

Java Object - Đối tượng trong Java

Object Class

Mặc định lớp Object là lớp cha của tất cả các lớp trong java. Nói cách khác nó là một lớp cáo nhất trong java.

[java-object.png](#) image object.png type unknown

Sử dụng lớp Object là hữu ích nếu bạn muốn tham chiếu bất kỳ đối tượng nào mà bạn chưa biết kiểu dữ liệu của đối tượng đó. Chú ý rằng biến tham chiếu của lớp cha có thể tham chiếu đến đối tượng của lớp con được gọi là *upcasting*.

Ví dụ: giả sử phương thức getObject() trả về một đối tượng nhưng nó có thể là bất kỳ kiểu nào như Employee, Student... chúng ta có thể sử dụng biến tham chiếu của lớp Object để tham chiếu tới đối tượng đó.

```
Object obj = getObject();//Chúng ta không biết đối tượng nào sẽ được trả về từ phương thức này
```

Lớp Object cung cấp một vài cách xử lý chung cho tất cả các đối tượng như đối tượng có thể được so sánh, đối tượng có thể được cloned, đối tượng có thể được notified...

Các phương thức của lớp Object

Lớp Object cung cấp các phương thức như trong bảng sau:

Phương thức	Mô tả
public final Class getClass()	trả về đối tượng lớp Class của đối tượng hiện tại. Từ lớp Class đó có thể lấy được các thông tin metadata của class hiện tại.
public int hashCode()	trả về số hashcode cho đối tượng hiện tại.
public boolean equals(Object obj)	so sánh đối tượng đã cho với đối tượng hiện tại.
protected Object clone() throws CloneNotSupportedException	tạo và trả về bản sao chép (clone) của đối tượng hiện tại.
public String toString()	trả về chuỗi ký tự đại diện của đối tượng hiện tại.
public final void notify()	đánh thức một luồng, đợi trình giám sát của đối tượng hiện tại.

<code>public final void notifyAll()</code>	đánh thức tất cả các luồng. đợi trình giám sát của đối tượng hiện tại.
<code>public final void wait(long timeout)throws InterruptedException</code>	làm cho Thread hiện tại đợi trong khoảng thời gian là số mili giây cụ thể, tới khi Thread khác thông báo (gọi phương thức <code>notify()</code> hoặc <code>notifyAll()</code>).
<code>public final void wait(long timeout,int nanos)throws InterruptedException</code>	làm cho Thread hiện tại đợi trong khoảng thời gian là số mili giây và nano giây cụ thể, tới khi Thread khác thông báo (gọi phương thức <code>notify()</code> hoặc <code>notifyAll()</code>).
<code>public final void wait()throws InterruptedException</code>	làm Thread hiện tại đợi, tới khi Thread khác thông báo (gọi phương thức <code>notify()</code> hoặc <code>notifyAll()</code>).
<code>protected void finalize()throws Throwable</code>	Được gọi bởi Garbage Collector trước khi đối tượng bị dọn rác.

Khai báo Object

Một Object (đối tượng) nó chứa trong đó bao gồm các method (phương thức) và properties (thuộc tính) để tạo ra một kiểu dữ liệu hữu ích.

Object xác định hành vi của class. Khi bạn gửi một thông điệp vào một object, có nghĩa là bạn đang yêu cầu gọi các object hoặc thực hiện một trong những phương thức của nó.

Từ một quan điểm của lập trình hướng đối tượng, một đối tượng có thể là một cấu trúc dữ liệu (data structure), một biến (variable) hoặc một chức năng (function).

Object được phân bổ vị trí bộ nhớ. Các Object được thiết kế như class phân cấp.

```
ClassName ReferenceVariable = new ClassName();
```

Điểm khác biệt giữa Class và Object trong Java

Một **Class** là một **Blueprint** (kế hoạch) hay **Prototype** (nguyên mẫu) xác định biến và các phương thức (hay function) chung với tất cả các đối tượng cùng loại.

Một **Object** là một cụ thể của một Class. Các đối tượng thường được dùng để mô tả đối tượng trong thế giới thực mà bạn thấy hàng ngày.

Revision #3

Created 24 September 2019 18:23:53 by Admin

Updated 12 April 2020 14:36:02 by Laptrinh.vn