02-Command-Line-Input Notes

theKeyboard

System.in

Keyboard

Java created the **System.in** object to represent the keyboard

We must define a **Scanner object** the "wraps" **System-in**

`theKeyboard.nextLine()`

Program

`System.out.println()`

System.out object

Computer Screen

Java created the **System.out** object to represent the screen
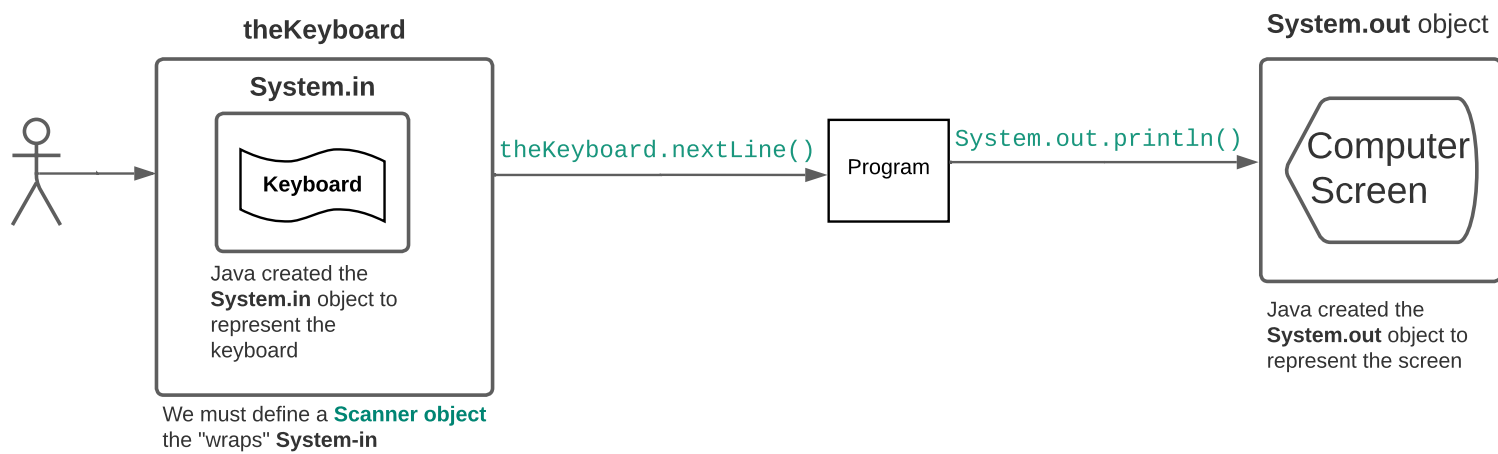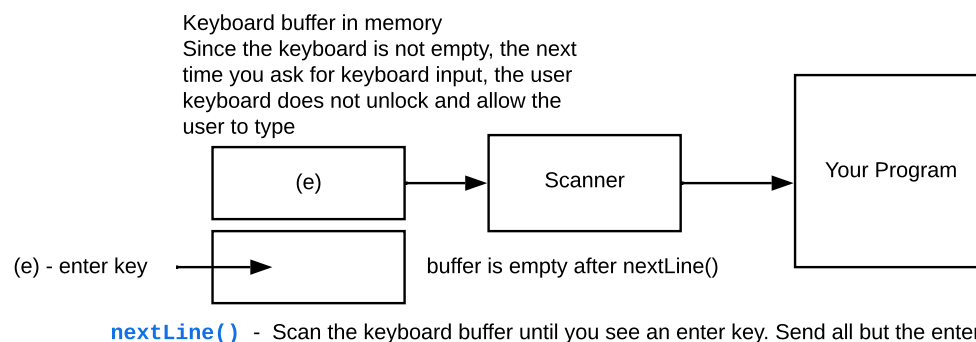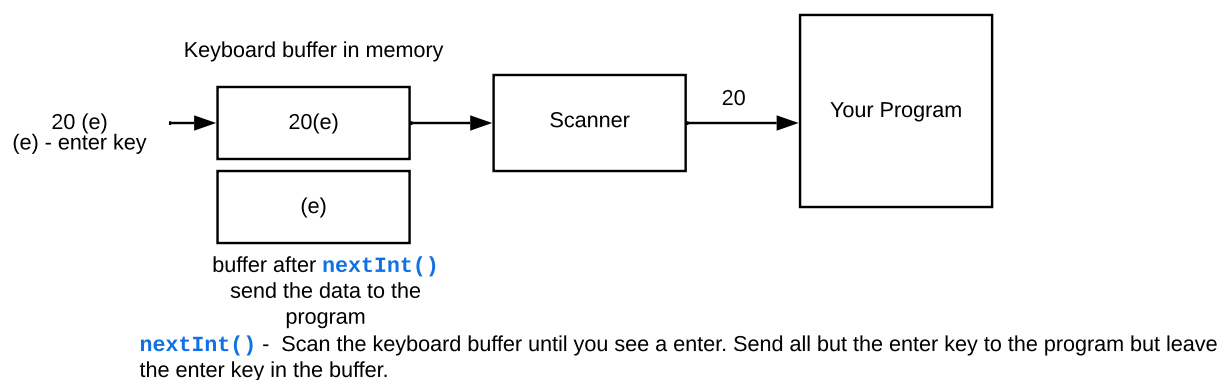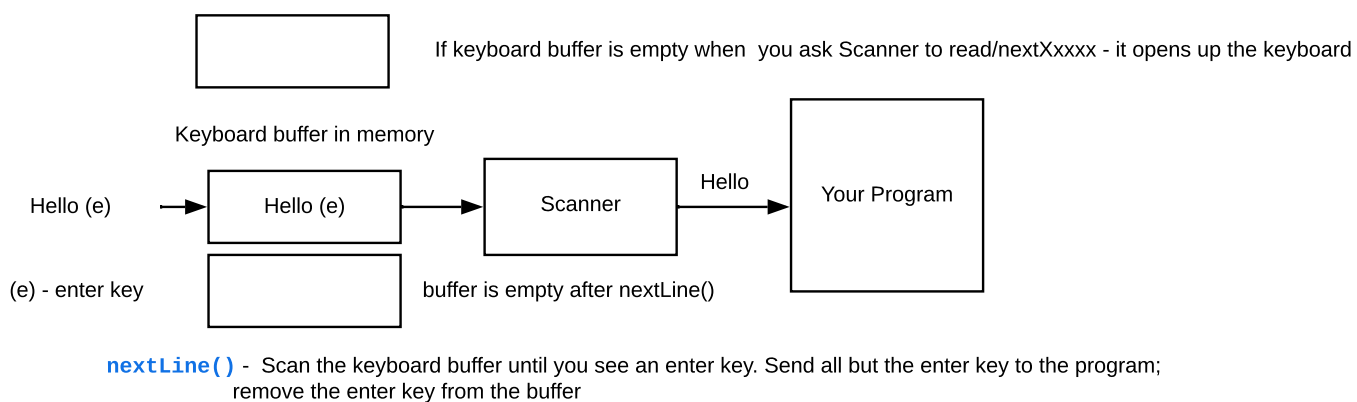
Any keyboard input is saved in a memory area called a buffer as it is typed by the user.

