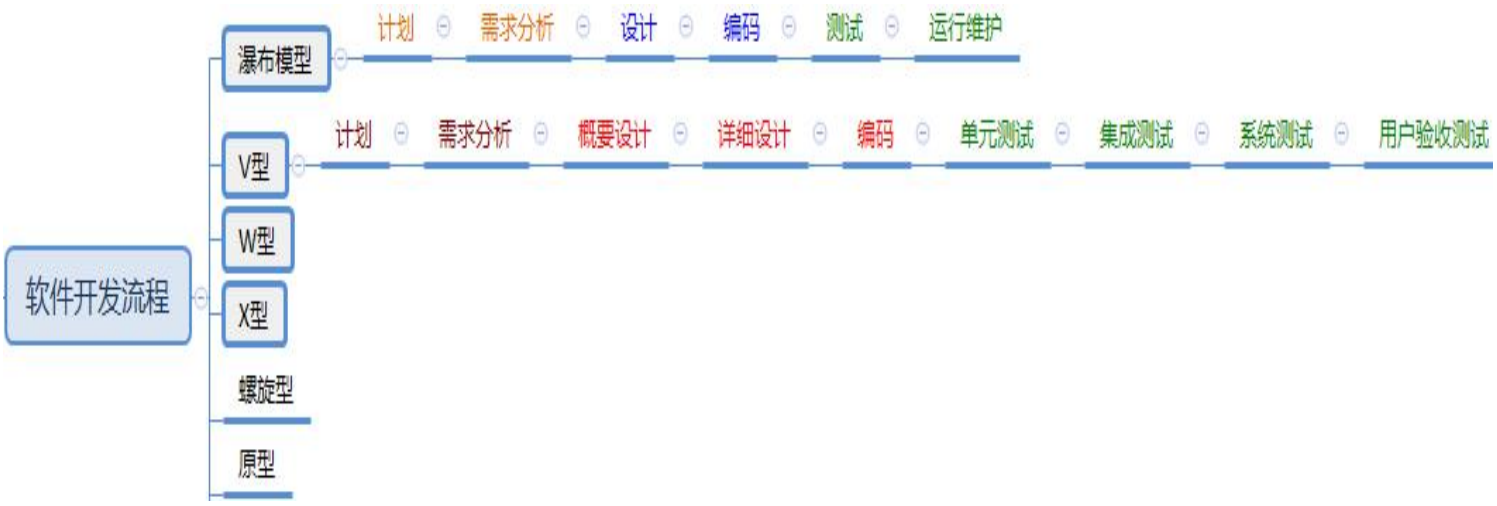
tar -zxvf a.tar.gz

13、当前目录下有 t101.tar.gz 包，将其释放到/opt/d101 目录下？

tar -zxvf t101.tar.gz -C /opt/d101

测试理论：

1 软件开发模型



2 软件测试流程

3 软件测试计划（也叫软件测试方案）有哪些内容？

1 版本号
2 概要描述
3 测试目的
4 测试范围
5 测试环境
6 测试工具
7 测试人员
8 任务分配、进度安排
9 测试风险

4 软件测试报告的内容？

1 数据统计
    1.1 人力投入
    1.2 用例执行情况
    1.3 bug 分类统计
2 测试风险
3 测试对象评估
4 测试结论

7 写出软件质量模型(测试点的万能公式，软件六大质量特性)

（UI）
功能性
可靠性（安全方面）
易用性
效率（性能方面）
维护性
可移植性（兼容性方面）

8 写出软件测试风险

1 进度风险（开发和测试的进度滞后）
2 软件质量风险（软件 bug 多，并且不能及时修复）
3 人员风险（离职，请假，生病）
4 需求变更风险
5 成本风险

9 写出所有的软件测试分类

按照普通流程分类：

单元测试（开发自测代码，也叫白盒测试）
集成测试（接口测试，灰盒测试）
ST 系统测试（黑盒测试，功能测试。
回归测试（上线前的一种测试，一般两三天时间，测试时，缺陷基本没有了）
UAT 用户验收测试（黑盒测试，往往 ST 测试第二轮时候启动）

