

Number trong Java

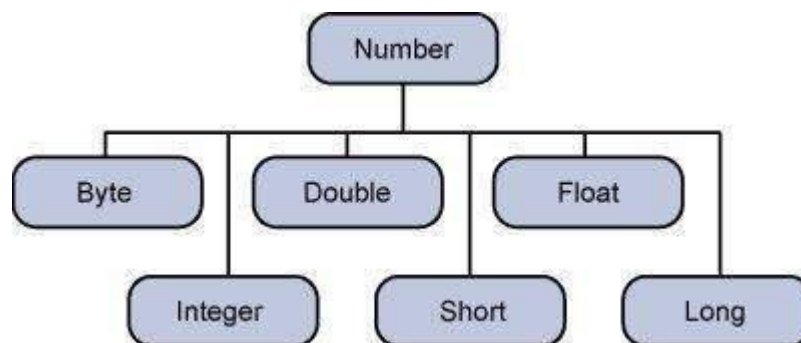
Thường thì, khi làm việc với Number, chúng ta sử dụng các kiểu dữ liệu gốc như byte, int, long, double, ...

Ví dụ:

```
int i = 5000;  
float gpa = 13.65;  
byte mask = 0xaf;
```

Tuy nhiên, khi mở rộng, chúng ta cũng gặp các tình huống mà chúng ta cần sử dụng các đối tượng thay cho các kiểu dữ liệu gốc. Để thực hiện điều này, Java cung cấp các lớp wrapper (lớp bọc) cho mỗi kiểu dữ liệu gốc.

Tất cả các lớp wrapper (Integer, Long, Byte, Double, Float, Short) là lớp phụ của lớp trừu tượng Number.



Sự bao bọc này được chăm sóc bởi bộ biên dịch, tiến trình này được gọi là boxing. Vì thế khi một kiểu dữ liệu gốc được sử dụng khi một đối tượng được yêu cầu, bộ biên dịch "đóng hộp" kiểu dữ liệu gốc trong lớp wrapper của nó. Tương tự, bộ biên dịch cũng "mở hộp" đối tượng thành một kiểu dữ liệu gốc. *Number* là một phần của java.lang package.

Dưới đây là ví dụ của box và unbox:

```
public class Test{  
  
    public static void main(String args[]){  
        Integer x = 5; // đóng hộp int thành đối tượng Integer  
        x = x + 10;    // mở hộp Integer thành int  
    }  
}
```

```
System.out.println(x);  
}  
}
```

Nó sẽ cho kết quả sau:

15

Khi x được gán các giá trị integer, bộ biên dịch "đóng hộp" giá trị integer này bởi vì x là đối tượng integer. Sau đó, x được "mở hộp" để mà chúng có thể được cộng thêm các integer.

Các phương thức của lớp Number trong Java

Bảng dưới liệt kê các phương thức mà tất cả các lớp phụ của lớp Number thực thi:

STT	Phương thức và miêu tả
1	<u>xxxValue() trong Java</u> Biến đổi giá trị của đối tượng Number này thành kiểu dữ liệu xxx mà được trả về từ phương thức
2	<u>compareTo() trong Java</u> So sánh đối tượng Number này với tham số
3	<u>equals() trong Java</u> Kiểm tra có hay không đối tượng Number này là bằng với tham số
4	<u>valueOf() trong Java</u> Trả về một đối tượng integer giữ giá trị gốc đã xác định
5	<u>toString() trong Java</u>

	Trả về đối tượng String biểu diễn giá trị Integer đã xác định
6	<u>parseInt() trong Java</u> Phương thức này được sử dụng để nhận kiểu dữ liệu gốc của một String cụ thể
7	<u>abs() trong Java</u> Trả về giá trị tuyệt đối của tham số
8	<u>ceil() trong Java</u> Trả về số integer nhỏ nhất mà lớn hơn hoặc bằng tham số. Giá trị trả về ở dạng double.
9	<u>floor() trong Java</u> Trả về số integer lớn nhất mà nhỏ hơn hoặc bằng tham số. Giá trị trả về ở dạng double.
10	<u>rint() trong Java</u> Làm tròn số, trả về giá trị integer gần nhất với giá trị tham số. Giá trị trả về ở dạng double.
11	<u>round() trong Java</u> Trả về long hoặc int gần nhất, như kiểu dữ liệu được chỉ dẫn bởi phương thức, với tham số.
12	<u>min() trong Java</u> Trả về số nhỏ nhất trong hai tham số

13	<u>max() trong Java</u> Trả về số lớn nhất trong hai tham số
14	<u>exp() trong Java</u> Hàm mũ cơ số e.
15	<u>log() trong Java</u> Hàm logarit cơ số e
16	<u>pow() trong Java</u> Trả về giá trị hàm mũ với cơ số là tham số đầu tiên và mũ là tham số thứ hai
17	<u>sqrt() trong Java</u> Trả về căn bậc hai của tham số
18	<u>sin() trong Java</u> Trả về sin của giá trị double đã cho.
19	<u>cos() trong Java</u> Trả về cos của giá trị double đã cho.
20	<u>tan() trong Java</u> Trả về tan của giá trị double đã cho.
21	<u>asin() trong Java</u>

	Trả về arcsin của giá trị double đã cho.
22	<u>acos() trong Java</u> Trả về arccos của giá trị double đã cho
23	<u>atan() trong Java</u> Trả về arctan của giá trị double đã cho
24	<u>atan2() trong Java</u> Biến đổi tọa độ (x, y) thành tọa độ cực (r, theta) và trả về theta.
25	<u>toDegrees() trong Java</u> Biến đổi tham số thành giá trị độ đo góc
26	<u>toRadians() trong Java</u> Biến đổi tham số thành radian.
27	<u>random() trong Java</u> Trả về một số ngẫu nhiên

Chương tiếp bàn luận vấn đề gì trong Java?

Trong chương tới, chúng ta sẽ xem xét lớp Character trong Java. Bạn sẽ học cách sử dụng đối tượng Character và kiểu dữ liệu char gốc trong Java.