Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No.2
Accepting Input Through Keyboard
Date of Performance:
Date of Submission:

Aim: To apply basic programing for accepting input through keyboard.

Objective: To use the facility of java to read data from the keyboard for any program

Theory:

Java brings various Streams with its I/O package that helps the user perform all the Java input-output operations. These streams support all types of objects, data types, characters, files, etc. to fully execute the I/O operations. Input in Java can be with certain methods mentioned below in the article.

Methods to Take Input in Java

There are two ways by which we can take Java input from the user or from a file

1. BufferedReader Class
2. Scanner Class

Using BufferedReader Class for String Input In Java

It is a simple class that is used to read a sequence of characters. It has a simple function that reads a character another read which reads, an array of characters, and a readLine() function which reads a line.

InputStreamReader() is a function that converts the input stream of bytes into a stream of characters so that it can be read as BufferedReader expects a stream of characters. BufferedReader can throw checked Exceptions.

Using Scanner Class for Taking Input in Java

It is an advanced version of BufferedReader which was added in later versions of Java. The scanner can read formatted input. It has different functions for different types of data types.

The scanner is much easier to read as we don't have to write throws as there is no exception thrown by it.

It was added in later versions of Java

It contains predefined functions to read an Integer, Character, and other data types as well.

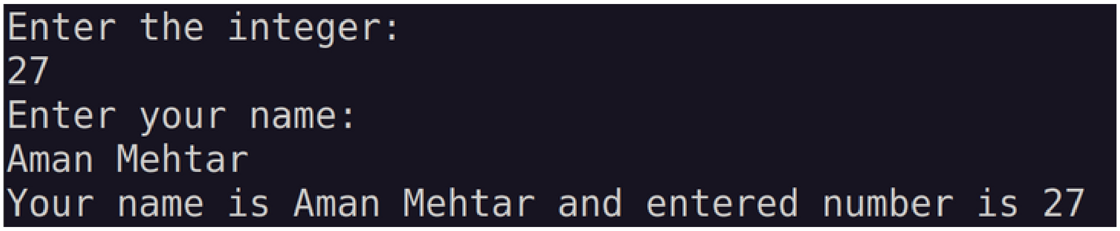
Syntax of Scanner class

Scanner scn = new Scanner(System.in);

Code for scanner class:

```
import java.io.*;
import java.util.*;
class hello {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the integer:");
      int num = sc.nextInt();
      System.out.println("Enter your name:");
      String name= sc.nextLine();
      name+=sc.nextLine();
      System.out.println("Your name is "+name+" and entered number is "+num);
      sc.close();
   }
}
```

Output:

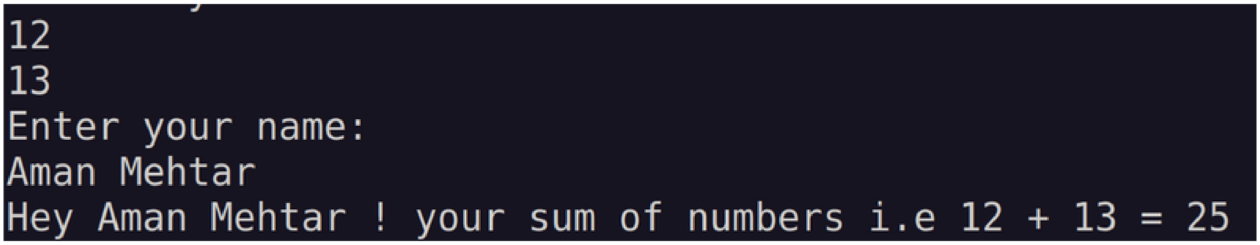


```
Enter the integer:
27
Enter your name:
Aman Mehtar
Your name is Aman Mehtar and entered number is 27
```

Code for scanner class:

```
import java.io.*;

import java.util.*;

class hello {

   public static void main(String[] args)

      throws IOException

    {

      BufferedReader bf= new BufferedReader(new InputStreamReader(System.in));

      System.out.println("Enter any two numbers:");

      int a=Integer.parseInt(bf.readLine());

      int b =Integer.parseInt(bf.readLine());

      System.out.println("Enter your name:");

      String name= bf.readLine();

      System.out.println("Hey "+name+" ! your sum of numbers i.e "+a+" + "+b+" = "+(a+b));


   }

}
```

Output:



```
12
13
Enter your name:
Aman Mehtar
Hey Aman Mehtar ! your sum of numbers i.e 12 + 13 = 25
```

Conclusion:

In my experience, both BufferedReader and Scanner serve distinct purposes when accepting user input in Java. BufferedReader is optimal for high-performance scenarios where bulk reading of data is required, especially in applications involving large datasets or file processing. On the other hand, Scanner is ideal for simple and versatile input scenarios, providing easy parsing of different data types and making it beginner-friendly.