按照技术分类：

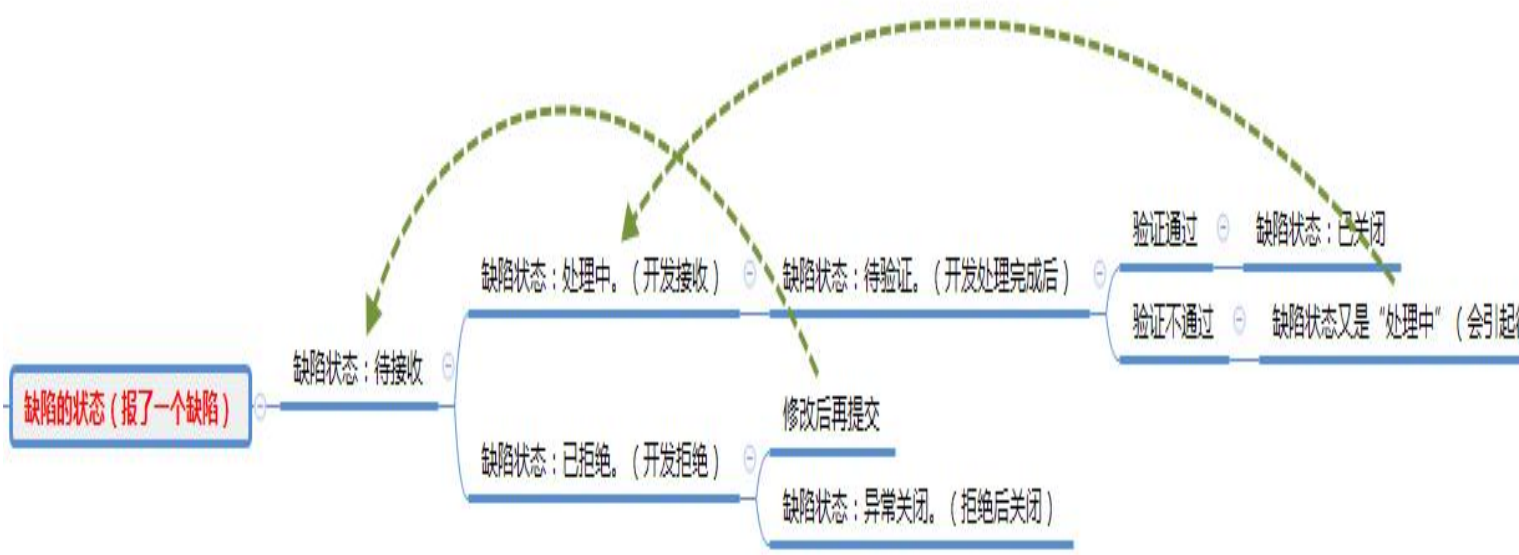
功能
性能（包含压力测试）测试项目的并发和承受压力的能力
自动化（自动化一般是针对功能比较完善的项目测试，跑自动化脚本代码来模仿人的鼠标操作，速度快，复用性强）

按照是否运行代码：

静态测试
动态测试（开发的代码评审）

10 写出案例状态和缺陷的状态(建议画出流程图的形式)

测试用例状态：未执行、测试通过、测试未通过、阻塞
缺陷的状态 ：待接收、已拒绝、异常关闭、处理中、待验证、已关闭



11 案例编写方法有哪些常见的 6 种方法？

1 等价类
2 边界值
3 错误推断法（根据经验判断，在容易出错的地方多设计案例）
4 判定表法（数学上的排列，先穷举，再挑选代表）
5 因果法
6 场景法（流程分析法）
7 特殊方法
    1 花瓣分析法
    2 正交分析法

12 设计案例的完整流程

1 产品经理把需求书发下来给开发和测试

2 测试先看一遍，进行需求分析。测试组长编写测试计划，并且分配测试任务给测试人员（此时开发也在进行需求分析）

3 产品经理再把测试和开发召集在一起，进行需求讲解（或者说需求评审），有问题可以直接问，如果发现需求有问题，也可以提出来，产品经理回去会修改。

4 讲完需求后，测试同事要进行测试场景的梳理和案例的编写了（xmind 和 Excel 就要用上了）。（此时开发在编写代码），案例主要有标题，步骤，预期结果，预置条件。编写案例方法主要有等价类边界值，错误推断法，判定表法。

5 之后就要进行案例评审了，评审时候有产品经理、测试同事、开发同事，评审时候一般产品经理、测试组长、对应模块的开发同事会提出一点意见，评审完之后，回去修改、补充一下案例。

6 修改完以后，有两种处理情况：

6.1 对大项目有时候要进行案例的第二次评审。

6.2 对小项目，在时间紧的时候，一般不会二审，但是要以邮件的形式把修改或者新增后的案例发出来，给领导看，并抄送给其他同事。

13 简述做好测试用例设计工作的关键点？

透彻了解需求、选用好的用例管理工具（xmind，excel，禅道）、做好用例评审、及时更新用例。

14 功能测试用例要详细到什么程度才算合格

1.将每一个功能点分解成可以进行测试（需要写明可操作的步骤）的测试点，步骤要简单明了。
2.用例每一条都是一个测试点，不要将多个测试点写到一个用例里，做到需求的全覆盖。
3.无法测试的点需要提出与程度沟通定出解决方案。
4.案例的基本原则是让新进来的测试同事，能很快的看懂并进入测试。

