

🕒 Cập nhật tháng 8 năm 2024

[Bài Đọc] Reference Method

1. Khái niệm Tham chiếu của Phương thức

- **Tham chiếu của Phương thức (Method Reference)** là một tính năng trong Java 8, cho phép bạn tham chiếu trực tiếp đến một phương thức hoặc constructor bằng cách sử dụng cú pháp ::
- Là một cách viết ngắn gọn của biểu thức lambda khi nó chỉ gọi một phương thức đã tồn tại
- Ví dụ:

- Sử dụng Lambda:

```
Consumer<String> print = s -> System.out.println(s);
```

- Sử dụng Method references

```
Consumer<String> print = System.out::println;
```

2. Cú pháp Method References

- Có bốn loại tham chiếu của phương thức trong Java:
- **Tham chiếu đến phương thức tĩnh**
 - Cú pháp: `ClassName::staticMethodName`
 - Sử dụng khi gọi một phương thức tĩnh của một lớp
 - Ví dụ:

```
public static void main(String[] args) {
    new *
    {
        Function<String, Integer> parseInt = Integer::parseInt; // Tham chiếu đến Integer.parseInt
        System.out.println(parseInt.apply("123")); // Output: 123
    }
}
```

- **Tham chiếu đến phương thức của một đối tượng**
 - Cú pháp: `instance::instanceMethodName`
 - Sử dụng khi gọi một phương thức không tĩnh của một đối tượng cụ thể
 - Ví dụ:

```
public static void main(String[] args) new *
{
    String message = "Hello, Java!";
    Runnable print = message::toUpperCase; // Tham chiếu đến toUpperCase của chuỗi cụ thể
    System.out.println(message.toUpperCase()); // Output: HELLO, JAVA!
}
```

• Tham chiếu đến Constructor

- Cú pháp: `ClassName::new`
- Sử dụng để tạo một đối tượng mới
- Ví dụ:

```
public static void main(String[] args) new *
{
    Supplier<StringBuilder> create = StringBuilder::new; // Tham chiếu đến constructor của StringBuilder
    StringBuilder sb = create.get();
    System.out.println(sb.append("Hello")); // Output: Hello
}
```

3. So sánh Lambda và Method References

Đặc điểm	Lambda expression	Method references
Độ dài mã nguồn	Dài hơn nếu chỉ gọi một phương thức	Ngắn gọn hơn
Tính rõ ràng	Có thể kém rõ ràng với đoạn mã phức tạp	Rõ ràng khi tham chiếu đến phương thức cụ thể
Hiệu năng	Như nhau	Như nhau
Tính linh hoạt	Linh hoạt hơn	Ít linh hoạt hơn

Link tài nguyên đọc thêm: <https://www.baeldung.com/java-method-references>

Danh sách các bài học

