**1**

- SerialVersionUID là một thuộc tính định danh dùng trong serialize/deserialize một object của một class implement [Serializable interface.](https://shareprogramming.net/marker-interface-serializable-trong-java/)

Serializable là một marker interface không chứa bất kỳ một abstract method nên một class implement Serializable không cần triển khai bất kỳ method nào. Việc implement Serializable chỉ để thông báo rằng class này có thể sử dụng trong việc tuần tự hoá (ví dụ như đọc ghi file, chuyển qua network).

private static final long ***serialVersionUID*** = 1L;

- serialVersionUID được sử dụng bởi các object serialization API của java. Nếu một object được **deserialized**(**đọc dữ liệu vào object)** và **serialized(ghi object)**giống serialVersionUID thì quá trình **deserialized**sẽ cố gắng thực hiện thành công. Ngược lại quá trình **deserialized**có thể không tương ứng.

**2.**

Phương thức invokeLater() sẽ chạy ứng dụng của chúng ta trong một luồng do EventQueue quản lý. Thực ra bạn cũng không cần đến EventQueue làm gì, chỉ cần tạo một đối tượng BasicEx rồi setVisible(true) là có thể chạy được nhưng trên tài liệu của Oracle thì lại khuyên chúng ta đặt bên trong EventQueue vì lý do là làm như vậy sẽ đảm bảo an toàn cho ứng dụng (mình cũng không hiểu tại sao) nên thôi thì mình cứ làm vậy

**3**

Lớp **Student** cũng implement interface **Serializable** để đối tượng có thể được lưu trữ và truyền qua các kết nối mạng hoặc tệp tin.

Ngoài ra, lớp **Student** sử dụng annotation **@XmlRootElement** để chỉ định rằng đối tượng **Student** có thể được chuyển đổi thành định dạng XML, và sử dụng annotation **@XmlAccessorType(XmlAccessType.FIELD)** để chỉ định rằng các trường dữ liệu của đối tượng sẽ được sử dụng trong quá trình chuyển đổi này.