AP_API: Scanners (Kailyn & Liver)

The following is adapted from https://www.w3schools.com/java/java_user_input.asp by Kailyn and Liver.

Basics (K&L)

The *Scanner* class is in the *java.util* package. You can use it to get user input from the command line.

Scanner can read and save primitive data types as well as *Strings*.

Example 1 (K)

This example uses the *nextLine()* method, which is used to read Strings:

```
import java.util.Scanner; // Import the Scanner class

class Main {
  public static void main(String[] args) {
    Scanner myObj = new Scanner(System.in); // Create a Scanner object

    System.out.println("Enter username");
    String userName = myObj.nextLine(); // Read user input

    System.out.println("Username is: " + userName); // Output user input
  }
}
```

Input Types (K)

nextBoolean() Reads a *boolean* value from the user

nextByte() Reads a *byte* value from the user

nextDouble() Reads a *double* value from the user

nextFloat() Reads a *float* value from the user

nextInt() Reads a *int* value from the user

nextLine() Reads a *String* value from the user

nextLong() Reads a *long* value from the user

nextShort() Reads a *short* value from the user

Example 2 (K&L)

```
import java.util.Scanner;

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

    System.out.println("Enter name, age and salary:");
    String name = myObj.nextLine(); // String input
    int age = myObj.nextInt(); // Numerical inputs
    double salary = myObj.nextDouble();
    // Please note: User should input specified values with spaces between
    // or they will all be interpreted as one thing by the Scanner.
    // Refer to "Other Info" for why this happens.

    System.out.println("Name: " + name); // Output input by user
    System.out.println("Age: " + age);
    System.out.println("Salary: " + salary);
  }
}
```

The following is original by Liver.

Clearing the “Buffer” (L)

When scanning anything other than *nextLine()*, it might be wise to clear the scanner's "buffer" by writing this in your code:

```
[Scanner object].nextLine();
```

This will scan all the way to the end of the line and create a new line for the user to input properly.

Not doing this might cause an error, as the scanner has not left the line that they previously scanned and thus might scan the previous line again.

The following is adapted from https://www.javatpoint.com/post/java-scanner-hasnext-method and https://origin.geeksforgeeks.org/scanner-class-in-java/?ref=lbp by Kailyn and Liver.

hasNext() Method (K&L)

Java *Scanner hasNext()* Method: *Scanner* class method that returns true if this scanner has another token in its input. This method may block while waiting for input to scan.

Java *Scanner hasNext(String pattern)* Method: *Scanner* class method that returns true if the next token matches the pattern constructed from the specified string.

Java *Scanner hasNext(Pattern pattern)* Method: *Scanner* class method that returns true if the next complete token matches the specified pattern.

Example 3 (K)

```
import java.util.Scanner;

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

         int sum = 0, count = 0;
```

```
        // Checks if an int value is available
        while (myObj.hasNextInt()) {
            // Read an int value
            int num = myObj.nextInt();
             sum += num;
            count++;
        }
        if (count > 0) {
            int mean = sum / count;
            System.out.println("Mean: " + mean);
        } else {
            System.out.println(
             "No integers were input. Mean cannot be calculated.");
        }
    }
}
```

Similar Methods (K)

hasNextBigDecimal()
hasNextByte()
hasNextDouble()
hasNextFloat()
hasNextInt()
hasNextLine()
hasNextLong()
hasNextShort()

Other Information (K&L)

To read a single character, you can use *next().charAt([index])*. The *charAt([index])* function will return the character at position *[index]*.

The *Scanner* class reads an entire line and divides it into tokens. Tokens are separated by spaces.

For example, the *Scanner* object will read *"How are you"* and divide the string into tokens: *“How”*, *“are”* and *“you”*. The object will then take each token and read it using the methods called.

Formatting Credits: L