Java 核心技术

第八章 Java常用类

第五节 格式化(Format)相关类

华东师范大学 陈良育

格式化类(1)

- java.text包java.text.Format的子类
  - NumberFormat：数字格式化，抽象类
    - DecimalFormat
  - MessageFormat：字符串格式化
  - DateFormat：日期/时间格式化，抽象类
    - SimpleDateFormat
- java.time.format包下
  - DateTimeFormatter

格式化类(2)

- NumberFormat：数字格式化，抽象类
  - DecimalFormat 工厂模式
  - 例如：将1234567格式化输出为1,234,567

- 查看DecimalFormaterRuleTest.java
- 查看DecimalFormatTest.java

格式化类(3)

- MessageFormat：字符串格式化
  - 支持多个参数-值对位复制文本
  - 支持变量的自定义格式
  - 例如将”**Hello {1}**”根据变量值格式化为Hello World
- 查看MessageFormatTest.java

格式化类(2)

- DateFormat：时间格式化，抽象类
  - SimpleDateFormat 工厂模式
  - parse：将字符串格式化为时间对象
  - format：将时间对象格式化为字符串
  - 如将当前时间转为化YYYY-MM-DD HH24:MI:SS输出

- 查看SimpleDateFormatTest.java

格式化类(2)

- java.time.format.DateFormatter： 时间格式化
  - JDK 8 发布，线程安全(vs SimpleDateFormat 线程不安全)
  - ofPattern: 设定时间格式
  - parse：将字符串格式化为时间对象
  - format：将时间对象格式化为字符串
  - 如将当前时间转为化YYYY-MM-DD HH24:MI:SS输出

- 查看DateFormatterTest.java

总结

- 三种格式化
  - 数字格式化
  - 字符串格式化
  - 时间格式化
- 学会查阅相关的API文档，并多进行代码练习

代码(1) DecimalFormaterRuleTest


```java
import java.text.DecimalFormat;

public class DecimalFormaterRuleTest {
    public static void main(String[]args){

        DecimalFormat df1,df2;

        System.out.println("整数部分为0的情况，0/#的区别");
        // 整数部分为0 ， #认为整数不存在，可不写； 0认为没有，但至少写一位，写0
        df1 = new DecimalFormat("#.00");
        df2 = new DecimalFormat("0.00");

        System.out.println(df1.format(0.1)); // .10
        System.out.println(df2.format(0.1)); // 0.10

        System.out.println("小数部分0/#的区别");
        //#代表最多有几位，0代表必须有且只能有几位
        df1 = new DecimalFormat("0.00");
        df2 = new DecimalFormat("0.##");

        System.out.println(df1.format(0.1)); // 0.10
        System.out.println(df2.format(0.1)); // 0.1

        System.out.println(df1.format(0.006)); // 0.01
        System.out.println(df2.format(0.006)); // 0.01
```

代码(2) DecimalFormaterRuleTest


```
        System.out.println("整数部分有多位");
        //0和#对整数部分多位时的处理是一致的 就是有几位写多少位
        df1 = new DecimalFormat("0.00");
        df2 = new DecimalFormat("#.00");

        System.out.println(df1.format(2)); // 2.00
        System.out.println(df2.format(2)); // 2.00

        System.out.println(df1.format(20)); // 20.00
        System.out.println(df2.format(20)); // 20.00

        System.out.println(df1.format(200)); // 200.00
        System.out.println(df2.format(200)); // 200.00


    }
}
```

代码(3) DecimalFormatTest.java


```java
import java.text.DecimalFormat;

public class DecimalFormatTest {

    public static void main(String[] args) {
        DecimalFormat df1 = new DecimalFormat("0.0");

        DecimalFormat df2 = new DecimalFormat("#.#");

        DecimalFormat df3 = new DecimalFormat("000.000");

        DecimalFormat df4 = new DecimalFormat("###.###");

        System.out.println(df1.format(12.34)); //12.3

        System.out.println(df2.format(12.34)); //12.3

        System.out.println(df3.format(12.34)); //012.340

        System.out.println(df4.format(12.34)); //12.34
```

代码(4) DecimalFormatTest.java


```
        DecimalFormat df5 = new java.text.DecimalFormat("0.00");// 保留2位小数
        double d1 = 123456789.123456;
        double d2 = 987654321.987654321;

        System.out.println("format1_d1=" + df5.format(d1));// 输出format1_d1=123456789.12
        System.out.println("format1_d2=" + df5.format(d2));// format1_d2=987654321.99
                                                          // 四舍五入

        DecimalFormat df6 = new DecimalFormat("#,##0.00");
        System.out.println("format2_d1=" + df6.format(d1));// 输出：format2_d1=123,456,789.12
        System.out.println("format2_d2=" + df6.format(d2));// 输出：format2_d2=987,654,321.99
                                                          // 四舍五入

    }

}
```

