중복된 숫자 개수

종료일	@2023년 5월 24일
태그	base java propgrammers
상태	Done
작업 시간	5
# Level	0

제목 입력

> 프로그래머스 알고리즘 문제 URL
>
> https://school.programmers.co.kr/learn/courses/30/lessons/120583

문제 설명

▼ 문제

정수가 담긴 배열 `array`와 정수 `n`이 매개변수로 주어질 때, `array`에 `n`이 몇 개 있는지를 return 하도록 solution 함수를 완성해보세요.

제한사항

▼ 제한사항

- 1 ≤ `array`의 길이 ≤ 100
- 0 ≤ `array`의 원소 ≤ 1,000
- 0 ≤ `n` ≤ 1,000

입출력 예

▼ 입출력 예

array	n	result
[1, 1, 2, 3, 4, 5]	1	2
[0, 2, 3, 4]	1	0

▼ 입출력 예 설명

입출력 예 #1

- [1, 1, 2, 3, 4, 5] 에는 1이 2개 있습니다.

입출력 예 #2

- [0, 2, 3, 4] 에는 1이 0개 있습니다.

풀이

▼ 풀이과정

문제 설명처럼 배열에 있는 값에서 전달받은 인수에서 비교하여 값은 수인지 판단후 카운트 증가 해주는 식
for문과 삼항연산자를 이용하여 수 비교후 숫자를 더해줌

▼ 테스트 전체 코드

내용

```
package level_0.dupleNum;

public class Main {
  public static void  main(String[]  args)throws Exception {
    Solution solution = new Solution();
    System.out.println("결과 값" + solution.solution(new int[] {1,1,2,3,4,5}, 1));
  }

  public static class Solution{
    // TODO  중복된 숫자 개수
    public int solution(int[] array, int n) {
      int answer = 0;
      for (int i : array) {
        answer += i == n ? 1 : 0;
      }
      return answer;
    }
```

```
  }
}
```

▼ 다른 사람 풀이

```
import java.util.*;
class Solution {
    public int solution(int[] array, int n) {
        return (int) Arrays.stream(array)
                .filter(e -> e == n)
                .count();
    }
}
```

Java8 의 스트림을 사용하여 카운터를 증가해주는 방법으로 문제를 해결한 코드

스트림(Streams)은 람다를 활용할 수 있는 기술 중 하나이며, Java8 이전에는 배열 또는 컬렉션 인스턴스를 다루는 방법은 for 또는 foreach 문을 돌면서 요소 하나씩을 꺼내서 다루는 방법이 있다. 간단한 경우라면 상관없지만 로직이 복잡해질수록 코드의 양이 많아져 여러 로직이 섞이게 되고, 메소드를 나눌 경우 루프를 여러 번 도는 경우가 발생, `스트림은 '데이터의 흐름'` 으로 배여 또는 컬렉션의 인스턴스에 함수 여러 개를 조합해서 원하는 결과를 필터링하고 가공된 결과를 얻을 수 있다. 또한 람다를 이용해서 코드의 양을 중리고 간결하게 표현할 수 있다. 즉, 배열과 컬렉션을 함수형으로 처리할 수 있다.

- 생성하기
  - 배열 / 컬렉션 / 빈 스트림
  - *Stream.builder()* / *Stream.generate()* / *Stream.iterate()*
  - 기본 타입형 / *String* / 파일 스트림
  - 병렬 스트림 / 스트림 연결하기
- 가공하기
  - Filtering
  - Mapping
  - Sorting
  - Iterating
- 결과 만들기
  - Calculating
  - Reduction

- Collecting
- Matching
- Iterating

자세한 내용은 아래의 페이지를 참고하자

Stream