15 测试活动中发现需求有问题，咋办

把问题暴露给测试组长和开发组长，咨询他们意见，组长们再知会开发分组经理和项目经理，然后大家和产品经理一起探讨解决，需要改需求的地方就要改了。

16 在你以往的测试工作中,从事过哪方面的工作,最擅长哪方面？

自己写，主要写功能，自动化，性能，最擅长，功能测试的案例编写与测试，还有和开发进行 bug 的交流。

17 需求评审的重点

需求评审，产品经理把开发和测试召集在一起，进行需求的讲解。评审的重点在于把开发和测试的关于需求的不明白的地方进行解答。然后确定一些业务细节，数据流和状态的的细节。

18 功能测试和界面测试,你分别关注哪些内容

功能测试：关注的是业务流程，数据的正确性。
界面测试：关注的是界面的美观度，控件的布局，易用性，用户的体验性。

19 你在以前的工作中做过能测试吗？性能测试的流程和做好性能测试的关键？

流程：

1、测试准备 2、脚本开发与调试 3、测试执行 4、测试结果评估 5、测试后跟踪

关键：

1、资源的占用情况：查看资源的使用情况。资源包括 CPU，内存，硬盘等。

2、异常测试：

3、长时间运行：

4、查看业务响应时间

20 你以前的测试工作流程？

1 产品经理把需求书发下来给开发和测试

2 测试先看一遍，进行需求分析。测试组长编写测试计划，并且分配测试任务给测试人员（2 天时间）（此时开发也在进行需求分析）

3 过了 2 天，产品经理再把测试和开发召集在一起，进行需求讲解（或者说需求评审），有问题可以直接问，如果发现需求有问题，也可以提出来，产品经理回去会修改。
（需求讲解时间 0.5 天）

4 讲完需求后，测试同事要进行测试场景的梳理和案例的编写了（xmind 和 Excel 就要用上了），一共 5 个工作日。（此时开发在编写代码）

5 之后就要进行案例评审了，评审时候有产品经理、测试同事、开发同事，评审时候一般产品经理、测试组长、对应模块的开发同事会提出一点意见，评审完之后，回去修改、补充一下案例。（案例评审 0.5 天）

