1. 日期時間API的疊代：

第一代：jdk 1.0 Date類

第二代：jdk 1.1 calender類，一定程度上替換Date類

第三代：jdk 1.8 提出了新的一套API

1. 前兩代存在的問題舉例：

可变性：像日期和时间这样的类应该是不可变的。

偏移性：Date中的年份是从1900开始的，而月份都从0开始。

格式化：格式化只对Date有用，Calendar则不行。

此外，它们也不是线程安全的；不能处理闰秒等。

1. java 8 中新的日期時間API涉及到的包



1. 本地日期、本地時間、本地日期時間的使用：LocalDate / LocalTime / LocalDateTime
   1. 說明：
2. 分别表示使用ISO-8601日历系统的日期、时间、日期和时间。它们提供了简单的本地日期或时间，并不包含当前的时间信息，也不包含与时区相关的信息。
3. *LocalDateTime相较于LocalDate、LocalTime，使用频率要高*
4. *类似于Calendar*
   1. 常用方法：



1. 時間點：Instant
   1. 說明：
2. 时间线上的一个瞬时点。概念上讲，它只是简单的表示自1970年1月1日0时0分0秒（UTC）开始的秒数。
3. 類似於 java.util.Date類
   1. 常用方法：

 

1. 日期時間隔式化類：DateTimeFormatter
   1. 說明：
2. *格式化或解析日期、时间*
3. *类似于SimpleDateFormat*
   1. 常用方法：
4. 實例化方法

预定义的标准格式。如：**ISO\_LOCAL\_DATE\_TIME;ISO\_LOCAL\_DATE;ISO\_LOCAL\_TIME**

本地化相关的格式。如：ofLocalizedDateTime(FormatStyle.LONG)

自定义的格式。如：ofPattern(“yyyy-MM-dd hh:mm:ss”)

1. 常用方法：



特別的：自定義的格式。如：ofPattern(“yyyy-MM-dd hh:mm:ss”)

*重点： 方式：自定义的格式。如：ofPattern(“yyyy-MM-dd hh:mm:ss”)* DateTimeFormatter formatter3 = DateTimeFormatter.*ofPattern*("yyyy-MM-dd hh:mm:ss");  
 *//格式化* String str4 = formatter3.format(LocalDateTime.*now*());  
 System.*out*.println(str4);*//2019-02-18 03:52:09  
  
 //解析* TemporalAccessor accessor = formatter3.parse("2019-02-18 03:52:09");  
 System.*out*.println(accessor);