1.包装类（Integer）



1>.将字符串变为一个数字

将数子变为一个对像

Integer s = **new** Integer(100); //讲一个数字包装成一个对象(s 就是代表数字)

**下面的就是将字符串变为数字**

Integer s1 = **new** Integer("321"); //同上

Integer i = Integer.*parseInt*("123");（或者Int i都是可以的）

System.*out*.println(s+" "+s1);

2>.将数字转化为字符串

String s2 = 123+"";

System.*out*.println(s2);

（像我们之前利用一个for循环改变String字符中的东西）

2.自动装箱

Integer s = **new** Integer(100);

Integer s1 = 100; //jdk 5.0 新增的内容

//(自动装箱：编译器自动给我我们装箱也就是编译器直接

//将100变为了 new Integer(100);

3.自动拆箱

int s3 = s1;这里其实代表的是s1.intValue();

int s2 = new Integer(100) //这里就是自动拆箱

实际上这就是 new Integer(100).intValue();

4.确定 == 到底是怎么用

Integer s4 = 1001; //[-128,127]之间的数字仍然在基本类型之间

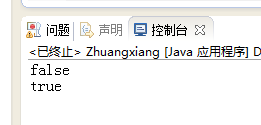
Integer s5 = 1001;

System.*out*.println(s4 == s5); //false （Integer 表示的是类型，== 也是看类型是否是相同的）**属性是相同的但是属于不同对象**

Integer s6 = -12; //[-128,127]之间的数字仍然在基本类型之间

Integer s7 = -12;

System.*out*.println(s6== s7); //ture （基本类型是相同的也就是数字是相同的）



2.Date时间类

1》

Date a = **new** Date();

**long** b = System.*currentTimeMillis*(); //时间使用毫秒来计算的所以是long类型 （这里输出的是当前时间）

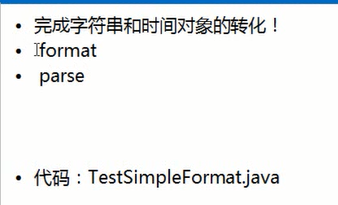
**long** c = a.getTime();这里的a就是代表的long

Date a1 = **new** Date(2000)；

System.*out*.println(a1);

System.*out*.println(b);

System.*out*.println(c);

 2》

3》将时间转化为字符串 方法如下

DateFormat s = **new** SimpleDateFormat("yyyy年MM月dd日"); //抽象类（必须对它里面的时间进行转化）

//或者 yy年MM月dd日 或者yy-MM-dd hh:mm:ss 或者yy-MM-dd

Date s1 =**new** Date(111111111); //输入一个时间

String str = s.format(s1);//进行转化 将时间转化为 字符串

System.*out*.println(str);

4》将字符串转化为时间

//将字符串转化为时间

String a = "2014,8,8";

DateFormat a1 = **new** SimpleDateFormat("yyyy,MM,dd"); //必须同上面的格式一样

下面这一个异常分析是自己出现的

**try** {

Date a2 = a1.parse(a);

System.*out*.println(a2); //时分秒 默认是0

} **catch** (ParseException e) {

// **TODO** 自动生成的 catch 块

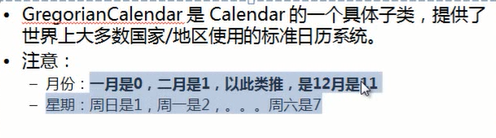
e.printStackTrace();

} //设置一个时间对象

}

3.Calender日期类

1》解释与国外不同的地方



2.将我们输入的数字按照日期的方式输出

**public** **static** **void** main(String args[]){ 父类

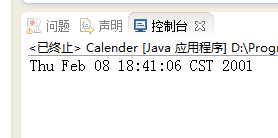
Calendar a = **new** GregorianCalendar(); 子类

a.set(2014,Calendar.*FEBRUARY*,8,24,1,3); //或者数字都是可以的

Date s = a.getTime();

System.*out*.println(s);

}**给的是当前时刻的 时分秒**



**set 还可以这么用 （ 可以单独定义）**

//a.set(2014,Calendar.FEBRUARY,8,24,1,3); //或者数字都是可以的

a.set(2014,6 ,8,24,1,3); 年 月 日 时 分 秒

//或者set还可以这么用

a.set(Calendar.*YEAR*,2001);

a.set(Calendar.*MONTH*,1);//这里就是代表的二月

3》日期计算（不可以用Date进行计算）

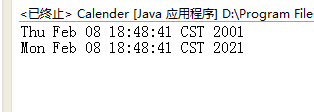
Calendar a = **new** GregorianCalendar();

a.set(2014,Calendar.FEBRUARY,8,24,1,3); //或者数字都是可以的

a.add(Calendar.*YEAR*,20 ); //不能使用Date 进行计算 必须是日期的 这里表示的是年加上20年

Date s1 = a.getTime();

