**4 tính chất đặc trưng của lập trình hướng đối tượng OOP**

**Tính đóng gói:**

Che dấu đi những tính chất của đối tượng, những đối tượng bên ngoài không thể tác đông trực tiếp vào những tính chất của đối tượng, mà phải thông qua các phương thức do nó quy định. Set/get

Không cho biết luồng xử lý bên trong của đối tượng.

Tính chất bảo đảm sự toàn vẹn, bảo mật của đối tượng.

**Tính kế thừa: Inheritance**

Là tính chất cho phép tạo ra một lớp mới mà dựa trên từ một lớp có sẵn. Lớp mới tọa ra là lớp con, lớp có sẵn là lớp cha. Và đương nhiên nếu kế thừa lại thì lớp con có khả năng dùng lại, mở rộng các đặc tính mà lớp cha có (như method, thuộc tính).

**Tính đa hình: Polymorphism**

Khi một tác vụ được thực hiện theo nhiều các khác nhau -> gọi là đa hình. Thông qua các phương thức của đối tượng, các phương thức hoàn toàn có thể giống nhau, nhưng việc xử lý logic thì lại khác nhau. Trong java có 2 phương thức thể hiện tính đa hình một cách rõ rang:

* Nạp chồng phương thức: Còn biết đến là Overloading. Khi một lớp có nhưng phương thức cùng tên nhưng khác nhau về số lượng, kiểu tham số truyền vào thì sẽ khác nhau.
* Khi gọi phương thức -> dựa trên số lượng tham số và kiểu thì sẽ gọi các phương thức khác nhau.
* Ví dụ như việc tạo các Constructor của class.
* Ghi đè phương thức: Overriding

Là phương thức cùng tên, cùng tham số, cùng một kiểu trả về, nhưng class viết theo logic khác với logic của lớp cha. Và phương thức xuất hiện ở cả lớp cha và con.

Trong khi thực thi, nếu class con không có phương thức riêng, thì phương thức của lớp cha được gọi. Và ngược lại.

**Tính trừu tượng: Abstraction**

Là một tiến trình ẩn các chi tiết trình triển khai và chỉ hiển thị tính năng tới người dùng. Tính trừu tượng cho phép loại bỏ các tính chất phức tạp của đối tượng, bằng cách chỉ đưa ra các thuộc tính và phương pháp cần thiết của đối tượng trong lập trình.

Tình trừu tượng cho phép tập trung vào những tính chất cần thiết của đối tượng thay vì quan tâm đến cách thức thực hiện nó.

Sử dụng Abstract class và abstract interface để thể hiện tính trừu tượng.

**Ví dụ Demo:**

Mô tả ví dụ:

**B1: Xây dựng một lớp**

Animal:

|  |  |
| --- | --- |
|  | Chứa một thuộc tính: name lưu tên của Animal  Phương thức sayHello -> những class kế thừa lại Animal -> nên để kiểu abstract |

**Mục tiêu:** Do có phương thức abstract -> class abstract -> class thể hiện dù là con vật gì thì đều phải có phương thức sayHello này.

Tính trừu tượng được thể hiện ở điều này.

**B2: Class kế thừa**

Tạo các class Cat, Dog kế thừa từ class Animal này -> khi đó override lại method sayHello đề kêu theo các các riêng của từng con vật.

|  |  |
| --- | --- |
|  | Class Cat kế thừa lại từ class Animals -> kế thừa lại thuộc tính name, và phương thức sayHello.  Class Cat override sayHello theo cách riêng của nó |
|  | Tương tự như lớp Cat -> lớp Dog kế thừa và override lại thộc tính và method sayHello |

Tính đóng gói thể hiện ở phạm vi truy cập của thuộc tính name.

Tính kế thừa ở việc lớp Cat và Dog kế thừa lại class Animal

**B3: Tạo một class Zoo để quản lý các con vật được tạo ra.**

Zoo có các phương thức thêm, xóa các con vật. Sử dụng phương thức showListAnimal để gọi phương thức sayHello. Tính đa hình thế hiện do một phương thức sayHello nhưng tùy từng con vật sẽ có phương thức sayHello khác nhau.

|  |  |
| --- | --- |
|  | Class Zoo dùng để quản lý các loài vật như ở Cat và Dog.  Lưu danh sách thành ArrayList  Method: add để thêm mới một loài vật  Method: remove để xóa một loài vật  Tình đa hình thể hiện trong phương thức showListAnimals  Khi gọi cùng một phương thức showListAnimals thì tùy từng con vật mà có những cách khác nhau. |

Source đính kèm:



Tài liệu tham khảo:

|  |  |
| --- | --- |
| Tính chất cơ bản của OOP | https://bit.ly/32ssbyR |