If you use the nextLine() method immediately following the nextInt() method, recall that nextInt() reads integer tokens; because of this, the last newline character for that line of integer input is still queued in the input buffer and the next nextLine() will be reading the remainder of the integer line (which is empty).

The `nextInt()` method only reads the integer value, the newline character which will be present because of the user pressing enter,is saved in the buffer. This means that if you call the `nextLine()` method immediately after the `nextInt()` method, it will read the newline character as part of the input line.

To avoid this, you can call the `skip()` method before calling the `nextLine()` method. The `skip()` method will skip any whitespace characters, including newline characters, from the input buffer.

For example:

Code snippet

```
import java.util.Scanner;

public class Solution {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        int i = scan.nextInt();
        scan.skip("\n");
        String s = scan.nextLine();

        System.out.println("String: " + s);
        System.out.println("Int: " + i);
    }
}
```

This code will first read an integer value using the `nextInt()` method. Then, it will call the `skip()` method to skip any whitespace characters from the input buffer. Finally, it will read a string value using the `nextLine()` method.

When you run this code, it will prompt you to enter an integer value. Enter the value and then press Enter. The code will then read the next line of input, which will be the string value. The code will then print the values that it read to the console.

Sources

1. https://www.iditect.com/how-to/52311881.html