Reading data from keyboard

There are many ways to read data from the keyboard. For example:

1. InputStreamReader
2. Console
3. Scanner
4. DataInputStream etc.

**InputStreamReader class**

InputStreamReader class can be used to read data from keyboard.It performs two tasks:

* connects to input stream of keyboard
* converts the byte-oriented stream into character-oriented stream

**BufferedReader class b**

BufferedReader class can be used to read data line by line by readLine() method.

**Example of reading data from keyboard by InputStreamReader and BufferdReader class**

In below example, we are connecting the BufferedReader stream with the InputStreamReader stream for reading the line by line data from the keyboard.

import java.io.\*;

class Test{

public static void main(String args[]){

try{

InputStreamReader r=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(r);

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

String name=br.readLine();

System.out.println("Welcome "+name);

}

catch(Exception e)

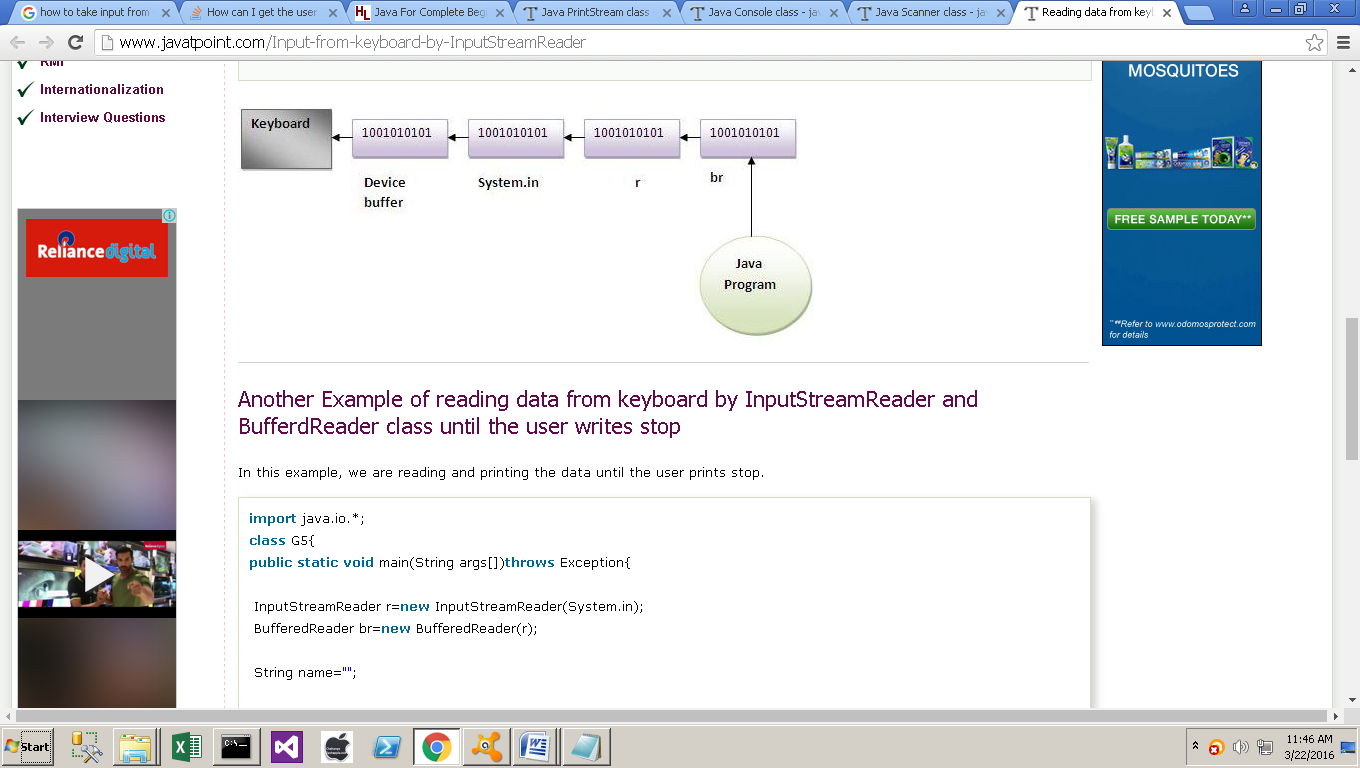
{ System.out.println(e);

}} }

Output:Enter your name

Amit

Welcome Amit



**Another Example of reading data from keyboard by InputStreamReader and BufferdReader class until the user writes stop**

In this example, we are reading and printing the data until the user prints stop.

import java.io.\*;

classTest{

public static void main(String args[])throws Exception{

try{

InputStreamReader r=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(r);

String name="";

while(!name.equals("stop")){

System.out.println("Enter data: ");

name=br.readLine();

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

}

br.close();

r.close();

}

catch(Exception e)

{ System.out.println(e);

}} }

Output:Enter data: Amit

data is: Amit

Enter data: 10

data is: 10

Enter data: stop

data is: stop

**Java Console class**

The Java Console class is be used to get input from console. It provides methods to read text and password.

If you read password using Console class, it will not be displayed to the user.

The java.io.Console class is attached with system console internally. The Console class is introduced since 1.5.

Let's see a simple example to read text from console.

