Dates and Times

Periods, Durations, Instants

```
// Period can be used only with LocalDate and LocalDateTime

Period p1 = Period.ofYears(2);

  => P2Y

Period p2 = Period.ofMonths(3);

  => P3M

Period p3 = Period.ofWeeks(1);

  => P7D

Period p4 = Period.ofDays(11);

  => P11D

Period p5 = Period.of(2, 0, 15);

  => P2Y15D
```

```
// periods are used with plus/minus on date


Period period = Period.of(1, 2, 5);

LocalDate date = LocalDate.of(2022, 11, 20);

date = date.plus(period);

  => 2023-01-25

date = date.minus(period);

  => 2022-11-20
```

```
// Duration can be used only with LocalTime and LocalDateTime
Duration d1 = Duration.ofDays(3);
  => PT72H
Duration d2 = Duration.ofHours(2);
  => PT2H
Duration d3 = Duration.ofMinutes(45);
  => PT45M
Duration d4 = Duration.ofSeconds(10);
  => PT10S
Duration d5 = Duration.ofMillis(240);
  => PT0.24S
Duration d6 = Duration.ofNanos(2503);
  => PT0.000002503S
```

```
// using chronounits

import java.time.*;

import java.time.temporal.*;


Duration d0 = Duration.of(3, ChronoUnit.HALF_DAYS);          // PT36H

Duration d1 = Duration.of(3, ChronoUnit.DAYS);               // PT72H

Duration d2 = Duration.of(2, ChronoUnit.HOURS);              // PT2H

Duration d3 = Duration.of(45, ChronoUnit.MINUTES);           // PT45M

Duration d4 = Duration.of(10, ChronoUnit.SECONDS);           // PT10S

Duration d5 = Duration.of(240, ChronoUnit.MILLIS);           // PT0.24S

Duration d6 = Duration.of(2503, ChronoUnit.NANNOS);          // PT0.000002503S
```

```
// ChronoUnits can also be used to determine how far apart are two times:


LocalTime t1 = LocalTime.of(17, 30);

LocalTime t2 = LocalTime.of(20, 45);

System.out.println(ChronoUnit.HOURS.between(t1, t2));

  => 3

System.out.println(ChronoUnit.MINUTES.between(t1, t2));

  => 195
```

```
// durations are used with plus/minus on time


LocalTime time = LocalTime.of(17, 30);

Duration d3 = Duration.ofMinutes(45);

time = time.plus(d3);

  => 18:15

time = time.minus(d3)

  => 17:30
```

```
// instants are used to record time-stamps in the application

Instant now = Instant.now();

System.out.println(now);

  => 2023-04-18T09:20:52.904935284Z


// example: measure duration of the process:

Instant before = Instant.now();

  // ... some time-consuming process

Instant after = Instant.now();

Duration dur = Duration.between(before, after);

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

  => 255 (milliseconds)
```

Under the hood:

`Instant` uses a `long` representing epoch-seconds which are measured from the standard Java epoch of of `1970-01-01T00:00:00Z`

```
// converting ZonedTimeDate to instant:


ZoneId zone = ZoneId.of("Europe/Zagreb");

ZonedDateTime z = ZonedDateTime.of(2022, 11, 2, 21, 50, 14, 145, zone);

Instant inst = z.toInstant();

System.out.println(inst);

  => 2022-11-02T20:50:14.000000145Z
```