6 修改完以后，有两种处理情况：

6.1 对大项目有时候要进行案例的第二次评审。

6.2 对小项目，在时间紧的时候，一般不会二审，但是要以邮件的形式把修改或者新增后的案例发出来，给领

导看，并抄送给其他同事。（案例评审 0.5 天，修改案例 0.5 天，案例二审 0.5 天）

7 案例评审完就要开始测试了，一般测试环境开发搭建好（要说自己也会搭建，搭建流程背老师总结的）：

7.1 中型版本的测试一般分 2 轮：第一轮：5 天；第二轮：3 天；回归测试 2 天；（共 10 个工作日）。

8 回归测试完后，达到了上线标准，就会如期上线，一般当天晚上 12 点上线

21 在以往的测试工作中，获取到的工作经验；

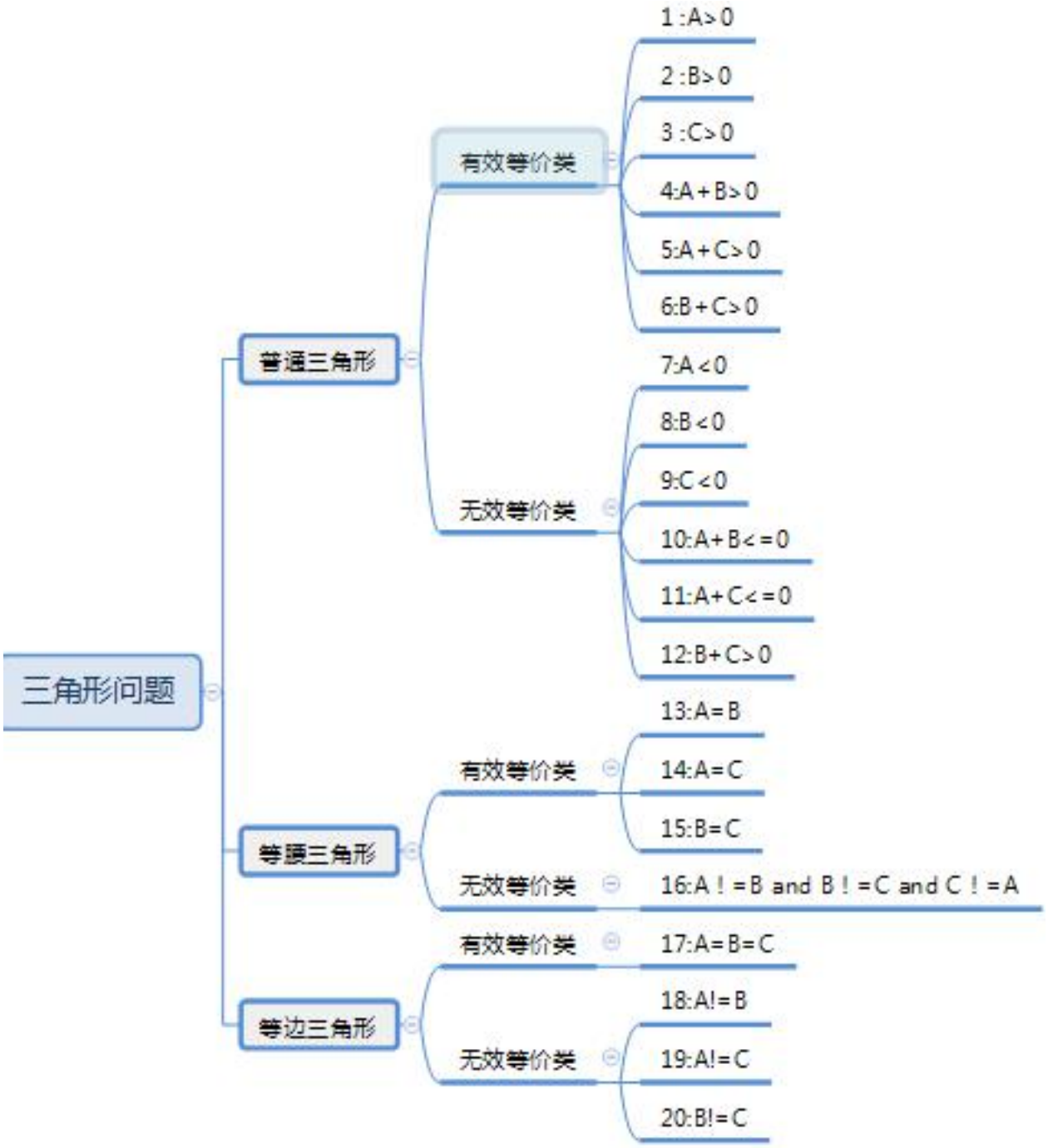
回答：在忙碌的工作当中，既充实，又有成就感。通过不断的测试，我的沟通能力、协调能力得到了提高，同时还收获了行业知识经验等，深刻感受到了团体精神的重要性。

22 一个程序读入 3 个数，把这三个数值看作一个三角形的 3 条边的长度值。这个程序要打印出信息，说明这个三角形是普通的、是等腰的、还是等边的。”利用等价类划分的方法，给出足够的测试用例。

