

Java 8 新特性（三）新的日期时间类



乐百川 (/u/7753478e1554) [+ 关注](#)

2017.09.14 19:14 字数 1468 阅读 71 评论 3 喜欢 4

(/u/7753478e1554)

老日期类

一说到在Java中处理日期和时间，我们首先想到的肯定是 `java.util.Date` 类。确实，由于很多教材和文献都比较老，所以大多数人学习Java的时候使用的都是这个类。而且这个类确实在整个Java生态中担任过重要的作用。

但是随着时间推移，这个类已经完全不能满足需要。首先，现在各种数据库和编程语言都将日期和时间区分开，它们的存储方式也不相同。但是 `java.util.Date` 类对此却没有做明确区分，这就导致有时候我们不得不手动区分。下面是Hibernate官方文档的一个例子，由于数据库需要明确日期类型来进行存储，所以在Java代码中必须显式指定 `Temporal` 来区分类型。

```
@Entity(name = "DateEvent")
public static class DateEvent {

    @Id
    @GeneratedValue
    private Long id;

    @Column(name = "`timestamp`")
    @Temporal(TemporalType.DATE)
    private Date timestamp;

    //Getters and setters are omitted for brevity
}
```

同样的，旧日期类对于时区等特性也没有很好的支持。这些问题都在新日期时间类中得到了解决。

新日期时间类

Java 8引入了三个新的日期时间类，分别是 `LocalDate`、`LocalTime` 和 `LocalDateTime`，分别处理日期、时间和日期时间。而且这些类的使用方法也是大同小异，基本上很快就能学会。

创建实例

要构造这些新类的实例也很简单，有两种方法，第一种是使用 `now()` 方法，会创建当前时间的实例。第二种是使用 `of` 方法，传入要构造的参数。

```
//日期
LocalDate date = LocalDate.now();
System.out.println(date);
//时间
LocalTime time = LocalTime.now();
System.out.println(time);
//日期时间
LocalDateTime dateTime = LocalDateTime.now();
System.out.println(dateTime);
LocalDate birthday = LocalDate.of(1994, Month.JANUARY, 18);
```

日期运算



新类附带了很多新的日期运算方法，可以帮助我们进行很多计算。 `plusXXX` 和 `minusXXX` 方法可以帮助我们计算一段时间之前/之后的日期和时间。 `withXXX` 方法需要 `TemporalAdjuster` 类型的对象，这个方法可以帮助我们确定今年的第几天这样的问题。 `TemporalAdjusters` 类包含了很多现成的实例，可供我们使用。当然如果寿星的话，我们还可以自己实现 `TemporalAdjuster` 接口。这里的例子使用的是 `LocalDate`，不过这些方法对于其他几个类也同样适用。

```
//日期运算
LocalDate date = LocalDate.of(1993, 12, 25);
//十天以后
LocalDate dateAfter10Days = date.plusDays(10);
System.out.println("十天以后:" + dateAfter10Days);
//一个月前
LocalDate dateBefore1Month = date.minusMonths(1);
System.out.println("一个月前:" + dateBefore1Month);
//这个月的最后一天
LocalDate lastDayOfMonth = date.with(TemporalAdjusters.lastDayOfMonth());
System.out.println("这个月最后一天:" + lastDayOfMonth);
//今年第一天
System.out.println("今年第一天:" + date.with(TemporalAdjusters.firstDayOfYear()));
```

查询

有时候需要对日期和时间进行比较复杂的查询，这时候可以利用 `TemporalQuery<R>` 接口，它只有唯一的一个查询方法 `R queryFrom(TemporalAccessor temporal)`。下面的例子利用这个接口和lambda表达式实现了一个查询到年底还有几天的查询对象，然后调用 `query` 方法执行这个查询并返回结果。

```
TemporalQuery<Integer> remainDaysOfYear = temporal -> {
    LocalDate d = LocalDate.from(temporal);
    LocalDate lastDayOfYear = d.with(TemporalAdjusters.lastDayOfYear());
    Period period = d.until(lastDayOfYear);
    return period.getDays();
};
int days = date.query(remainDaysOfYear);
System.out.println("到年底还剩几天:" + days);
```