Scanner will scan the keyboard buffer until it finds an invalid character or enter.

When it finds the enter key, it returns all characters from the start of the scan up to but not including the enter key.

`nextLine()` **removes the enter key from the buffer**;
`nextInt()`, `nextDouble` and any other numeric next **leaves the enter key in the buffer**.

If it finds an invalid character it throws an Exception (error).

If keyboard buffer is empty when you ask Scanner to read/nextXxxxx - it opens up the keyboard

Keyboard buffer in memory

Hello (e) → Hello (e) → Scanner → Hello → Your Program

(e) - enter key

buffer is empty after nextLine()

`nextLine()` - Scan the keyboard buffer until you see an enter key. Send all but the enter key to the program; remove the enter key from the buffer

Keyboard buffer in memory

20 (e)
(e) - enter key → 20(e) → Scanner → 20 → Your Program

(e)

buffer after `nextInt()` send the data to the program

`nextInt()` - Scan the keyboard buffer until you see a enter. Send all but the enter key to the program but leave the enter key in the buffer.

Keyboard buffer in memory
Since the keyboard is not empty, the next time you ask for keyboard input, the user keyboard does not unlock and allow the user to type

(e) → Scanner → Your Program

(e) - enter key

buffer is empty after nextLine()

`nextLine()` - Scan the keyboard buffer until you see an enter key. Send all but the enter key to the program

To avoid issues when reading from the keyboard:

1. Use **nextLine()** to read the input as a String
2. Convert the String to numeric value if needed using **Integer.parseInt()** or **Double.parseDouble()**, et al

-OR-

if you use **nextInt()**, **nextDouble()** or any other next with a numeric type,
do a **nextLine()** immediately afterwards to clear the keyboard buffer of the enter key left in it.