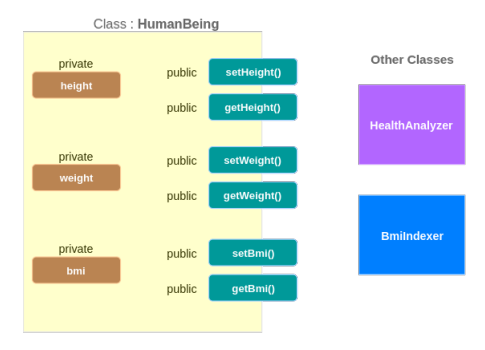
**Tính đóng gói (Encapsulation)**

Đóng gói trong Java – Ẩn các biến của một lớp khỏi các lớp khác và chỉ cấp quyền truy cập vào chúng thông qua các phương thức (setters và getters). Do đó, Đóng gói trong ngôn ngữ Java có nghĩa là ràng buộc dữ liệu (biến) với mã (phương thức – setters và getters). Trong hướng dẫn này, chúng ta sẽ tìm hiểu trực giác và thực hiện đóng gói trong ngôn ngữ Java với các chương trình ví dụ.



Các điểm được thực hiện từ sơ đồ trên là

* Các biến (trong ví dụ: chiều cao, cân nặng, bmi) được khai báo là riêng tư và do đó không hiển thị đối với các lớp khác.
* Đối với mỗi biến, có một phương thức setter và phương thức getter, đặt giá trị cho biến và nhận giá trị của biến tương ứng. Ví dụ: Đối với biến height , phương thức setter là setHeight(), phương thức getter là getHeight() .
* Các phương thức Setter và Getter là công khai, do đó các lớp khác có thể nhìn thấy.

|  |
| --- |
| package encapsulation;  class Human{  private float weight;  private float height;  public float getWeight() {  return weight;  }  public void setWeight(float weight) {  this.weight = weight;  }  public float getHeight() {  return height;  }  public void setHeight(float height) {  this.height = height;  }  }  public class Encapsulation {  public static void main(String[] args) {  Human h1 = new Human();  // using setters of Human  h1.setHeight(1.65f);  h1.setWeight(68);  System.out.println("Person has "+h1.getWeight()+" kgs and is "+h1.getHeight()+" meters in h");  }  } |

this : truy cập đến tham số ẩn bên trong lớp của bạn.

Lợi ích của đóng gói trong java

* Tất cả các trường (field) của lớp có có chế độ chỉ đọc (read-only) hoặc chỉ ghi (write-only), tức là chỉ có hàm getter hoặc setter.
* Một lớp có thể có toàn bộ điều khiển thông qua những gì được lưu giữ trong các trường (field) của nó.
* Người sử dụng của class không biết cách các class lưu trữ dữ liệu. Một class có thể thay đổi kiểu dữ liệu của một trường và người dùng class không cần sự thay đổi trong code.

|  |
| --- |
| //Account class - private data members bundled with getters and setters  **class** Account {      //private data members  **private** **long** acc\_no;  **private** String name,email;  **private** **float** amount;      //public getter and setter methods for each data member  **public** **long** getAcc\_no() {  **return** acc\_no;      }  **public** **void** setAcc\_no(**long** acc\_no) {  **this**.acc\_no = acc\_no;      }  **public** String getName() {  **return** name;      }  **public** **void** setName(String name) {  **this**.name = name;      }  **public** String getEmail() {  **return** email;      }  **public** **void** setEmail(String email) {  **this**.email = email;      }  **public** **float** getAmount() {  **return** amount;      }  **public** **void** setAmount(**float** amount) {  **this**.amount = amount;      }    }  **public** **class** Main {  **public** **static** **void** main(String[] args) {          //create instance of Account class          Account myAcc=**new** Account();          //set values for data members through setter methods          myAcc.setAcc\_no(775492842L);          myAcc.setName("SoftwareTestingHelp.com");          myAcc.setEmail("sth\_account@sth.com");          myAcc.setAmount(25000f);          //read data member values through getter methods          System.out.println("Account No:" + myAcc.getAcc\_no()+" "+"Account Name:"                  + myAcc.getName()+" \n"+"Account holder email:" + myAcc.getEmail()+"\n "                  + "Amount in Account:" + myAcc.getAmount());      }  } |