**Tính đóng gói (Encapsulation) trong Java**

**1. Khái niệm tính đóng gói**

Trong bài trước, tôi có đưa ra một số ví dụ từ 1 lớp chúng ta có thể truy cập trực tiếp đến thuộc tính của đối tượng trong một lớp thông qua tên của thuộc tính đó. Tuy nhiên, chúng ta nên hạn chế dùng cách này vì nó không đảm bảo một trong những nguyên tắc quan trọng nhất của lập trình hướng đối tượng đó là tính bảo mật và che giấu thông tin. Vì vậy, để khắc phục, trong lập trình hướng đối tượng có 1 tính chất có thể cho phép chúng ta truy cập và thay đổi giá trị thuộc tính thông qua các phương thức của lớp đó là tính đóng gói (Encapsulation).

Vậy đóng gói là gì? Đóng gói là sự che giấu bên trong dữ liệu riêng của mỗi đối tượng của lớp được khai báo và chỉ được truy xuất thông qua hệ thống các phương thức có sẵn của lớp (chỉ có thể gọi những phương thức có sẵn của lớp). Vì vậy, nó còn được gọi là **data hiding** (nghĩa là che giấu dữ liệu).

Tính đóng gói có những đặc điểm như sau:

* Tạo ra cơ chế để ngăn ngừa việc gọi phương thức của lớp này tác động hay truy xuất dữ liệu của đối tượng thuộc về lớp khác.
* Dữ liệu riêng (khi được khai báo là private) của mỗi đối tượng được bảo vệ khỏi sự truy xuất không hợp lệ từ bên ngoài.
* Người lập trình có thể dựa vào cơ chế này để ngăn ngừa việc gán giá trị không hợp lệ vào thành phần dữ liệu của mỗi đối tượng.
* Cho phép thay đổi cấu trúc bên trong của một lớp mà không làm ảnh hưởng đến những lớp bên ngoài có sử dụng lớp đó.

Để cài đặt tính đóng gói, chúng ta có 2 bước như sau:

* Khai báo các thuộc tính của đối tượng trong lớp là private để các lớp khác không thể truy cập trực tiếp/sửa đổi được.
* Cung cấp các phương thức getter/setter có phạm vi truy cập là public để truy cập và sửa đổi các giá trị của thuộc tính trong lớp. Phương thức getter là phương thức truy cập vào thuộc tính của đối tượng và trả về các thuộc tính của đối tượng, còn phương thức setter là phương thức truy cập vào thuộc tính của đối tượng và gán giá trị cho các thuộc tính của đối tượng đó.

**2. Ví dụ về tính đóng gói**

Ví dụ sau sẽ minh họa tính đóng gói trong Java:

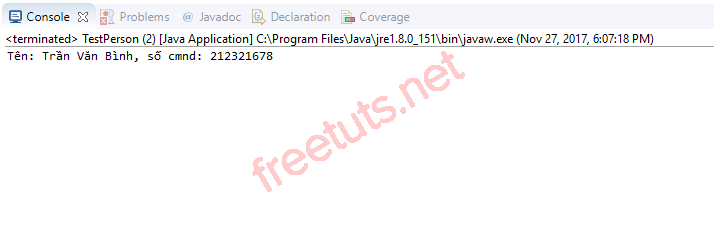
**Person.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | package vidu;    public class Person {      // khai báo các thuộc tính của đối tượng là private      private String cmnd;      private String hoTen;        // tạo các phương thức getter/setter      // 2 phương thức getCmnd() và getHoTen() là phương thức getter      // dùng để trả về số chứng minh nhân dân và họ tên của đối tượng      // và kiểu trả về của hai phương thức này tương ứng với kiểu dữ liệu của thuộc tính      // 2 phương thức setCmnd() và setHoTen() là phương thức setter      // dùng để gán giá trị cho thuộc tính chứng minh nhân dân và họ tên của đối tượng      // trong đó tham số truyền vào của 2 phương thức này được gọi là tham số (biến cục bộ)      // và có kiểu dữ liệu tương ứng với kiểu dữ liệu của thuộc tính (biến đối tượng)      public String getCmnd() {          return cmnd;      }        // this là từ khóa có ý nghĩa là một tham chiếu đặc biệt      // chiếu tới đối tượng chủ của phương thức hiện hành      // this có thể được dùng để truy cập biến đối tượng (instance variable)      // hoặc gọi phương thức đối với đối tượng hiện hành.      // Thông thường, công dụng này của this chỉ có ích      // khi tên biến đối tượng bị trùng với tham số (biến cục bộ - local variable) của phương thức      public void setCmnd(String cmnd) {          this.cmnd = cmnd;      }      public String getHoTen() {          return hoTen;      }      public void setHoTen(String hoTen) {          this.hoTen = hoTen;      }    } |

