**1.“child method” “parent smethod”**

**2.10,20 8,5 10,20,8,5**

**二，改错题**

**1.list list =new arraylist();**

**2.string类是被final修饰的，不能被继承**

**3.map 的类型 k-value对应关系**

**4.1个类不能同时继承2个类**

**5.amethod不是静态方法，所以不能直接调用。**

**二，解答题1.JAVA的基本数据类型有哪几种? String是不是基本数据类型，他们有何区别。**

byte，short，int，char，long，float，double，boolean这些是java的基本数据类型。

String不是基本数据类型，它是java.lang包下的一个类。算复合数据类型。

两者的区别是，基本数据类型是数据的最小单位，不可再分了。而复合数据类型则可以再分成基本数据类型的数据。

**2.怎样将数值类型转化为数字类型(interger、double) 怎样将数字类型转化为字符串类?怎样取得小数点的前两位，并且进行四舍五入？**

1.用integer.parsent([string])

2.integer. to string (2);

3.double n=2.1356;

math. round(moth.floor(123.45)%100)

NumberFormat formatter = new DecimalFormat("#0.00");  
double money = 100550000.75;  
formatter.format(money);

**3. 简述abstract class和interface的区别**

前者是抽象类，后者是接口。  
抽象类和接口都可以被继承，不同的是，接口内的方法不能实现，需要继承的类来具体实现其方法，而抽象类中的方法可以被具体实现，并且继承的类可以重写其方法。一个类可以继承一个抽象类，但是可以继承多个接口。通常，接口被运用于实现比较常用的功能，便于日后维护或者添加删除方法，而抽象类更倾向于充当公共类的角色。不适于在日后重新对里面的代码进行修改。

**4.sting和stringbuffer的区别**

string是字符串常量，而stringbuffer是变量，字符串连接操作中StringBuffer的效率要明显比String高: String对象是不可变对象,每次操作Sting 都会重新建立新的对象来保存新的值. StringBuffer对象实例化后，只对这一个对象操作。

**5.简述逻辑操作(&,|,^)与条件操作(&&,||)的区别**

a.条件操作只能操作布尔型的,而逻辑操作不仅可以操作布尔型,而且可以操作

数值型

b.逻辑操作不会产生短路.

**大题**

**1.javascript校验数字型**

var re=/^\d{1,8}$|\.\d{1,2}$/;

var str=document.form1.all(i).value;

var r=str.match(re);

if (r==null)

{

sign=-4;

break;

}

else{

document.form1.all(i).value=parseFloat(str);

}

**2.java正则表达式验证邮件地址**

bool is email(string email) {

//邮件地址表达式

return regex.is match(mail,@" ^([1w-\.]+@((\[[0-9][a-z A-z][0-4]{1,3}};

原理：从字母a-z，A-Z,数字0-9，加@，有点（“.”）就可以验证

**3.利用递归方法求5!**

public class Test {  
 public static void main(String args[]) {  
 int x = 5;  
 int rs = Fac(x);  
  
 System.out.println("" + x + "! = " + rs);  
 }  
  
 public static long Fac(int x) { /\*[阶乘](http://zhidao.baidu.com/search?word=%BD%D7%B3%CB&fr=qb_search_exp&ie=gbk)算法\*/  
 if (x > 1)  
 return (x \* Fac(x - 1)); /\*[递归](http://zhidao.baidu.com/search?word=%B5%DD%B9%E9&fr=qb_search_exp&ie=gbk)\*/  
 else  
 return 1;  
 }  
}