测试点的提取：

经典笔试题-三角形问题

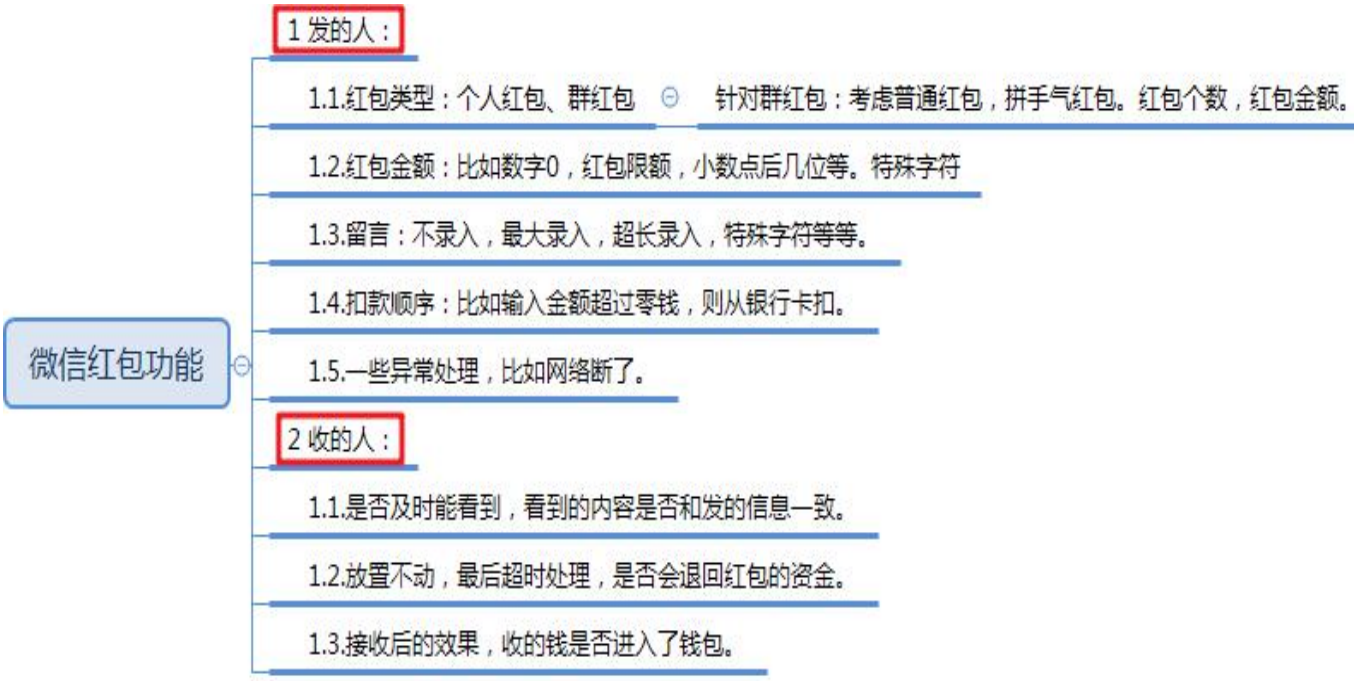
输入条件	有效等价类	无效等价类
是否是普通三角形	(A>0)，(1) (B>0)，(2) (C>0)，(3) (A+B>C)，(4) (B+C>A)，(5) (A+C>B)，(6)	(A≤0)，(7) (B≤0)，(8) (C≤0)，(9) (A+B≤C)，(10) (B+C≤A)，(11) (A+C≤B)，(12)
是否等腰三角形	(A=B)，(13) (B=C)，(14) (C=A)，(15)	(A≠B) and (B≠C) and (C≠A) (16)
是否等边三角形	A=B=C (17)	(A≠B)，(18) (B≠C)，(19) (C≠A)，(20)


三角形问题
普通三角形
有效等价类
1 :A>0
2 :B>0
3 :C>0
4:A+B>0
5:A+C>0
6:B+C>0
无效等价类
7:A<0
8:B<0
9:C<0
10:A+B<=0
11:A+C<=0
12:B+C>0
等腰三角形
有效等价类
13:A=B
14:A=C
15:B=C
无效等价类
16:A!=B and B!=C and C!=A
等边三角形
有效等价类
17:A=B=C
无效等价类
18:A!=B
19:A!=C
20:B!=C

