

Bài: Ép kiểu trong Java

Xem bài học trên website để ủng hộ Kteam: [Ép kiểu trong Java](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở bài trước, Kteam đã giới thiệu cho các bạn về [CÁC HẠNG TOÁN TỬ](#) & [KIỂU DỮ LIỆU](#).

Trong lập trình ta sẽ gặp một số trường hợp khi lúc đầu ta khai báo biến ở kiểu dữ liệu này, nhưng sau đó ta cần chuyển nó sang kiểu dữ liệu khác để phù hợp trong tính toán. Chính vì vậy, bài viết này Kteam sẽ hướng dẫn các bạn cách **ép kiểu trong Java**.

Nội dung

Để đọc hiểu bài này, tốt nhất các bạn nên có kiến thức cơ bản về các phần sau:

- [BIẾN TRONG JAVA](#)
- [CÁC KIỂU DỮ LIỆU TRONG JAVA](#)
- [CÁC TOÁN TỬ TRONG JAVA](#)

Bài này chúng ta sẽ tìm hiểu những vấn đề sau:

- Ép kiểu là gì? Ý nghĩa
- Cách sử dụng ép kiểu

Ép kiểu là gì? Ý nghĩa

Ép kiểu là cách chuyển biến thuộc kiểu dữ liệu này thành biến thuộc kiểu dữ liệu khác.

Ý nghĩa:

- Việc chuyển kiểu dữ liệu sẽ đến lúc phải cần trong quá trình xử lý chương trình
- Có thể **định dạng đúng kiểu dữ liệu** mình **mong muốn** (Như cách hiển thị kiểu ngày tháng năm trên thế giới khác với Việt Nam nên ta sẽ chuyển kiểu ngày theo phong cách địa phương).

Cách sử dụng ép kiểu

Trong bài này chỉ nói đến ép kiểu dữ liệu đối với **dữ liệu nguyên thủy (Primitive Data Types)**, còn đối với ép kiểu **dữ liệu tham chiếu (Reference Types)** thì cách ép kiểu là những **hàm (phương thức) ép kiểu** do người ta viết riêng cho mỗi kiểu dữ liệu tham chiếu đó.

Thì trong ép kiểu trong kiểu dữ liệu nguyên thủy được chia ra làm 2 loại:

- Chuyển đổi kiểu ngầm định (implicit)
- Chuyển đổi kiểu tường minh (explicit)

Kiểu chuyển đổi ngầm định (implicit)

Việc chuyển đổi sẽ tự thực hiện bởi **compiler** và chúng ta không cần làm gì. Việc chuyển đổi này dành cho **kiểu dữ liệu nhỏ sang kiểu dữ liệu lớn hơn**. Ta có thể xem chiều từ nhỏ sang lớn như sau:



Ví dụ: Ta lấy một biến kiểu **int** gán giá trị cho biến kiểu **long**

java:

```

public class HelloWorld{

    public static void main(String []args){
        int a = 5;
        long b = a;
        System.out.print(b);
    }
}
  
```

```

$javac HelloWorld.java
$java HelloWorld
5
  
```

Kiểu chuyển đổi tường minh (explicit)

Ngược lại với cách **chuyển đổi ngầm định**, việc **chuyển đổi tường minh** là chiều ngược lại từ **kiểu dữ liệu lớn hơn sang kiểu dữ liệu nhỏ hơn** (với điều kiện giá trị đó kiểu dữ liệu sẽ thay đổi có thể lưu trữ được trong kiểu dữ liệu mới).



Với ép kiểu theo cú pháp:

(<Kiểu dữ liệu>) <Tên biến>;

Ví dụ: Ta lấy một biến kiểu **long** gán giá trị cho biến kiểu **int**

java:

```

public class HelloWorld{

    public static void main(String []args){
        long a = 6;
        int b = (int) a;
        System.out.print(a);
    }
}
  
```

```
$javac HelloWorld.java
$java HelloWorld
6
```

h

Lưu ý: Nếu ép kiểu dữ liệu kí tự **char** sang **kiểu dữ liệu số** hoặc ngược lại.

- Khi ép kiểu char sang số thì sẽ ép kiểu ngầm định chuyển kí tự sang hệ thập phân ASCII tương ứng kí tự đó.
- Nếu ngược lại thì phải sử dụng ép kiểu tường minh để chuyển sang kiểu kí tự.

Ví dụ: Ta sẽ thử kí tự 'A' được biết hệ thập phân ASCII tương ứng

java:

```
public class HelloWorld{

    public static void main(String []args){
        char a = 'A';
        int b = a;
        char c = (char) b;
        System.out.println(b);
        System.out.println(c);
    }
}
```

```
$javac HelloWorld.java
$java HelloWorld
65
A
```

h

Kết

Như vậy chúng ta đã tìm hiểu ép kiểu dữ liệu trong Java

Ở bài sau, Kteam sẽ giới thiệu đến bạn về [CẤU TRÚC RẼ NHÁNH JAVA](#)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "Luyện tập – Thử thách – Không ngại khó".