String text=System.console().readLine();

System.out.println("Text is: "+text);

Methods of Console class

**Let's see the commonly used methods of Console class.**

|  |  |
| --- | --- |
| **Method** | **Description** |
| 1) public String readLine() | is used to read a single line of text from the console. |
| 2) public String readLine(String fmt,Object... args) | it provides a formatted prompt then reads the single line of text from the console. |
| 3) public char[] readPassword() | is used to read password that is not being displayed on the console. |
| 4) public char[] readPassword(String fmt,Object... args) | it provides a formatted prompt then reads the password that is not being displayed on the console. |

**How to get the object of Console**

System class provides a static method console() that returns the unique instance of Console class.

public static Console console(){}

Let's see the code to get the instance of Console class.

Console c=System.console();

**Java Console Example**

import java.io.\*;

classReadStringTest{

public static void main(String args[]){

Console c=System.console();

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

String n=c.readLine();

System.out.println("Welcome "+n);

} }

**Output:**

Enter your name: james gosling

Welcome james gosling

**Java Console Example to read password**

import java.io.\*;

classReadPasswordTest{

public static void main(String args[]){

Console c=System.console();

System.out.println("Enter password: ");

char[] ch=c.readPassword();

String pass=String.valueOf(ch);//converting char array into string

System.out.println("Password is: "+pass);

} }

Output:

Enter password:

Password is: sonoo

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Java Scanner class**

There are various ways to read input from the keyboard, the java.util.Scanner class is one of them.

The Java Scanner class breaks the input into tokens using a delimiter that is whitespace bydefault. It provides many methods to read and parse various primitive values.

Java Scanner class is widely used to parse text for string and primitive types using regular expression.

Java Scanner class extends Object class and implements Iterator and Closeable interfaces.

**Commonly used methods of Scanner class**

List of commonly used Scanner class methods:

|  |  |
| --- | --- |
| **Method** | **Description** |
| public String next() | it returns the next token from the scanner. |
| public String nextLine() | it moves the scanner position to the next line and returns the value as a string. |
| public byte nextByte() | it scans the next token as a byte. |
| public short nextShort() | it scans the next token as a short value. |
| public intnextInt() | it scans the next token as an int value. |
| public long nextLong() | it scans the next token as a long value. |
| public float nextFloat() | it scans the next token as a float value. |
| public double nextDouble() | it scans the next token as a double value. |

**Java Scanner Example to get input from console**

Simple example of the Java Scanner class which reads the int, string and double value as an input:

import java.util.Scanner;

classScannerTest{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("Enter your rollno");

introllno=sc.nextInt();

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

String name=sc.next();

System.out.println("Enter your fee");

double fee=sc.nextDouble();

System.out.println(" Hello: "+name+" your Rollno: "+rollno+" and fee: "+fee);

sc.close();

} }

Output:

Enter your rollno

111

Enter your name

Ratan

Enter

450000

Rollno:111name:Ratan fee:450000

**Java Scanner Example with delimiter**

**Example of Scanner class with delimiter. The \s represents whitespace**.

importjava.util.\*;

public class ScannerTest2{

public static void main(String args[]){

String input = "10 tea 20 coffee 30 tea buiscuits";

Scanner s = new Scanner(input).useDelimiter("\\s");

System.out.println(s.nextInt());

System.out.println(s.next());

System.out.println(s.nextInt());

System.out.println(s.next());

s.close();

}}

Output:

10

tea

20

coffee

**java.io.PrintStream class:**

The PrintStream class provides methods to write data to another stream. The PrintStream class automatically flushes the data so there is no need to call flush() method. Moreover, its methods don't throw IOException.

**Commonly used methods of PrintStream class:**

There are many methods in PrintStream class. Let's see commonly used methods of PrintStream class:

1. public void print(boolean b): it prints the specified boolean value.
2. public void print(char c): it prints the specified char value.
3. public void print(char[] c): it prints the specified character array values.
4. public void print(inti): it prints the specified int value.
5. public void print(long l): it prints the specified long value.
6. public void print(float f): it prints the specified float value.
7. public void print(double d): it prints the specified double value.
8. public void print(String s): it prints the specified string value.
9. public void print(Object obj): it prints the specified object value.
10. public void println(boolean b): it prints the specified boolean value and terminates the line.
11. public void println(char c): it prints the specified char value and terminates the line.
12. public void println(char[] c): it prints the specified character array values and terminates the line.
13. public void println(inti): it prints the specified int value and terminates the line.
14. public void println(long l): it prints the specified long value and terminates the line.
15. public void println(float f): it prints the specified float value and terminates the line.
16. public void println(double d): it prints the specified double value and terminates the line.
17. public void println(String s): it prints the specified string value and terminates the line./li>
18. public void println(Object obj): it prints the specified object value and terminates the line.
19. public void println(): it terminates the line only.
20. public void printf(Object format, Object... args): it writes the formatted string to the current stream.
21. public void printf(Locale l, Object format, Object... args): it writes the formatted string to the current stream.
22. public void format(Object format, Object... args): it writes the formatted string to the current stream using specified format.
23. public void format(Locale l, Object format, Object... args): it writes the formatted string to the current stream using specified format.

**Example of java.io.PrintStream class:**

In this example, we are simply printing integer and string values.

import java.io.\*;

classPrintStreamTest{

public static void main(String args[])throws Exception{

FileOutputStreamfout=new FileOutputStream("mfile.txt");

PrintStream pout=new PrintStream(fout);

pout.println(1900);

pout.println("Hello Java");

pout.println("Welcome to Java");

pout.close();

fout.close();

}}

**Example of printf() method of java.io.PrintStream class:**

Let's see the simple example of printing integer value by format specifier.

classPrintStreamTest{

public static void main(String args[]){

int a=10;

System.out.printf("%d",a);//Note, out is the object of PrintStream class

} }

Output:10