Vidyavardhini’s College of Engineering & Technology

Department of Computer Science And Engineering (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:

- **Using BufferedReader Class:**

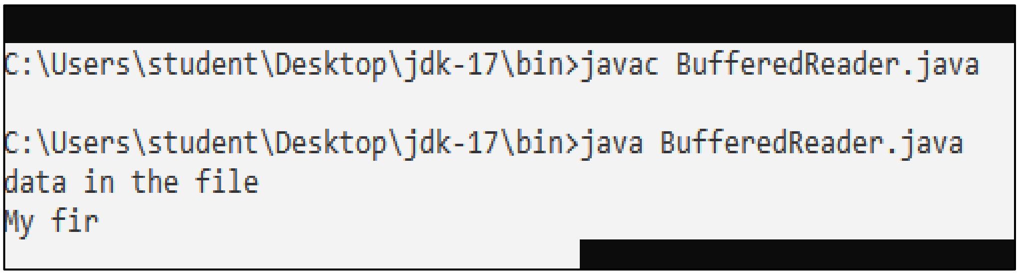
```
import java.io. FileReader;

import java. io. BufferedReader;

class BufferReader

{

public static void main(String args[])

{

char[] array=new char[6];

try

{

FileReader File=new FileReader("input.txt");

BufferedReader input = new BufferedReader(File);

input.read(array);

System.out.println("data in the file");

System.out.println(array);

input.close();

}

catch(Exception e)

{

e.getStackTrace();

}

}
}
```

- **Using Scanner Class:**

```
import java .util.Scanner;

class ScannerExample

{

public static void main(String args[])

{

Scanner in = new Scanner(System.in);

System.out.print("Enter Your Name:");

String name=in.nextLine();

System.out.println("Name is:" +name);

in.close();

}

}
```

Output:

- **Using BufferedReader Class:**

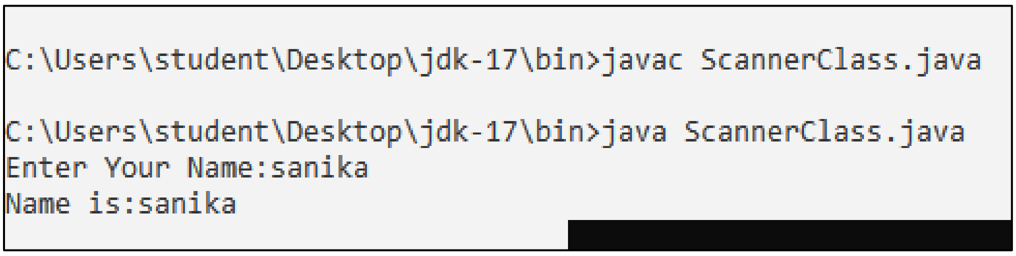


[redacted]

```
C:\Users\student\Desktop\jdk-17\bin>javac BufferedReader.java

C:\Users\student\Desktop\jdk-17\bin>java BufferedReader.java
data in the file
My fir
```

[redacted]

- **Using Scanner Class:**



```
C:\Users\student\Desktop\jdk-17\bin>javac ScannerClass.java

C:\Users\student\Desktop\jdk-17\bin>java ScannerClass.java
Enter Your Name:sanika
Name is:sanika
```

Conclusion:

I have used both the BufferedReader and Scanner classes to accept user input in Java, but I prefer to use the Scanner class whenever possible. The Scanner class is easier to use and provides more features than the BufferedReader class. For example, the Scanner class can be used to parse primitive data types, such as integers, floats, and doubles.

BufferedReader

- I have used the BufferedReader class to accept user input when I needed to read a large amount of data or when I needed to read input from a file.
- I have also used the BufferedReader class when I needed to control the buffer size or when I needed to synchronize the input stream.

Scanner

- I have used the Scanner class to accept user input in most other cases.
- The Scanner class is easier to use than the BufferedReader class, and it provides more features, such as parsing primitive data types.