

## Java Observable và Observer

Bất kỳ lớp Java nào quan tâm đến việc có thể quan sát được sẽ mở rộng lớp Có thể **quan sát**. Sau đó, lớp mở rộng có thể được quan sát bởi các lớp quan tâm khác trong các phần khác của chương trình. Các lớp đã đăng ký được thông báo ngay khi có bất kỳ thay đổi nào được kích hoạt trong lớp có thể quan sát được mà lớp quan sát được đăng ký. Đây là mối quan hệ giữa các lớp Observable và Observer. Các lớp Observer không là gì khác ngoài việc triển khai giao diện **Observer**. Các **Observer** giao diện cung cấp một *bản cập nhật()* phương pháp đó được gọi khi một người quan sát được thông báo về một sự thay đổi trong đối tượng quan sát được.

Một lớp muốn được quan sát hoặc lớp mở rộng *Observable* phải gọi phương thức *setChanged()* nếu nó đã thay đổi và phải kích hoạt thông báo bằng cách gọi phương thức *notifyObservers()*. Điều này lần lượt gọi phương thức *update()* của lớp quan sát viên. Tuy nhiên, nếu một đối tượng gọi phương thức *informObserver()* mà không gọi phương thức *setChanged()* thì sẽ không có hành động nào diễn ra. Do đó, một đối tượng có thể quan sát phải gọi phương thức *setChange()* trước khi gọi phương thức *notifyObservers()* khi có bất kỳ thay đổi nào. Thứ tự này sau đó sẽ gọi phương thức *update()* của các lớp quan sát một cách liên mạch.

## JavaFX Observer

Giao diện JavaFX *Observable* là giao diện cơ sở cho nhiều lớp vùng chứa và giao diện trong JavaFX Collection Framework. Ví dụ, *ObservableList<E>* và *ObservableArray<T>* là hai cách triển khai phổ biến của giao diện *Observable* cơ sở. API Bộ sưu tập lỗi của Java đã chứa nhiều lớp vùng chứa hữu ích để đại diện cho cấu trúc dữ liệu chung của danh sách, tập hợp và bản đồ. Ví dụ: *java.util.ArrayList<E>* là một triển khai mảng khá lớn của giao diện *java.util.List<E>* để chứa một danh sách các đối tượng. Tuy nhiên, chúng không có khả năng hoạt động liên tục khi cần có chức năng đồng bộ giữa mô hình danh sách và thành phần chế độ xem trong kịch bản GUI.

Trước JavaFX, các nhà phát triển Swing đã dựa vào *ArrayList* để chứa danh sách các đối tượng và sau đó hiển thị chúng trong một điều khiển giao diện người dùng dạng danh sách, chẳng hạn như *JList*. Tuy nhiên, *ArrayList* quá chung chung và không được xây dựng lưu ý đến yêu cầu đồng bộ hóa khi liên kết với một thành phần khung nhìn. Do đó, việc cập nhật, thêm hoặc loại bỏ các đối tượng khỏi danh sách mô hình là khá khó khăn và đồng thời phản ánh những thay đổi trong thành phần khung

nhìn. Để khắc phục sự cố này, JavaFX sử dụng các giao diện có thể quan sát được và việc triển khai chúng, chẳng hạn như *ObservableList* *<E>* .

Vì *ObservableList* *<E>* tuân thủ các quy tắc của mô hình có thể quan sát và quan sát được của MVC, chẳng hạn như cung cấp thông báo cho người quan sát quan tâm của nó về bất kỳ cập nhật, bổ sung hoặc loại bỏ các đối tượng khỏi danh sách mô hình, nó đã trở thành một vùng chứa thực tế để sử dụng bất kỳ danh sách trong đấu trường JavaFX. Ở đây, biểu diễn dữ liệu trong mô hình và chế độ xem được đồng bộ hóa liền mạch. JavaFX *ObservableList* *<E>* thường được sử dụng trong các điều khiển giao diện người dùng như *ListView* và *TableView* . Hãy xem qua một ví dụ nhanh để xem cách *ObservableList* *<E>* thực sự được sử dụng.