System.*out*.println(s1);



4.制作一个日历

1》想法：先制作出一个基本的形状来，使用String 字符的日期，将它转化为日期Date

String time = "2014-8-1";

DateFormat zhuan = **new** SimpleDateFormat("yyyy-MM-dd");

**try** {

Date a = zhuan.parse(time);

Calendar ca = **new** GregorianCalendar();

ca.setTime(a); //将日期导入到日历当中去

//捕获我们的日期中的信息

System.*out*.println(ca.get(Calendar.*DAY\_OF\_WEEK*)); //试验得到日期中的 周几（表示这个；月开头是周几）

System.*out*.println("日\t一\t二\t三\t四\t五\t六");

System.*out*.printf("\n");

**for**(**int** i = 1;i<32;i++){

System.*out*.print(i+"\t");

**if**(i%7==0){

System.*out*.printf("\n");

}

}

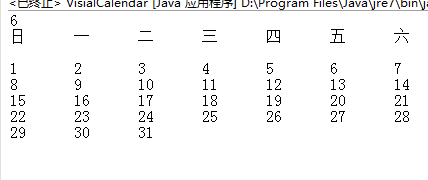
} **catch** (ParseException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}

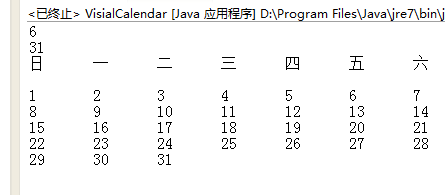
2.》现在的日历出来是这样的



3》我们现在知道了它的开始时星期 日 （外国人是 日），现在我们需要的是这个月到底有多少天

System.*out*.println(ca.getActualMaximum(Calendar.*DATE*));

4》此时的到的是



5》这里我们可以很简单的解决最后一个数字的问题，只要一个变量就可以解决了

**int** zuihou = ca.getActualMaximum(Calendar.*DATE*);

System.*out*.println("日\t一\t二\t三\t四\t五\t六");

System.*out*.printf("\n");

**for**(**int** i = 1;i<=zuihou;i++){

System.*out*.print(i+"\t");

**if**(i%7==0){

System.*out*.printf("\n");

}

6》现在我们已经解决了最后一个数字的问题，现在我们要解决的是第一个数字的开头

5.File类

1.用来引入文件或者是目录

观察下面的程序的用法

//引入java文件

File a = **new** File("E:/eclipse工作空间/新类/src/新类/Baozhuanglei.java");

//引入目录

File a1 = **new** File("E:/eclipse工作空间/新类/src");

//通过父的目录引入文件 a1 是父类 (只要是之前的父类以上都是可以的)

File a2 = **new** File(a1,"Calender.java");

2.如何判断是不是一个目录还是文件

**if**(a.isFile()){

System.*out*.println("是一个文件");

}

**if**(a1.isDirectory()){

System.*out*.println("是一个目录");

}

3.创建一个java文件（先将它的路径写下类）createNewFile()

创建的java是可以编写的，但是我们现在还没有学习过

File a3 = **new** File(a1,"Calender22222222.java");

**try** {

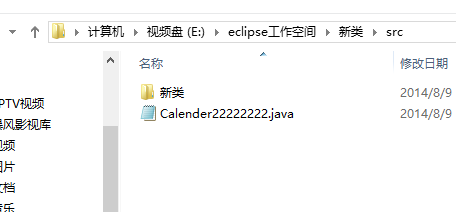
a3.createNewFile();

} **catch** (IOException e) {

// **TODO** 自动生成的 catch 块

e.printStackTrace();

}



4.删除一个文件或者目录（这里之前必须知道他的文件的路径，也就是必须写下它的路径来）delete();

File a3 = **new** File(a1,"Calender22222222.java");

File a4 = **new** File("F:/新建文件夹");

a3.delete(); 这个是删除文件

a4.delete() 这个是删除目录

5.创建一个目录（包括多层目录mkdirs()和单个目录mkdir()）

//下面这个不可以创建（因为有点多了）

File a6 = **new** File("F:/wo/shi/wjo");

a6.mkdir();//当我们要创建的目录前面也有需要创建的目录是后那么我们不可以使用这个创建目录

//下面这个将会创建的

File a5 = **new** File("F:/zhangyujin");

a5.mkdir();

//下面子这个会创建多个目录

File a7 = **new** File("F:/wo/shi/wjo/slasjdhfkhjk");

a7.mkdirs();

6.利用Filie打印一个树状的结构目录结构