

Nội dung báo cáo JEP 8368874

Tên và số JEP:

JEP 8368874 – Add a JDatePicker UI Component to the Swing UI Toolkit (Preview)

Chủ sở hữu: Tejesh R

Trạng thái: Dự thảo (Draft / Preview)

Phạm vi: Java SE – Client Libraries (javax.swing)

JEP 8368874 bổ sung thành phần JDatePicker cho Swing, giúp chọn ngày hoặc dải ngày qua giao diện lịch popup, hỗ trợ `LocalDate` và kiến trúc MVC, thể hiện nỗ lực hiện đại hóa Swing của OpenJDK.

Mục tiêu / Motivation:

JEP 8368874 được đề xuất nhằm giải quyết một trong những thiếu sót lớn nhất của Swing: thiếu một thành phần chọn ngày (Date Picker) tiêu chuẩn. Trong nhiều năm, các lập trình viên Java buộc phải tự xây dựng tính năng này từ các thành phần cơ bản như `JTextField`, `JComboBox` hoặc `JSpinner`, hoặc phụ thuộc vào các thư viện bên thứ ba như `JCalendar`, `LGoodDatePicker` hay `SwingX (JXDatePicker)`. Tuy nhiên, những giải pháp đó thường gây ra nhiều vấn đề về giao diện, khả năng tương thích, và không hỗ trợ chuẩn thời gian mới `java.time`. Ngoài ra, việc sử dụng thư viện ngoài khiến ứng dụng trở nên phức tạp, nặng nề và khó bảo trì. Vì vậy, mục tiêu chính của JEP 8368874 là cung cấp một Date Picker chính thức, được tích hợp sẵn trong JDK, đảm bảo tính đồng nhất về giao diện, hỗ trợ đầy đủ API thời gian hiện đại, và giảm thiểu sự phụ thuộc vào các giải pháp không chính thức. Bên cạnh đó, tính năng này còn hướng đến việc cải thiện trải nghiệm người dùng và tăng năng suất cho lập trình viên khi phát triển các ứng dụng giao diện trên nền Swing.

Chi tiết kỹ thuật:

JDatePicker bao gồm ba thành phần chính: một ô nhập liệu (`TextField`) hiển thị ngày đã chọn hoặc cho phép nhập trực tiếp, một nút bật lịch (popup button) và một bảng lịch (calendar panel) hiển thị tháng hiện tại. Khi người dùng nhấn vào biểu tượng lịch, một popup hiển thị sẽ xuất hiện, cho phép lựa chọn ngày, tháng và năm bằng thao tác chuột. Thành phần này hỗ trợ hai chế độ chính: chọn một ngày duy nhất (Single Selection) và chọn dải ngày (Range Selection). Trong chế độ đầu tiên, người dùng có thể chọn một ngày duy nhất, giá trị đó sẽ được hiển thị trong ô nhập; còn ở chế độ thứ hai, người dùng có thể chọn hai mốc thời gian, và toàn bộ dải ngày nằm giữa sẽ được đánh dấu rõ ràng trên lịch. Tất cả

dữ liệu được xử lý thông qua `LocalDate` hoặc `SortedSet<LocalDate>`, tùy vào chế độ sử dụng.

Ngoài hai chế độ chọn, `JDatePicker` còn cho phép thiết lập ngày mặc định bằng phương thức `setDate(LocalDate.of(...))`, đặt giới hạn phạm vi năm có thể cuộn (ví dụ ± 200 năm so với năm hiện tại), và thay đổi định dạng ngày hiển thị thông qua `DateTimeFormatter`. Thành phần này cũng hỗ trợ `ChangeListener` – một cơ chế giúp ứng dụng có thể phản hồi tự động khi người dùng thay đổi ngày hoặc dải ngày. Giao diện lịch popup được quốc tế hóa (i18n) để tự động thích ứng với ngôn ngữ và định dạng ngày tháng của hệ thống. Về mặt cấu trúc, `JDatePicker` được xây dựng theo kiến trúc MVC của Swing, có thể nhúng trực tiếp vào container hoặc hoạt động độc lập dưới dạng popup. Việc lựa chọn tháng và năm được thực hiện qua các panel riêng biệt, đồng thời người dùng có thể điều hướng tháng/năm bằng các nút điều hướng hoặc cuộn chuột. JEP cũng quy định rõ việc triển khai mô hình `DateSelectionModel` để quản lý sự kiện và cho phép mở rộng tùy chỉnh giao diện thông qua lớp `AbstractCalendarPanel`.

Ảnh hưởng:

Sự ra đời của JEP 8368874 mang lại ảnh hưởng tích cực cả về mặt kỹ thuật lẫn trải nghiệm người dùng. Đối với lập trình viên, việc có sẵn một `DatePicker` chuẩn trong Swing giúp giảm đáng kể công sức và thời gian phát triển, đồng thời loại bỏ sự phụ thuộc vào các thư viện bên thứ ba. API của `JDatePicker` thân thiện, dễ hiểu, và đồng bộ với phong cách thiết kế của các thành phần Swing khác. Việc hỗ trợ trực tiếp `LocalDate` giúp các ứng dụng Java hiện đại hóa luồng xử lý thời gian, dễ dàng tích hợp với các tính năng mới của nền tảng Java như `java.time`, `DateTimeFormatter` hay các thư viện phân tích dữ liệu. Với người dùng cuối, `JDatePicker` mang lại trải nghiệm chọn ngày trực quan, chính xác và thân thiện hơn so với các ô nhập liệu truyền thống. Giao diện lịch popup giúp hạn chế lỗi nhập sai định dạng, đồng thời tăng tính chuyên nghiệp và dễ sử dụng của ứng dụng.

Về tổng thể, JEP 8368874 không chỉ khắc phục một lỗ hổng kéo dài nhiều năm của Swing mà còn đóng vai trò như một minh chứng cho nỗ lực hiện đại hóa các thư viện giao diện của Java. Tính năng này góp phần củng cố vị thế của Swing như một công cụ phát triển giao diện bền vững, phù hợp với thời đại mới, đồng thời giúp hệ sinh thái Java trở nên nhất quán và dễ tiếp cận hơn đối với lập trình viên trong kỷ nguyên Java 25.