原 JDK 1.8中Instant时间戳类,Duration类，Period类,等一些与事件相关的类和接口

2018年01月28日 17:14:34 鱼刺_64 阅读数 6588

Instant时间戳类

（一）含义：从1970-01-01 00：00：00到当前时间的毫秒值

（二）常用的方法:

now():

获取当前的时间，获取的是当前的美国时间，和处于东八区的我们相差八个小时。

```
Instant ins=Instant.now();
System.out.println(ins);
```

atOffset():
设置偏移量例如：

```
OffsetDateTime time=ins.atOffset(ZoneOffset.ofHours(8));
System.out.println(time);
```

atZone()
获取系统默认时区时间，参数为一个时区的编号，可以通过时区编号类获取出来

还可以通过ZoneId.systemDefault()来获取本地的默认时区ID

```
ZonedDateTime zoneDateTime=ins.atZone(ZoneId.systemDefault());
System.out.println(zonedDateTime);
```

get系列的方法
getEpochSecond()：获取从1970-01-01 00：00：00到当前时间的秒值

getNano():把获取到的当前时间的描述换算成纳秒

ofEpochSecond()方法：在计算机元年（1970-01-01 00：00：00）的基础上增加秒数

```
Instant instant=Onstant.ofEpochSecond(5);
System.out.println(instant)
```

Duration类

用法：用于计算两个时间间隔的类

常用方法：
between():计算两个时间的间隔，默认的单位是秒

例：Duration between =Duration.between(start,end);

toMillis()方法：将秒转换成毫秒

System.out.println(between.toMillis());

Period 类

常用方法：between():计算两个时间之间的间隔

```
LocalDate s=LocalDate.of(1985,03,05);
LocalDate now=LocalDate,now();
Periodbe=Period.between(s,now);
System.out.println(be.getYears());
System.out.println(be.getMonth());
System.out.println(be.getDays())
```

TemporalAdjuster: 时间校正器（接口）

一般我们使用该接口的一个实现类TemporalAdjusters中的一些常量来指定日期。

（1）使用TemporalAdjusters自带的常量来设置日期

```
LocalDate now=LocalDate.now();
System.out.println(now);
LocalDate with =now.with(TemporalAdjusters.lastDayOdYear());
System.out.println(with);
```

(2)采用TemporalAdjusters中的next方法来指定日期

```
  LocalDate date = now.with(TemporalAdjusters.next(DayOfWeek.SUNDAY));
        System.out.println(date);
```

(3)采用自定义的方法来指定日期

```
 LocalDateTime ldt = LocalDateTime.now();
        LocalDateTime workDay = ldt.with(new TemporalAdjuster() {
public Temporal adjustInto(Temporal temporal) {

                LocalDateTime ld = (LocalDateTime) temporal;

                DayOfWeek dayOfWeek = ld.getDayOfWeek();
                if (dayOfWeek.equals(DayOfWeek.FRIDAY)) {
                    return ld.plusDays(3);
                } else if (dayOfWeek.equals(DayOfWeek.SATURDAY)) {
                    return ld.plusDays(2);
                } else {

                    return ld.plusDays(1);
                }
```

DateTimerFormatter

解析和格式化日期或时间

常用方法：

ofPattern("yyyy-MM-dd");:获取对象的静态方法

```
DateTimeFormatter dateFormat = DateTimeFormatter.ofPattern("yyyy-MM-dd");
```

```
LocalDateTime now = LocalDateTime.now();
```

format():把一个日期对象的默认格式格式化成指定的格式

```
String format1 = dateFormat.format(now);

System.out.println(format1);
```

还可以使用日期类中的format 方法传入一个日期格式化类对象。

now1.format(DateTimeFormatter.ISO_LOCAL_DATE_TIME);

ZoneID世界时区类

常用方法

getAvailableZoneIds():获取世界各个地方的时区的集合

```
Set<String> availableZoneIds = ZoneId.getAvailableZoneIds();
```

ZoneId.systemDefault()：获取系统默认时区的ID

```
ZoneIdzoneId = ZoneId.systemDefault(); //中国为Asia/Shanghai
```