


 Cập nhật tháng 8 năm 2024

[Bài đọc] Các phương thức làm việc với Queue

1. Làm việc với Queue trong Java

- Trong Java, **Queue** là một **interface** trong gói `java.util`, thường được triển khai bởi các lớp như:
 - **LinkedList** (Queue thông thường)
 - **PriorityQueue** (có sắp xếp thứ tự)
 - **ArrayDeque** (hàng đợi hai đầu)
- Khởi tạo Queue



```
import java.util.Queue;  
import java.util.LinkedList;  
  
Queue<String> queue = new LinkedList<>();
```

2. Các phương thức phổ biến trong Queue

- **add(E e)** – Thêm phần tử vào hàng đợi
 - Thêm phần tử vào cuối hàng
 - Ném exception nếu không thể thêm được

```
import java.util.Queue;
import java.util.LinkedList;

Queue<String> queue = new LinkedList<>();

queue.add("A");
```

- **offer(E e)** – Thêm phần tử (an toàn hơn add)
 - Cũng thêm vào cuối hàng nhưng trả về false nếu không thêm được (thay vì ném exception)

```
import java.util.Queue;
import java.util.LinkedList;

Queue<String> queue = new LinkedList<>();

queue.offer("B");
```

- **remove()** – Lấy và xóa phần tử đầu hàng
 - Xóa phần tử đầu tiên
 - Ném NoSuchElementException nếu hàng rỗng

```
import java.util.Queue;
import java.util.LinkedList;

Queue<String> queue = new LinkedList<>();

String head = queue.remove();
```

- **poll()** - Lấy và xóa phần tử đầu hàng (an toàn hơn remove)
 - Giống remove(), nhưng **trả về null** nếu hàng rỗng

```
import java.util.Queue;
import java.util.LinkedList;

Queue<String> queue = new LinkedList<>();

String head = queue.poll();
```

- **element()** - Xem phần tử đầu hàng
 - Trả về phần tử đầu tiên mà không xóa
 - Ném NoSuchElementException nếu hàng rỗng

```
import java.util.Queue;
import java.util.LinkedList;

Queue<String> queue = new LinkedList<>();

String head = queue.element();
```

- **peek()** - Xem phần tử đầu hàng (an toàn hơn element)
 - Giống element(), nhưng **trả về null** nếu hàng rỗng

```
import java.util.Queue;
import java.util.LinkedList;

Queue<String> queue = new LinkedList<>();

String head = queue.peek();
```

3. Tổng kết sự khác nhau giữa các cặp phương thức

Mục đích	Có ném Exception	Không ném Exception
Thêm phần tử	add()	offer()
Xóa phần tử đầu tiên	remove()	poll()
Xem phần tử đầu tiên	element()	peek()

4. Ví dụ tổng hợp:

```
import java.util.*;

public class QueueExample {
    public static void main(String[] args) {
        Queue<String> queue = new LinkedList<>();

        queue.offer("A");
        queue.offer("B");
        queue.offer("C");

        System.out.println("Phần tử đầu tiên: " + queue.peek()); // A

        queue.poll(); // Xóa A
        System.out.println("Sau khi poll: " + queue); // [B, C]

        System.out.println("Phần tử tiếp theo: " + queue.element()); // B
    }
}
```