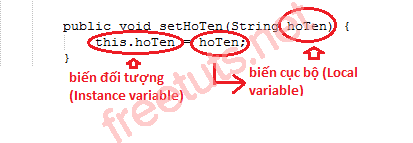
**TestPerson.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | package vidu;    public class TestPerson {        public static void main(String[] args) {          Person person = new Person();            // gán giá trị họ tên cho đối tượng person vừa tạo thông qua setHoTen()          // và gán số chứng minh nhân dân thông qua setCmnd()          person.setHoTen("Trần Văn Bình");          person.setCmnd("212321678");            // truy cập đến tên của đối tượng person thông qua phương thức getHoten()          // và số chứng minh nhân dân thông qua phương thức getCmnd()          System.out.println("Tên: " + person.getHoTen() + ", số cmnd: " + person.getCmnd());      }    } |

Kết quả sau khi biên dịch chương trình:



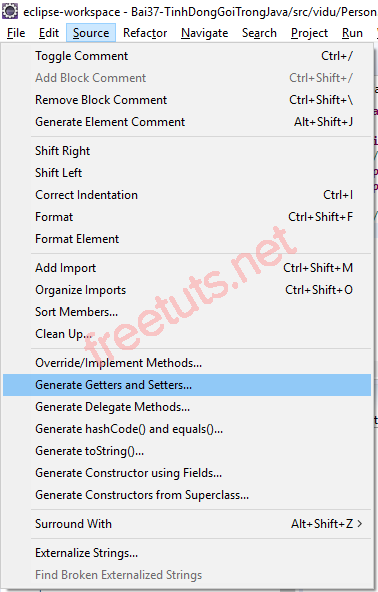
Trong lớp Person, chúng ta có đoạn code như sau:



this là từ khóa có ý nghĩa là một tham chiếu đặc biệt chiếu tới đối tượng chủ của phương thức hiện hành. Trong lớp Person, phương thức setHoTen() lấy một tham số (hay còn gọi là biến cục bộ - Local Variable) hoTen kiểu String trùng tên với thuộc tính hoTen của lớp đó, nếu chúng ta chỉ viết "hoTen" mà không có từ khóa this đằng trước thì trình biên dịch sẽ hiểu là ta đang nói đến tham số hoTen. Vì vậy, để gọi đến thuộc tính (hay còn gọi là biến đối tượng - Instance Variable), cách duy nhất là sử dụng tham chiếu this để gọi một cách tường minh.

**Lưu ý:** Để tạo nhanh phương thức getter/setter của các thuộc tính trong lớp, chúng ta sẽ làm như sau:

Bước 1: Vào Source -> Generate Getters and Setters:



Bước 2: Hộp thoại Generate Getters and Setters xuất hiện. Chúng ta có thể chọn Select All để chọn tất cả các đối tượng cần tạo getter/setter hoặc có thể chọn riêng từng thuộc tính. Ngoài ra, nếu các bạn chỉ muốn tạo phương thức getter (hoặc setter) thì các bạn có thể chọn Select Getters (hoặc Select Setters). Sau đó bấm OK để kết thúc.