日期和字符串转换

这里使用 `LocalDateTime` 作为例子，其他类的转换方法也是类似的。用到的转换类只有一个，那就是 `DateTimeFormatter`。

首先先由日期类转换为字符串，我们想到的最简单的办法就是使用 `toString()` 方法。注意到结果中日期和时间中使用 `T` 进行分隔，这是Java新标准的要求。

```
LocalDateTime datetime = LocalDateTime.now();
//默认toString方法
System.out.println(datetime);
// 2017-09-14T18:12:00.943
```

如果希望自行控制日期和时间的格式，就需要使用 `DateTimeFormatter` 了。最简单的用法就是用 `ofPattern` 方法将格式传入，这样就会返回一个使用该格式的格式化器。然后用这个格式化器来进行格式化。

```
//手动格式化
DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
System.out.println(
    datetime.format(formatter));
// 2017-09-14 18:12:00
```

最后就是字符串到日期类的解析了。同样需要的是格式化器，不过这次要调用格式化器的 `parse` 方法传入字符串序列，如果格式匹配的话，就会返回相应的日期对象，否则会抛出异常。



```
datetime = LocalDateTime.from(formatter.parse("1993-11-25 12:00:15"));
System.out.println(datetime);
```

Instant类以及新老日期转换

Instant类

旧的 `java.util.Date` 有一个接受 `long` 类型的构造函数，这个 `long` 类型的意义是当前日期距 1970-01-01 00:00:00 的毫秒数。新的Java 8 中专门有一个类对这个参数进行了抽象，这就是 `Instant` 类，而且把精确度提高到纳秒级别。

新老日期转换

`Instant` 类一般不单独使用，比较常见的用法就是在新老日期键进行转换。Java 8为旧日期类型全部添加了 `toInstant()` 方法，可以将日期转换为 `Instant` 实例，然后将 `Instant` 实例转换为新日期类型。反过来也是一样的。

```
//旧日期转换为新日期
Date date = new Date();
LocalDateTime dateTime = LocalDateTime.ofInstant(date.toInstant(),
    ZoneId.of("Asia/Shanghai"));
System.out.println(dateTime);
//新日期转换为旧日期
Date now = Date.from(Instant.now());
System.out.println(now);
```

JDBC支持

现在新的JDBC驱动基本上都已经支持插入和读取新日期类了。不过具体到底那个版本支持还需要查阅其文档，基本上主流的最新的MySQL等数据库肯定都支持了。

原来需要使用 `java.sql.Date` 来添加日期，由于它继承自 `java.util.Date`，也算一个过时类。如果要插入和读取新的日期类对象，需要使用 `setObject` 和 `getObject` 方法。

这里用到了H2内存数据库。运行这个例子可以看到，现在新的JDBC驱动都可以直接将新日期对象存入数据库，并且可以直接取出。


```
public static void main(String[] args) throws SQLException {
    //使用H2嵌入式内存数据库
    String url = "jdbc:h2:mem:test";
    try (Connection connection = DriverManager.getConnection(url)) {
        //创建表
        Statement statement = connection.createStatement();
        statement.execute("CREATE TABLE person(id INT PRIMARY KEY,name VARCHAR(255),birthday
        //插入数据
        PreparedStatement p = connection.prepareStatement("INSERT INTO person VALUES(?,?)");
        p.setInt(1, 1);
        p.setString(2, "zhang3");
        p.setObject(3, LocalDate.of(1993, 5, 25));
        p.executeUpdate();
        //查询数据
        ResultSet rs = statement.executeQuery("SELECT *FROM person WHERE id=1");
        rs.next();
        Person person = new Person();
        person.setId(rs.getInt(1));
        person.setName(rs.getString(2));
        person.setBirthday(rs.getObject(3, LocalDate.class));
        System.out.println("数据库中:" + person);
    }
}
```

这里的 `Person` 是我编写的一个小实体类，内部就这三个字段，还有IDE自动生成的一堆样板代码，就不放这个类了。



以上就是Java 8 对于新的日期时间类的内容。这些类可以帮助我们轻松处理时间有关问题。如果不是处理老旧项目的话，我们最好都是用这些新类，享受它们带给我们的便利。

