

java认证:JSR310新Java日期_时间APIJava认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/584/2021_2022_java_E8_AE_A4_E8_AF_81_c104_584693.htm 对 Java 6 日期/时间 API 的改进 JSR 310 日期/时间 API 试图通过提供更好的性能和易用性改进 Java 的当前日期/时间 API。例如，Java Calendar 类将日期同时存储为与标准纪元之间的偏移量(以毫秒为单位)以及一组日历字段(例如，星期几、几号以及月份)。此双精度表示导致在意外的时间重新计算日历字段，从而产生不可预测的性能特点。与此相比，JSR 310 类仅将日期/时间表示存储为与 Date 和 Calendar 所使用的同一标准纪元之间的偏移量(以毫秒为单位)。仅当需要时才会计算日期等日历字段，并且不会使用这些日历字段进行内部日期表示。JSR 310 还对当前 Java 日期/时间模型进行了改进。Java 6 API 不包含表示本地时间(不具有关联时区的时间)、持续时间或时间间隔的类。这迫使程序员使用令人困惑的设计做法，例如使用 int 表示持续时间。JSR 310 包含表示上述各个概念的类，从而可以进行更为明确的程序设计。最后，JSR 310 API 通过使用不可改变的类努力实现线程安全。Java 当前的日期/时间类 Date 和 Calendar 都是可改变的，因而都不是线程安全的。JSR 310 日期/时间概念 JSR 310 API 利用了从多个第三方 Java 日期/时间 API 中获得的经验。JSR 310 主要基于 Joda Time API。其他影响因素包括 ICU、Time and Money 和 Calendar Date。JSR 310 的 API 是围绕 Joda Time 中使用的相同 5 个基本日期/时间概念构建的：离散的时间线 瞬间 不完全时间 持续时间 时段 时间间隔 离散的时间线 像 Joda Time 一样，JSR 310 使用离散化的时间线：时

间被建模为由小的固定持续时间分隔的连续的瞬间序列。JSR 310 的离散时间线具有纳秒分辨率，因此可以表示时间“2008年1月1日午夜后1纳秒”，但不能表示时间“2008年1月1日午夜后1皮秒”。该时间线上的每个离散纳秒都被视为一个瞬间，如下所示。瞬间是离散化时间线上的特定点。一个瞬间示例是“世界标准时间2008年1月7日23:00:00.0”。同样可以将瞬间定义为与标准纪元之间的偏移量(以纳秒为单位)，例如“世界标准时间1970年1月1日之后20,000,000纳秒”。这两个说明都定义了离散时间线上的单个唯一点。瞬间不同于不完全时间，后者定义了时间线上的一组时刻，而不是一个唯一的时刻。JSR 310 API 提供了多个表示瞬间的类：Instant、OffsetDateTime 和 ZonedDateTime，所有这些类都实现了 ReadableInstant 接口。OffsetDateTime 类表示日期、每日时间以及与世界标准时间 (coordinated universal time, UTC) 之间的偏移量(如 1:00)。类似的 ZonedDateTime 类包含时区 ID(如 America/New_York)而不是偏移量。给定的时区可能使用多个不同的偏移量，具体取决于一年中的时间。例如，America/New_York 时区的偏移量在夏令时为 -4:00，在其他时间为 -5:00。因此，当必须考虑特定于区域设置的时间规则(如夏令时)时，应该使用 ZonedDateTime 类。ZonedDateTime 类提供了几种类别的用于创建、访问和修改瞬间的方法。要创建 ZonedDateTime 的新实例以表示计算机的默认时区中的当前系统时间，可以使用 Clock.currentZonedDateTime() 工厂方法，如下面的示例所示。Clock systemClock = Clock.system(); ZonedDateTime currentTime = systemClock.currentZonedDateTime(). 要创建一个

ZonedDateTime 实例以表示特定的、预先确定的日期，可以使用多个 ZonedDateTime.dateTime 工厂方法之一。下面的示例演示了如何创建一个 ZonedDateTime 以表示计算机的默认时区中的 2000 年 1 月 1 日午夜。Clock systemClock = Clock.system(). TimeZone defaultTimeZone = systemClock.timeZone(). int year = 2000. int month = 1. int day = 1. int hour = 0. int minute = 0. ZonedDateTime theDate = ZonedDateTime.dateTime(year, month, day, hour, minute, defaultTimeZone). OffsetDateTime 实例的创建方式与 ZonedDateTime 实例类似，区别在于传入 OffsetDateTime.dateTime 工厂方法的是 ZoneOffset 而不是 TimeZone。要获得 ZoneOffset 的实例，可以使用静态 ZoneOffset.zoneOffset(int hours) 方法，其中 hours 是与 UTC 之间的偏移量(以小时为单位)。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