## Java笔试题整理2

#### 一：选择题

1. Linux中切换使用者身份的命令是：su
2. 软件开开发中关于安全说发错误的是：软件出了安全问题都是程序员开发的问题
3. 743标识的文件的权限是：rwxr---wx

解析：linux系统的权限一般是用 1-3 位数字代表文件所有者的权限，4-6 位数字代表同组用户的权限，7-9 数字代表其他用户的权限。具体的权限是由数字来表示的，读取的权限等于 4，用 r 表示；写入的权限等于 2，用 w 表示；执行的权限等于 1，用 x 表示；通过4、2、1的组合，得到以下几种权限：0（没有权限）；4（读取权限）；5（4+1 | 读取+执行）；6（4+2 | 读取+写入）；7（4+2+1 | 读取+写入+执行）。

#### 二：判断题

1. Java中使用了垃圾回收：√
2. ls -l 是查看目录下的所有文件：×

解析：-a 全部的文件，连同隐藏文件（ 开头为 . 的文件） 一起列出来

-l长格式的形式查看当前目录下所有可见文件的详细属性

1. 程序能够按照人们的想法执行是因为存储在内存中：×

解析：程序是存储在外存（如硬盘，u盘，光盘），执行的时候必须先加载到内存中

#### 三：简答题

1：进程和线程的区别：

1.线程是程序执行的最小单位，进程是操作系统分配资源的最小单位；

2.一个进程由一个或者多个线程组成，一个进程中至少有一个主线程；

3.不同进程进程之间内存空间是独立的，同一个进程下的各个线程之间共享程序的内存空间；

4.线程上下文切换要比进程上下文切换快很多；

2：线程的5种状态：

1.新建状态：新创建一个线程对象。

2.就绪状态：线程对象被创建后，其他线程调用了该对象的start方法，该状态的线程获取到了除CPU以外的资源，位于可运行线程池中，等待获取CPU的使用权。

3.运行状态：就绪状态的线程获取到CPU使用权，执行程序代码。

4.阻塞状态：线程因为某种原因放弃了CPU使用权，暂时停止运行，直到线程进入就绪状态，才有机会转到运行状态。

5.死亡状态：当一个线程正常执行完，那么就进入死亡状态。系统就会回收这个线程占用的资源。

3：作为软件开发人员需要哪些能力？

1.扎实的计算机专业知识以及编程知识；

2.良好的编码能力、自觉的规范意识；

3.良好的语言沟通，表达能力以及团队精神和协作能力；

4.持续不断的学习精神，快速的适应新的开发环境；

5.吃苦耐劳的精神，每天的事情尽量当天完成，不拖泥带水；

4: 重写和重载的区别？

重写是子类对父类的非私有的方法的实现过程进行重新编写, 返回值和形参都不能改变。重写注意以下几点：父类私有的方法不可以被重写，重写方法和被重写方法必须具有相同的方法名和参数列表，重写方法返回值必须和被重写方法的返回值类型相同或者是其子类型，重写方法不能缩小方法的访问权限。

重载是在同一个类中可以存在多个方法名相同的方法，通过参数的数量和类型区分。重载注意以下几点：必须在同一个类下，方法名相同，方法参数个数或者参数类型不同，与方法返回值返回类型方法修饰符无关系。

5：while循环和do while循环的区别？

while是先判断，若为真则执行循环，若为假则不循环；

dowhille是无论为真还是假，先执行一遍代码，然后再进行判断是否为真或者是否为假，若为真，则继续执行循环；

#### 四 程序题：

1: public static void main(String[] args) {

int a;

{

a = 23;

System.out.print(a);

}

{

a = 45;

System.out.print(a);

}

a = 81;

System.out.print(a);

}

输出结果是?

234581

2:给了一段js代码, 问那句代码有问题, 并改正

3:public static void main(String[] args) {

int x = 1;

int y = 1;

if(x++ == 2 && ++y ==2){

x=9;

}

System.out.println("x:"+x+", y:"+y);

}

输出结果是?

x:2, y:1

#### 五: 编程题:

1: 请手写冒泡排序:

public void bubbleSort(int[] nums) {

int length = nums.length;

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

for (int j = 0; j < length; j++) {

if (nums[j] > nums[j + 1]) {

int tem = nums[j];

nums[j] = nums[j + 1];

nums[j + 1] = tem;

}

}

}

}