代码(5) Test.java

```
import java.text.DecimalFormat;

public class Test{
    public static void main(String[] args){
        double pi=3.1415927;//圆周率
        //取一位整数
        System.out.println(new DecimalFormat("0").format(pi));//3
        //取一位整数和两位小数
        System.out.println(new DecimalFormat("0.00").format(pi));//3.14
        //取两位整数和三位小数，整数不足部分以0填补。
        System.out.println(new DecimalFormat("00.000").format(pi));//03.142
        //取所有整数部分
        System.out.println(new DecimalFormat("#").format(pi));//3
        //以百分比方式计数，并取两位小数
        System.out.println(new DecimalFormat("#.##%").format(pi));//314.16%

        long c=299792458;//光速
        //显示为科学计数法，并取五位小数
        System.out.println(new DecimalFormat("#.#####E0").format(c));//2.99792E8
        //显示为两位整数的科学计数法，并取四位小数
        System.out.println(new DecimalFormat("00.####E0").format(c));//29.9792E7
        //每三位以逗号进行分隔。
        System.out.println(new DecimalFormat(",###").format(c));//299,792,458
        //将格式嵌入文本
        System.out.println(new DecimalFormat("光速大小为每秒,###米").format(c)); //光速大小为每秒299,792,458米
    }
}
```

代码(6) TwoDigitsTest.java


```java
import java.math.BigDecimal;
public class TwoDigitsTest {

    public static void main(String[] args) {
        double   f   =   111231.5585;
        BigDecimal   b   =   new   BigDecimal(f);
        double   f1   =   b.setScale(2,BigDecimal.ROUND_HALF_UP).doubleValue();
        System.out.println(f1); //111231.56

        DecimalFormat   df   =new DecimalFormat("#.00");
        String f2 = df.format(f);
        System.out.println(f2); //111231.56

        String f3 = String.format("%.2f",f);
        System.out.println(f3); //111231.56

        NumberFormat ddf1=NumberFormat.getInstance() ;
        System.out.println(ddf1.getClass().getName());
        ddf1.setMaximumFractionDigits(2);
        String f4= ddf1.format(f) ;
        System.out.println(f4);  //111,231.56
    }
}
```

代码(7) MessageFormatTest.java


```java
import java.text.MessageFormat;

public class MessageFormatTest {

    public static void main(String[] args) {
        String message = "{0}{1}{2}{3}{4}{5}{6}{7}{8}{9}{10}{11}{12}{13}{14}{15}{16}";

        Object[] array = new Object[]{"A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q"};

        String value = MessageFormat.format(message, array);

        System.out.println(value);

        message = "oh, {0,number,#.##} is a good number";

        array = new Object[]{new Double(3.1415)};

        value = MessageFormat.format(message, array);

        System.out.println(value);
    }
}
```

代码(8) SimpleDateTest.java

```java
import java.text.DateFormat;

public class SimpleDateTest {

    public static void main(String[] args) {

        String strDate = "2008-10-19 10:11:30.345" ;
        // 准备第一个模板，从字符串中提取出日期数字
        String pat1 = "yyyy-MM-dd HH:mm:ss.SSS" ;
        // 准备第二个模板，将提取后的日期数字变为指定的格式
        String pat2 = "yyyy年MM月dd日 HH时mm分ss秒SSS毫秒" ;
        SimpleDateFormat sdf1 = new SimpleDateFormat(pat1) ;          // 实例化模板对象
        SimpleDateFormat sdf2 = new SimpleDateFormat(pat2) ;          // 实例化模板对象
        Date d = null ;
        try{
            d = sdf1.parse(strDate) ;    // 将给定的字符串中的日期提取出来
        }catch(Exception e){             // 如果提供的字符串格式有错误，则进行异常处理
            e.printStackTrace() ;        // 打印异常信息
        }
        System.out.println(sdf2.format(d)) ;     // 将日期变为新的格式

    }

}
```

代码(9) DateFormatterTest.java


```java
import java.time.LocalDate;

public class DateFormatterTest {

    public static void main(String[] args) {
        //将字符串转化为时间
        String dateStr= "2016年10月25日";
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy年MM月dd日");
        LocalDate date= LocalDate.parse(dateStr, formatter);
        System.out.println(date.getYear() + "-" + date.getMonthValue() + "-" + date.getDayOfMonth());

        System.out.println("==========================");

        //将日期转换为字符串输出
        LocalDateTime now = LocalDateTime.now();
        DateTimeFormatter format = DateTimeFormatter.ofPattern("yyyy年MM月dd日 hh:mm:ss");
        String nowStr = now.format(format);
        System.out.println(nowStr);


    }

}
```

谢 谢！