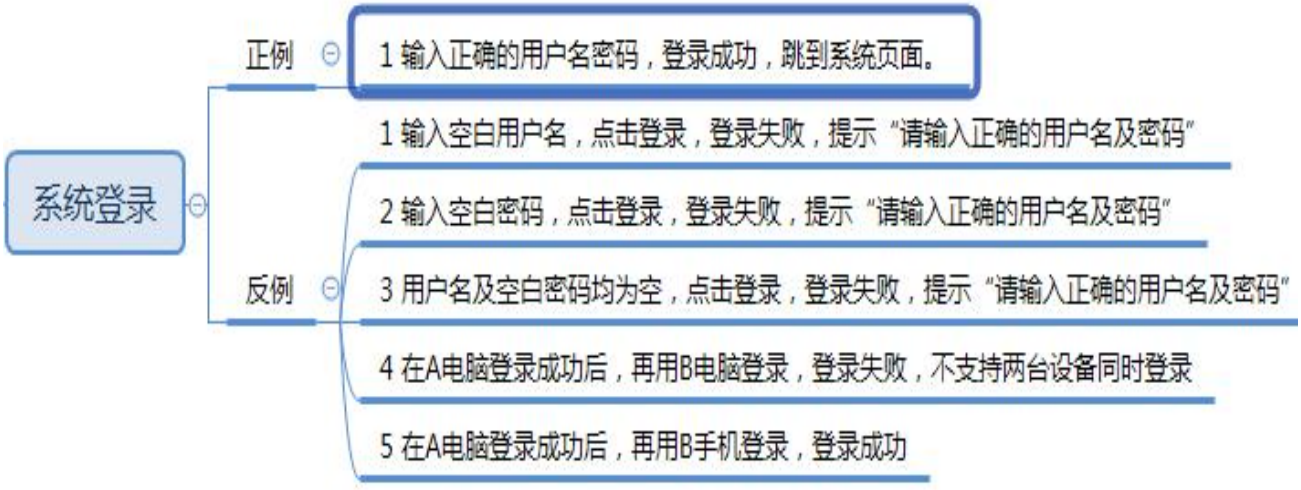
测试案例的梳理：

序号	输入（步骤） A，B，C	覆盖等价类	输出（预期结果）
1	【3，4，5】	（1），（2），（3），（4），（5），（6）	一般三角形
2	【0，1，2】	（7）	不能构成三角形
3	【1，0，2】	（8）	
4	【1，2，0】	（9）	
5	【1，2，3】	（10）	
6	【1，3，2】	（11）	
7	【3，1，2】	（12）	
8	【3，3，4】	（1），（2），（3），（4），（5），（6），（13）	等腰三角形
9	【3，4，4】	（1），（2），（3），（4），（5），（6），（14）	
10	【3，4，3】	（1），（2），（3），（4），（5），（6），（15）	
11	【3，4，5】	（1），（2），（3），（4），（5），（6），（16）	非等腰三角形
12	【3，3，3】	（1），（2），（3），（4），（5），（6），（17）	是等边三角形
13	【3，4，4】	（1），（2），（3），（4），（5），（6），（14），（18）	非等边三角形
14	【3，4，3】	（1），（2），（3），（4），（5），（6），（15），（19）	
15	【3，3，4】	（1），（2），（3），（4），（5），（6），（13），（20）	

23 针对微信红包功能提取测试点



24 对登录页面提取测试点，系统中已经有一个用户 lili，密码 1q2w3e，输入错误均提示“请输入用户名及密码”。



25 有个监控结果展示页面，请根据下图提取测试点

提取测试点：

26 针对线下商家的支付宝或者微信的二维码扫码器扫描支付功能，请提取测试点

1 UI：暂无
2 功能：
2.1 扫码器是否支持微信或者支付宝支付。支付成功后，是否及时收到扣款信息。
2.2 合作商户和非合作商户的扫描支付支持情况。
2.3 app 支付方式的选择，比如银行卡支付、账户余额、余额宝的顺序，支付时候是否按照优先级顺序来扣钱，所有支付方式的金额都不足以扣款金额时，是否有会提示金额不足，扣款失败信息。
2.4 扫码器扫描的二维码的面积，还有和手机的一个距离。
3 可靠（安全性）：
3.1 付款二维码会不会 1 分钟更新一次，已经付款的二维码，能不能进行第二次付款。
3.2 金额大于一定值时，是否会提示请输入支付密码。
3.3 二维码支付功能是否可以关闭和打开。
3.4 登录另外一个手机时，原来手机是否还支持付款。
4 易用：付款入口，很容易点击进去，一看就会使用 app 的付款功能。
5 效率：商家扫一扫后，交易能否很快完成，并且 app 收到付款通知信息。
6 维护：暂无。
7 可移植（兼容性）：
7.1 不同型号的手机是否均能支持付款。不同版本的微信或者支付宝 app 是否均能支持付款。
7.2 各种网络状态，包括飞行模式都应正常支持付款。

自动化性能：

1 你认为性能测试工作的目的是什么？做好性能测试工作的关键是什么？

目的：一、检验系统在大量用户下能否按要求运行，二、识别系统的弱点并进行调优
做好需求分析及测试计划，稳定的测试环境、有效的测试用例，做好结果分析及问题定位

2 LoadRunner 有哪些部件组成？

脚本生成器 vugen，压力调度和监控系统 controller,结果分析器 analysis，负载生成器 load generator 简称 LG

3 简述 LoadRunner 工作原理

LoadRunner 通过模拟上千万用户实施并发负载，实时性能监控的系统行为和性能方式来确认和查找问题。

4 简述手动关联的步骤。

先两个脚本进行对比，找出服务器返回的不同动态值，然后用动态值在生成日志中找出它的左右边界，再设置日志回放，用左右边界在回放日志中找出第一次返回动态值的地方，同时定位在脚本所在的位置，插入并设置关联函数（web_reg_save_param），参数化动态值。

5 LoadRunner 中如何实现多用户并发操作，需要进行哪些设置？

设置集合点，同时设置并发操作

6 简述性能测试的步骤和 LoadRunner 的使用步骤

性能测试的步骤：性能需求分析 性能测试计划 测试环境搭建 性能工具的引入 测试的执行 测试结果的分析 软件硬件配置调整与优化
LoadRunner 的使用步骤：选择协议 — 脚本录制设置 — 录制脚本—调试脚本—场景设计和运行—结果分析。

逻辑题：

1 蜗牛爬井一口井 10 米,蜗牛白天爬 3 米,夜晚滑 2 米.蜗牛几天能爬出?

