

13. [Java] ?????4: stack ? int[] ??

<https://school.programmers.co.kr/learn/courses/30/lessons/181918?language=java>

1. ?? ??

```
import java.util.*;

class Solution {
    public int[] solution(int[] arr) {
        Stack<Integer> stk = new Stack<>();
        int i = 0;

        while (i < arr.length) {
            if (stk.isEmpty()) {
                stk.push(arr[i]);
                i++;
            } else if (stk.peek() < arr[i]) {
                stk.push(arr[i]);
                i++;
            } else {
                stk.pop();
            }
        }

        // stack → int[]
        int[] answer = new int[stk.size()];
        for (int j = 0; j < stk.size(); j++) {
            answer[j] = stk.get(j);
        }
        return answer;
    }
}
```

```
}
```

2. for? ?? stack ? int[] ??

```
// stack → int[]
int[] answer = new int[stk.size()];
for (int j = 0; j < stk.size(); j++) {
    answer[j] = stk.get(j);
}
return answer;
```

- `for`
 - `for` 루프 (반복문) 사용
 - `stk.get(j)` 사용
 - `answer[j] = stk.get(j);`
- `forEach`
 - `stk.forEach` 사용

3. stream() ?? stack ? int[] ??

```
// stream을 이용해 stack → int[] 변환
return stk.stream().mapToInt(Integer::intValue).toArray();
```

- `stk.stream() : Stack<Integer> → Stream<Integer>`
- `.mapToInt(Integer::intValue) : Stream<Integer> → IntStream`
- `.toArray() : IntStream → int[]`
- `forEach`
 - `forEach` 사용, `Integer::intValue` 사용
 - `IntStream` 사용
- `forEachOrdered`
 - `forEachOrdered` 사용 (iterator) `Integer::intValue` 사용 → `IntStream` 사용
 - `IntStream` 사용

4. ?? ??

1000개 요소의 스택	for-loop	stream
1000	≈ 0.01ms	≈ 0.01ms

100,000	1.5ms	2.2ms
1,000,000	15ms	22ms

100,000 : 1.5ms stream() 2.2ms
 1,000,000 15ms 22ms

- 100,000 1.5ms 2.2ms (ex. 100,000 1.5ms , 1,000,000 ~15ms 22ms)
 for 100,000 1.5ms .
- 1,000,000 15ms 22ms (ex. 1000 1000) stream() 1000 1000 .