Java学习笔记 (/nb/5018579) 举报文章 © 著作权归作者所有



乐百川 (/u/7753478e1554)

写了 227799 字，被 1345 人关注，获得了 772 个喜欢 (/u/7753478e1554)

+ 关注

苦逼码农预备队大队二队队长

如果觉得我的文章对您有用，请随意赞赏。您的支持将鼓励我继续创作！

赞赏支持

喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button) | 4




更多分享

(http://cwb.assets.jianshu.io/notes/images/1676505)



登录后发表评论 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-comment-form)

3条评论 只看作者 按喜欢排序 按时间正序 按时间倒序



AndyJeaon (/u/4935913f0a7e)

2楼 · 2017.09.15 10:14 (/u/4935913f0a7e)

处理时间我一般都是用Calendar


赞 回复


- 乐百川 (/u/7753478e1554) : calendar我觉得太难用，自从用了java8以后再也回不去了


2017.09.15 15:51 回复
- AndyJeaon (/u/4935913f0a7e) : @乐百川 (/users/7753478e1554) 好的，学习了


2017.09.15 17:16 回复
- 添加新评论

被以下专题收入，发现更多相似内容

- 

野生技术宅协会 (/c/a4c0587558db?utm_source=desktop&utm_medium=notes-included-collection)
- 

程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)
- 


计算机大杂烩 (/c/0bd12eecbb8?utm_source=desktop&utm_medium=notes-included-collection)
- 

Java 杂谈 (/c/0b39448c4e08?utm_source=desktop&utm_medium=notes-

^

🔗

included-collection)

 java (/c/7cc1796735be?utm_source=desktop&utm_medium=notes-included-collection)

推荐阅读

更多精彩内容 > (/)

Spring Roo 简介 (/p/ee22d3ef3130?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

(/p/ee22d3ef3130?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

一直以来，Java/Spring开发被认为是笨重的代表，无法快速生成项目原型和骨架。所以，Spring推出了Spring Roo这个项目，帮助我们快速生成项目原型。...

乐百川 (/u/7753478e1554?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

设计模式（一）工厂方法模式 (/p/358d05fc5dc8?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

(/p/358d05fc5dc8?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

介绍 关于设计模式，大家可能已经有很多了解了。因此这里直接开始吧。首先来介绍一下工厂方法模式，它是一种创建型模式。在设计模式：可复用面向对象软件的基础中对工厂方法由如下的解释。工厂方法模式...

乐百川 (/u/7753478e1554?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

爸妈是很搞笑的人，包括离婚也是 (/p/bafca6c5aad4?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

(/p/bafca6c5aad4?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

我爸妈离婚，又复婚了。有些事情过去了，但不代表从来没有发生过。有些时候我们安之若素，维持着生活的波澜不惊，可心里那个受伤小孩的哭声从来没有...

烟袋斜姐 (/u/e8b5fe2926a7?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

如何把一件事讲得言简意赅，语出惊人，少即是多 (/p/09f4c707e95c?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

(/p/09f4c707e95c?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

01 你知道吗？一年时间，你会说出一千万句话。但是为什么？你却很难拥有“讲清楚一件事”这项技能。你经常发现，对一件事描述了半天。别人却满脸懵逼...

道长是名思维贩子 (/u/92eb338437ee?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

妹妹说没有拿到一等奖学金，姐姐这样回应 (/p/60598e585016?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

(/p/60598e585016?utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

姐姐，我没有拿到一等奖学金 晚上，我在看书的时候，听到手机滴滴声，是妹妹发来的信息。这次她没有拿到一等奖学金。我心里咯噔了一下，比起一等...

高学长 (/u/777fd60d9dd0?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

 登录/注册

为你个性化推荐内容

(/sign_in?utm_source=desktop&utm_medium=note&utm_campaign=maleskine&utm_content=note&utm_medium=pc_all_hots&utm_source=recommendation)

 下载简书App

随时随地发现和创作内容

(/apps/bind?utm_source=desktop&utm_medium=click-note-bottom-bind)

