

如何使用Java中的Date和Calendar类 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/267/2021_2022__E5_A6_82_E4_BD_95_E4_BD_BF_E7_c104_267114.htm Java 语言

的Calendar(日历),Date(日期), 和DateFormat(日期格式)组成了Java标准的一个基本但是非常重要的部分. 日期是商业逻辑计算一个关键的部分. 所有的开发者都应该能够计算未来的日期, 定制日期的显示格式, 并将文本数据解析成日期对象. 我们写了两篇文章, 这是第一篇, 我们将大概的学习日期, 日期格式, 日期的解析和日期的计算. 我们将讨论下面的类: 具体类(和抽象类相对)Java.util.Date 抽象类java.text.DateFormat 和它的一个具体子类,Java.text.SimpleDateFormat 抽象类java.util.Calendar 和它的一个具体子类,Java.util.GregorianCalendar 具体类可以被实例化, 但是抽象类却不能. 你首先必须实现抽象类的一个具体子类. Date 类从Java 开发包(JDK) 1.0 就开始进化, 当时它只包含了几个取得或者设置一个日期数据的各个部分的方法, 比如说月, 日, 和年. 这些方法现在遭到了批评并且已经被转移到了Calendar类里去了, 我们将在本文中进一步讨论它. 这种改进旨在更好的处理日期数据的国际化格式. 就象在JDK 1.1中一样, Date 类实际上只是一个包裹类, 它包含的是一个长整型数据, 表示的是从GMT(格林尼治标准时间)1970年, 1月1日00:00:00这一刻之前或者是之后经历的毫秒数. 创建一个日期对象 让我们看一个使用系统的当前日期和时间创建一个日期对象并返回一个长整数的简单例子. 这个时间通常被称为Java 虚拟机(JVM)主机环境的系统时间. 以下是引用片段:

```
import java.util.Date. public class DateExample1 { public static void
```

```
main(String[] args) { // Get the system date/time Date date = new
Date(). System.out.println(date.getTime()). } }
```

在星期六, 2001年9月29日, 下午大约是6:50的样子, 上面的例子在系统输出设备上显示的结果是 1001803809710. 在这个例子中, 值得注意的是我们使用了Date构造函数创建一个日期对象, 这个构造函数没有接受任何参数. 而这个构造函数在内部使用

了System.currentTimeMillis()方法来从系统获取日期. 那么, 现在我们已经知道了如何获取从1970年1月1日开始经历的毫秒数了. 我们如何才能以一种用户明白的格式来显示这个日期呢? 在这里类java.text.SimpleDateFormat和它的抽象基类

Java.text.DateFormat就派得上用场了. 日期数据的定制格式假如我们希望定制日期数据的格式, 比方星期六-9月-29日-2001年. 下面的例子展示了如何完成这个工作: 以下是引用片段:

```
import Java.text.SimpleDateFormat. import Java.util.Date. public
class DateExample2 { public static void main(String[] args) {
SimpleDateFormat bartDateFormat = new
SimpleDateFormat("EEEE-MMMM-dd-yyyy"). Date date = new
Date(). System.out.println(bartDateFormat.format(date)). } }
```

只要通过向SimpleDateFormat的构造函数传递格式字符串"EEE-MMMM-dd-yyyy", 我们就能够指明自己想要的格式. 你应该可以看见, 格式字符串中的ASCII字符告诉格式化函数下面显示日期数据的哪一个部分. EEEE是星期, MMMM是月, dd是日, yyyy是年. 字符的个数决定了日期是如何格式化的. 传递"EE-MM-dd-yy"会显示 Sat-09-29-01. 请察看Sun公司的Web站点获取日期格式化选项的完整的指示. 100Test 下载频道开通, 各类考试题目直接下载。详细请访问 www.100test.com