LocalDate and LocalTime

LocalDate represents a date without a time zone, such as 1-1-2000. LocalTime represents time without a time zone, such as 04:44:59.12 - unlike Instant which is an offset from the Java epoch and as such can be calculated into a precise point of time these two are just date or time without any relation to the epoch - a human readable date and time. There are several ways to obtain LocalTime and LocalDate instances, here are few:

```
LocalDate localDate = LocalDate.now();

localDate = LocalDate.ofYearDay(2005, 86); // The 86th day of 2005 (27-Mar-2005)

localDate = LocalDate.of(2013, Month.AUGUST, 10); //10th of Aug 2013


LocalTime localTime = LocalTime.of(22, 33); //10:33 PM

localTime = LocalTime.now();

localTime = LocalTime.ofSecondOfDay(4503); // The 4,503 second in a day (1:15:30 AM)
```

LocalDate and Local time follow the same general concept of multithreading as Instant does - and as such their instances are immutable. LocalDate and LocalTime have calculation and comparison methods similar to the ones Instant has (some of the methods are defined by the java.time.temporal.Temporal interface which implemented by all of these classes):

```
LocalDate localDate1 = localDate.plus(5, ChronoUnit.HOURS);

localDate.isBefore(localDate1);
```

LocalDateTime

The last important player in the simple date and time classes is LocalDateTime - this is a combination of LocalDate and LocalTime representing a date and the time within that date, again no time zone. LocalDateTime seems to be very similar to Instant, a reminder: "an Instant is point in time without time zone” and one could say that a point in time is nothing more than a date and time within that date. But there is a difference: LocalDateTime is not a point on the time line as Instant is, LocalDateTime is just a date and time as a person would write on a note. Consider the following example: two persons which were born at 11am, July the 2nd 2013. The first was born in the UK while the second in California. If we ask any of them for their birth date it will look that they were born on the same time (this is the LocalDateTime) but if we align the dates on the timeline (using Instant) we will find out that the one born in California is few hours younger than the one born in the UK (NB: to create the appropriate Instant we have to convert the time to UTC, this is where the difference lays). Beside of that LocalDateTime behaves very similar to the other classes illustrated above:

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

// Jump to 25 hours and 3 minutes into the future

LocalDateTime inTheFuture = localDateTime.plusHours(25).plusMinutes(3);

// We could do the same on localTime or localDate

System.out.println(localDateTime.toLocalTime().plusHours(25).plusMinutes(3));
System.out.println(localDateTime.toLocalDate().plusMonths(2));



// We could also use TemportalAmount (in this case a Duration and
Period)

System.out.println(localDateTime.toLocalTime().plus(Duration.ofHours(25
).plusMinutes(3)));

System.out.println(localDateTime.toLocalDate().plus(Period.ofMonths(2))
);
```