

PROGRAMOWANIE OBIEKTOWE JAVA – LABORATORIUM

LOCALDATE

LOCALDATE

Obiekty klasy `LocalDate` przechowują datę bez informacji o strefie czasowej. Jest to przyjęty międzynarodowy format ISO - yyyy-MM-dd, np. 2019-05-25.

TWORZENIE OBIEKTÓW

Ponieważ `LocalDate` przechowuje tylko datę w określonym formacie, tworzenie obiektów jest o wiele prostsze:

- `LocalDate.now()` - aktualna data
- `LocalDate.of(int year, int month, int day)` - data podawana jako liczby rok-miesiąc-dzień
- `LocalDate.parse(String date)` - data podawana jako format ISO ("2019-05-25")

MODYFIKACJA DAT

Podobnie do klasy `Calendar` ze starego API dat, korzystając `LocalDate` można również dodawać lub odejmować konkretną liczbę jednostek czasowych.

Metoda	Znaczenie
<code>LocalDate plusDays(long daysToAdd)</code>	Dodaje do daty podaną liczbę dni i zwraca nową zmienioną
<code>LocalDate plusWeeks(long weeksToAdd)</code>	Dodaje do daty podaną liczbę tygodni i zwraca nową zmienioną
<code>LocalDate plusMonths(long monthsToAdd)</code>	Dodaje do daty podaną liczbę miesięcy i zwraca nową zmienioną
<code>LocalDate plusYears(long yearsToAdd)</code>	Dodaje do daty podaną liczbę lat i zwraca nową zmienioną
<code>LocalDate minusDays(long daysToSubtract)</code>	Odejmuje od daty podaną liczbę dni i zwraca nową zmienioną
<code>LocalDate minusWeeks(long weeksToSubtract)</code>	Odejmuje od daty podaną liczbę tygodni i zwraca nową zmienioną
<code>LocalDate minusMonths(long monthsToSubtract)</code>	Odejmuje od daty podaną liczbę miesięcy i zwraca nową zmienioną
<code>LocalDate minusYears(long yearsToSubtract)</code>	Odejmuje od daty podaną liczbę lat i zwraca nową zmienioną

CHRONOUNIT

Na potrzeby reprezentacji różnych jednostek czasowych powstała klasa `ChronoUnit` ze stałymi reprezentującymi konkretne jednostki (np. `ChronoUnit.YEARS`). Przy ich pomocy można skorzystać z metod klasy `LocalDate`:

- `LocalDate plus(long amountToAdd, TemporalUnit unit)` - dodaj konkretną liczbę danych jednostek czasowych
- `LocalDate minus(long amountToSubtract, TemporalUnit unit)` - dodaj konkretną liczbę danych jednostek czasowych

PERIOD

To nie wszystko! Powstała również specjalna klasa reprezentująca okres - odstęp między dwiema datami wyrażony różnicą w latach, dniach, tygodniach i miesiącach.

Dodatkowo klasa `LocalDate` została wzbogacona o metody umożliwiające modyfikację daty korzystając z okresów:

- `public LocalDate plus(TemporalAmount amountToAdd)` - dodaj do daty okres
- `public LocalDate minus(TemporalAmount amountToAdd)` - odejmij od daty okres gdzie jako `TemporalAmount` można podać obiekt klasy `Period`

LOCALTIME

Analogicznie do klasy `LocalDate` powstała klasa `LocalTime` czyli sam czas (godzina, minuta, sekunda...) bez informacji o strefie czasowej.

LOCALDATETIME

Ostatnim typem "lokalnym" jest `LocalDateTime`, czyli połączenie dwóch poprzednich - `LocalDate` oraz `LocalTime`. Oznacza to, że klasa `LocalDateTime` przechowuje datę oraz czas bez informacji o strefie czasowej!. Posiada wszystkie metody z tych klas (opisane powyżej).

ZONEDATETIME

Ostatnim już typem reprezentującym datę i czas jest `ZonedDateTime` czy data + czas wraz z informacją o strefie czasowej.

MODYFIKACJE - ZMIANA STREFY CZASOWEJ

Klasa `ZonedDateTime` posiada wszystkie omawiane metody klasy `LocalDateTime`. Skupimy się zatem tylko na najważniejszej różnicy, tj. modyfikacji strefy czasowej. Omówimy to na przykładzie lotu samolotem.

Wyobraź sobie, że wylatujesz z Warszawy 22.05.2019 o godzinie 15:00, lecisz do Tokio przez 7 godzin. Która godzina i jaka data będzie na miejscu (w obecnej strefie czasowej czyli w Tokyo)?

`ZonedDateTime` można zamienić na inną strefę czasową w tym samym momencie można wykorzystać metodę `ZonedDateTime withZoneSameInstant(ZoneId zoneId)`.

FORMATOWANIE DAT

Java 8 wprowadziła klasę `DateTimeFormatter`, która jest odpowiednikiem klasy `SimpleDateFormat`. Różnica polega na tym, że nowy "formater" potrafi współpracować z nowymi typami czasowymi (`ZonedDateTime`, `LocalDateTime`). Przy jego pomocy można formatować daty czyli przedstawiać je jako tekst w określonym formacie lub parsować je z tekstu na obiekty.