

There are two ways to flush stdin in Java:

- Using the `System.in.flush()` method

The `System.in.flush()` method flushes the input stream, which forces any buffered data to be written immediately. This method is defined in the `java.io.InputStream` class, which is the superclass of `System.in`.

For example:

#### Code snippet

```
import java.io.InputStream;

public class FlushStdin {

    public static void main(String[] args) {
        InputStream in = System.in;
        in.read(); // Read one character from stdin
        in.flush(); // Flush the input stream
        in.read(); // Read another character from stdin
    }
}
```

In this example, the first `read()` method will read one character from stdin and store it in a buffer. The `flush()` method will then force the buffered data to be written immediately. The second `read()` method will then read the next character from stdin.

- Using the `Scanner.next()` method

The `Scanner.next()` method also flushes stdin. This is because the `Scanner` class internally uses the `System.in.read()` method to read data from stdin. When the `Scanner.next()` method is called, it will first flush the input stream to make sure that any buffered data is written immediately.

For example:

#### Code snippet

```
import java.util.Scanner;

public class FlushStdin {

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

```
        scanner.next(); // Flush stdin and read the next line from
stdin
    }
}
```

In this example, the `next()` method will flush stdin and then read the next line from stdin.

Which method you use to flush stdin depends on your specific needs. If you need to flush stdin immediately, then you can use the `System.in.flush()` method. If you are using a `Scanner` object to read data from stdin, then you can use the `Scanner.next()` method, which will automatically flush stdin for you.