Lec 50 JAVA 8 P6:

Java's Date and Time API (introduced in Java 8) provides easy tools to handle dates and times in software projects. With classes like LocalDate, LocalTime, and LocalDateTime, you can get the current date/time, do calculations, and format or manipulate them for real-world needs.

Importance of Date and Time in Projects

Dates and times are essential in most applications, such as tracking purchases, returns, and payments in e-commerce. Java provides built-in support for handling these needs.

Introduction to Java 8 Date and Time API

Java 8 introduced a new Date and Time API, which includes several classes and interfaces dedicated to working with dates and times more effectively than older Java versions.

Key Classes: LocalDate, LocalTime, LocalDateTime

The three main classes are LocalDate (for dates), LocalTime (for times), and LocalDateTime (for both).

These are simple to use and don't require prior knowledge of earlier date/time handling in Java.

Getting the Current Date

The LocalDate.now() method retrieves the current system date. This date is based on the computer's system clock, not from the internet or real-time sources.

Manipulating Dates with LocalDate

The LocalDate object offers many methods to add or subtract days, months, years, or weeks. You can also convert dates to strings for further manipulation or display.

Converting Dates to String and Manipulation

The toString() method converts date objects to string format, making it easier to display or further process them. This is a common Java method available in all classes.

Using LocalDate Methods for Calculations

You can check if a year is a leap year, find the date after a certain number of years or weeks, and chain multiple operations (like adding years and weeks together).

Formatting Dates with DateTimeFormatter

DateTimeFormatter allows you to customize how dates are displayed (e.g., day-month-year). This is easier and more reliable than manually manipulating strings.

Real-World Uses: Storing Dates in Projects

Dates are often stored in objects, such as a student's date of birth. LocalDate can be used to ensure the correct format and validity (e.g., making sure the date is in the past).

Extracting Date Components

Methods like getDayOfMonth, getDayOfWeek, getMonth, and getYear allow you to extract specific parts of a date for reporting or logic.

Working with LocalTime

LocalTime.now() gets the current time, including hours, minutes, seconds, and even nanoseconds. You can format, split, or manipulate these values as needed.

Manipulating Time Values

You can add or subtract hours, minutes, and seconds using methods like plusHours or minusMinutes. Formatting can be done using patterns in DateTimeFormatter.

Using the 'of' Method for Custom Dates/Times

The of method lets you create date or time objects for any specific value, not just the current moment. This is useful for calculations like checking leap years or setting shop opening times.

Combining Date and Time: LocalDateTime

LocalDateTime combines both date and time, allowing you to add or subtract years, months, days, hours, and minutes in one object. This is handy for tasks like calculating expiry dates or reminders.

Practical Project Applications

Date and time APIs are often used in projects for things like validating input (e.g., date of birth must be in the past), setting expiry dates, and scheduling reminders. Reliable date/time handling is crucial for correct business logic.