8 天。

2 一楼到十楼的每层电梯门口都放着一颗钻石，钻石大小不一。你乘坐电梯从一楼到十楼，每层楼电梯门都会打开一次，只能拿一次钻石，问怎样才能拿到最大的那颗？

拿着笔和纸从第一楼开始时，记录钻石大小，第一楼纸上画的钻石和第二楼比较一次，得到最大的钻石，并又把该钻石记录在纸上，每一次这样比较，最后从纸上就可以知道哪一楼的钻石最大，共比较 9 次后，就可以去拿最大钻石了。（此题为开放题，每次比较时，你也可以在心里记住哪个大点）

3 烧一根不均匀的绳,从头烧到尾总共需要 1 个小时。现在有若干相同材质的绳子,如何用它来判断半个小时？如何用烧绳的方法来计时一个小时十五分钟呢？

答：
1 绳从两头烧。
2 用 a、b、c 三条绳子。
2.1 a 绳从两头烧，同时 b 绳从一头烧，当 a 绳烧尽时，同时灭掉 b 绳（半个小时）；
2.2 同时 c 绳从两头烧，在 c 绳烧尽时（半个小时）；
2.3 同时 b 绳从两头烧（15 分钟）。结束时即为 1 小时 15 分钟。

4 一间囚房里关押着两个犯人。每天监狱都会为这间囚房提供一罐汤，让这两个犯人自己来分。起初，这两个 人经常会发生争执，因为他们总是有人认为对方的汤比自己的多。后来他们找到了一个两全其美的办法：一个人分汤，让另一个人先选。于是争端就这么解决了。可是，现在这间囚房里又加进来一个新犯人，现在是三个人来分汤。必须寻找一个新的方法来维持他们之间的和平。该怎么办呢？

分汤的人都是最后取汤的人，每次分汤都是不一样的人。

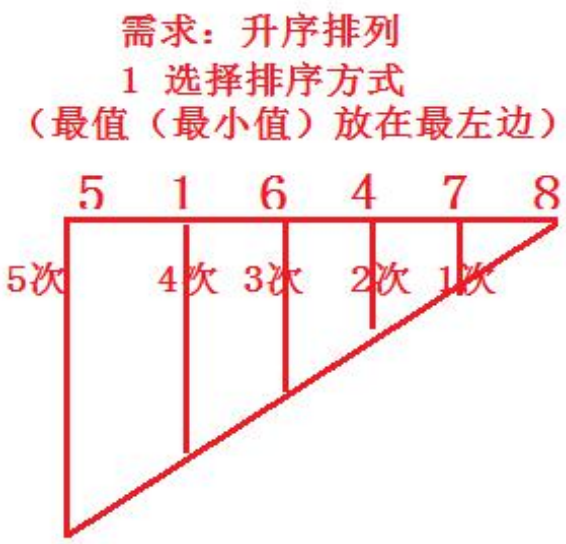
编程类笔试题：

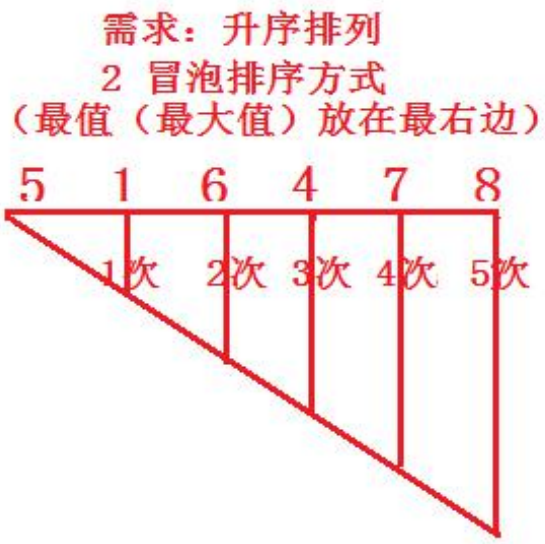
1 写出获取最值的方法

```
//获取数组中的最大值。
	public  int getMax(int[] arr){
		int max = arr[0];
		for(int x=1; x<arr.length; x++){
			if(arr[x]>max){
				max = arr[x];
			}
		}
		return max;
	}

//获取数组中的最小值。
	public  int getMin(int[] arr){
		int min = 0;
		for(int x=1; x<arr.length; x++){
			if(arr[x]<arr[min])
				min = x;
		}
		return arr[min];
	}
```

