


# Bài 40: Lớp LocalDate

---

- ✓ Giới thiệu
- ✓ Các phương thức
- ✓ Ví dụ minh họa



# Lớp LocalDate

- 
- Là một lớp final dùng để tạo các đối tượng đại diện cho thông tin về ngày tháng năm trong Java
  - Lớp này không chứa múi giờ nên không áp dụng với các định dạng về múi giờ
  - Lớp LocalDate nằm trong gói `java.time`

# Một số phương thức

➤ Sau đây là các phương thức thường dùng và mô tả:

Phương thức	Mô tả
<code>adjustInto(Temporal tem)</code>	Thiết lập một đối tượng thời gian với cùng thời gian như trong tham số.
<code>atTime(int hour, int minute)</code>	Kết hợp ngày hiện thời với thời gian giờ phút để tạo một đối tượng <code>LocalDateTime</code>
<code>atTime(int hour, int minute, int second)</code>	Kết hợp ngày hiện thời với thời gian giờ phút giây để tạo một đối tượng <code>LocalDateTime</code>
<code>atTime(int hour, int minute, int second, int nanoSec)</code>	Kết hợp ngày hiện thời với thời gian giờ phút giây và nano giây để tạo một đối tượng <code>LocalDateTime</code>
<code>atTime(LocalTime time)</code>	Kết hợp ngày hiện thời với thời gian cho trong tham số để tạo một đối tượng <code>LocalDateTime</code>
<code>compareTo(ChronoLocalDate other)</code>	So sánh ngày hiện thời với một ngày nào đó
<code>format(DateTimeFormatter formatter)</code>	Định dạng ngày hiện thời với formatter cho trước
<code>from(TemporalAccessor tem)</code>	Tạo một đối tượng <code>LocalDate</code> từ tham số nhận được

# Một số phương thức

➤ Sau đây là các phương thức thường dùng và mô tả:

<code>getDayOfMonth()</code>	Lấy ngày của tháng trong đối tượng hiện thời
<code>getDayOfWeek()</code>	Lấy ngày của tuần trong đối tượng hiện thời
<code>getDayOfYear()</code>	Lấy ngày của năm trong đối tượng hiện thời
<code>getEra()</code>	Lấy kỉ nguyên của đối tượng hiện thời
<code>getMonth()</code>	Lấy tháng của năm, trả về enum Month
<code>getMonthValue()</code>	Lấy giá trị số của tháng của đối tượng hiện thời, từ 1-12
<code>getYear()</code>	Lấy năm của đối tượng hiện thời
<code>isAfter(ChronoLocalDate other)</code>	Kiểm tra xem ngày hiện thời có muộn hơn ngày trong tham số hay không
<code>isBefore(ChronoLocalDate other)</code>	Kiểm tra xem ngày hiện thời có sớm hơn ngày trong tham số hay không
<code>isLeapYear()</code>	Kiểm tra xem năm hiện tại có phải năm nhuận không
<code>minusMonths(long n)</code>	Trả về bản copy của đối tượng hiện thời giảm thời gian đi n tháng
<code>minusDays(long n)</code>	Trả về bản copy của đối tượng hiện thời giảm thời gian đi n ngày

# Một số phương thức

➤ Sau đây là các phương thức thường dùng và mô tả:

<code>minusWeeks(long n)</code>	Trả về bản copy của đối tượng hiện thời giảm thời gian đi n tuần
<code>minusYears(long n)</code>	Trả về bản copy của đối tượng hiện thời giảm thời gian đi n năm
<code>now()</code>	Trả về đối tượng <code>LocalDate</code> đại diện cho ngày hiện thời theo thời gian hệ thống
<code>of(int year, int month, int day)</code>	Tạo một thể hiện của <code>LocalDate</code> từ năm, tháng, ngày cho trước
<code>of(int year, Month month, int day)</code>	Tạo một thể hiện của <code>LocalDate</code> từ năm, tháng, ngày cho trước
<code>of(int year, int dayOfYear)</code>	Tạo một thể hiện của <code>LocalDate</code> từ năm và ngày của năm
<code>parse(CharSequence text)</code>	Tạo đối tượng <code>LocalDate</code> từ một chuỗi với định dạng cho trước
<code>parse(CharSequence text, DateTimeFormatter formatter)</code>	Tạo đối tượng <code>LocalDate</code> từ một chuỗi với định dạng trong <code>formatter</code>
<code>plusYears(long n)</code>	Trả về bản sao của đối tượng hiện thời với số năm được thêm vào
<code>plusMonths(long n)</code>	Trả về bản sao của đối tượng hiện thời với số tháng được thêm vào
<code>plusWeeks(long n)</code>	Trả về bản sao của đối tượng hiện thời với số tuần được thêm vào
<code>plusDays(long n)</code>	Trả về bản sao của đối tượng hiện thời với số ngày được thêm vào

# Ví dụ

- Ví dụ sau tạo đối tượng của LocalDate từ hai cách khác nhau và thao tác trên đối tượng vừa tạo:

```
LocalDate today = LocalDate.now(); // ngày hiện thời
// ngày 25/10/2025
LocalDate date1 = LocalDate.of(2025, 10, 25);
System.out.println("Thông tin ngày hiện thời: " + today);
System.out.println("Thông tin ngày trong date1: " + date1);
// thay đổi vài năm thông tin ngày hiện thời
var hundredYearsLatter = today.plusYears(100);
System.out.println("100 năm sau: " + hundredYearsLatter);
```

```
Thông tin ngày hiện thời: 2020-09-25
Thông tin ngày trong date1: 2025-10-25
100 năm sau: 2120-09-25
```



# Minh họa trong công cụ lập trình

- Tạo đối tượng `LocalDate` và thực hiện một số thao tác trên nó.

# Nội dung tiếp theo

**Lớp LocalTime**