

DateTime

Instant

- Instant reprezentuje bod v čase (jako dříve `java.util.Date`), ale s drobným rozdílem – `java.util.Date` obsahuje počet milisekund od 1.1.1970, zatímco `java.time.Instant` obsahuje počet nanosekund.

```
Instant instant = Instant.now();
```

- Instant nepracuje s lidskými jednotkami času (roky, měsíce, dny), pokud chcete s nimi pracovat, musíte zkonvertovat Instant na jinou třídu jako například `LocalDateTime` (viz. dále).

Práce s datem od Java SE 8 I.

- Aktuální datum:

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

- Datum zítra:

```
LocalDate tomorrow = today.plusDays(1);
```

- Aktuální rok:

```
int currentYear = LocalDate.now().getYear();
```

- Datum v textové podobě ve formátu den.měsíc.rok:

```
String stringDate =
```

```
    LocalDate.now().format(DateTimeFormatter.ofPattern("dd.MM.yyyy"));
```

- Převod z data v textové podobě do objektu typu `java.time.LocalDate`:

```
LocalDate date = LocalDate.parse("01.06.2014",
```

```
    DateTimeFormatter.ofPattern("dd.MM.yyyy"));
```

Práce s datem od Java SE 8 II.

- Také existují třídy `LocalTime` a `LocalDateTime`.
- Také je možné provádět konverzi mezi `java.util.Date` a novými třídami.
- Další příklady:
 - <http://www.oracle.com/technetwork/articles/java/jf14-date-time-2125367.html>
 - <http://docs.oracle.com/javase/tutorial/datetime/overview/index.html>
 - <http://blog.progs.be/542/date-to-java-time>
 - <https://www.topjavablogs.com/news/how-to-transform-java-s-date-and-datetime-to-unix-time>
 - <https://jdbc.postgresql.org/documentation/head/8-date-time.html>
 - <http://www.thoughts-on-java.org/hibernate-5-date-and-time/>

Důležité metody

- Získání hodnoty (den, hodina, minuta, ...):
 - Dříve: `cal.get(Calendar.DAY_OF_MONTH);`
 - Nyní: `localDateTime.getDayOfMonth();`
- Nastavení hodnoty (den, hodina, minuta, ...):
 - Dříve: `cal.set(Calendar.DAY_OF_MONTH, 1);`
 - Nyní: `localDateTime.withDayOfMonth(1);`
 - Vrací objekt, na kterém byla operace provedena, tudíž je možné:
`localDateTime.withDayOfMonth(1).withHour(1);`
 - Obdobně metody `plus` (dříve `add`) a `minus` (dříve nelogicky `add` s mínusovou hodnotou)

DateTimeFormatterBuilder

- DateTimeFormatter slouží pro konverzi datumu na String a obráceně. Pro jeho konstrukci je také možné použít DateTimeFormatterBuilder:

```
DateTimeFormatterBuilder builder = new DateTimeFormatterBuilder();

DateTimeFormatter formatter = builder

    .optionalStart()

    .appendValue(ChronoField.HOUR_OF_DAY)

    .appendLiteral(":")

    .appendValue(ChronoField.MINUTE_OF_HOUR)

    .appendLiteral(" ")

    .optionalEnd()

    .appendValue(ChronoField.DAY_OF_MONTH)

    .appendLiteral(".")

    .appendValue(ChronoField.MONTH_OF_YEAR)

    .appendLiteral(".")

    .appendValue(ChronoField.YEAR).toFormatter();

LocalDateTime localDateTime = LocalDateTime.now();

System.out.println(localDateTime.format(formatter));

System.out.println(LocalDate.parse("12:00 1.1.2000", formatter));
```