2 写出选择排序，冒泡排序

```
/*
对给定数组进行从小到大排序。
{5,1,6,4,7,8}

*/
```





//选择排序。内循环结束一次，最值出现头角标位置(最左边)（系统提供方法 Arrays.sort(arr)）

```
public  void selectSort(int[] arr){

        for (int x=0; x<arr.length-1 ; x++){

                for(int y=x; y<=arr.length-1; y++){

                        if(arr[x]>arr[y]){
                                int temp = arr[x];
                                arr[x] = arr[y];
                                arr[y]= temp;
                        }
                }
        }
}
```

//冒泡排序,内循环结束一次,最值出现在尾角标位置（最右边）

```
public  void bubbleSort(int[] arr){

        for(int x=0; x<arr.length-1; x++){
```

```
    //-x:让每一次比较的元素减少，-1：避免角标越界。
            for(int y=0; y<arr.length-1-x; y++){
                    if(arr[y]>arr[y+1]){
                            int temp = arr[y];
                            arr[y] = arr[y+1];
                            arr[y+1] = temp ;
                    }
            }
    }
}
```

3 写出字符串反转的方法

```
//字符串反转函数。(s= "abcd")
        public String charReverse(String s){
                int len=s.length();
                StringBuilder sb=new StringBuilder();
                for(int x=len-1;x>=0;x--){
                        char c=s.charAt(x);
                        sb.append(c);
        }
                s=sb.toString();
                return s;
         }
```

4 求一个数的所有素数个数，比如输入 100，就要求出 1 到 100 之间的所以素数个数。

//素数也叫质数，是大于等于 2 的自然数，并且素数只能被 1 和本身整除，就是一个素数只存在这种情况 n=n*1。

解题思路：2 是最小的素数，所以当传入的数是 10 时，就要从 3 开始一直到 10 都要去判断。当数为 i 时（3<=i<=10）,i 如果能被 2 至 i-1 的任何一个值整除时，此 i 就不是素数了。

```
public  int countSuShu(int x){
    int count=0;
    for(int i=3;i<=x;i++){
       for(int j=2;j<i;j++){
          if(i%j==0){
             count++;//count 记录非素数的个数
             break;//break 中断本层（内层）循环，continue 是继续下一次循环。
          }
       }
    }
   return x-(count+1);//1 不是素数，所以非素数个数 count 要加上 1
}
```

5 判断一个数是否为闰年

```
//能被 4 整除但不能被 100 整除，或能被 400 整除的年份即为闰年
public static String qiuRunNian(int date){
   if((date%4==0&&date%100!=0)||date%400==0){
      return "闰年";
   }
   return "平年";

}
```

6 打印出 99 乘法表

```
1*1=1
2*1=2 2*2=4
3*1=3 3*2=6 3*3=9
4*1=4 4*2=8 4*3=12 4*4=16
5*1=5 5*2=10 5*3=15 5*4=20 5*5=25
6*1=6 6*2=12 6*3=18 6*4=24 6*5=30 6*6=36
7*1=7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49
8*1=8 8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64
9*1=9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81
```

```
public void jiuJiu(){
      for(int i=1;i<=9;i++){
         for(int j=1;j<=i;j++){
            //当次内循环不换行
            System.out.print(i+"*"+j+"="+i*j+" ");
         }
         //内循环每次结束后换行
         System.out.println("");
      }
 }
```

7 利用存储过程打印 99 乘法表

```
--1.1 存储过程创建语句
create or replace procedure jiujiu_p as
begin
  for i in 1..9
  loop
     for j in 1.. i
     loop
        --当次内循环不换行
        dbms_output.put(i||'*'||j||'='||i*j||' ');
     end loop;
     --内循环结束后换行
     dbms_output.put_line('');
  end loop;
end;

--1.2 存储过程调用语句
declare
begin
  jiujiu_p;
end;
```

```
--2.1 存储函数创建语句（这种方式不需要掌握）
create or replace function jiujiu_f return number as
begin
  for i in 1..9
  loop
    for j in 1.. i
    loop
      --当次内循环不换行
      dbms_output.put(i||'*'||j||'='||i*j||' ');
    end loop;
    --内循环结束后换行
    dbms_output.put_line('');
  end loop;
  return 0;--为了返回值而返回值
end;

--2.2 存储函数调用语句
declare
  a number;
begin
  a := jiujiu_f;
end;
```