JAVA I/O
PART -II

Silicon University

Prepared by

Dr. Rajesh Kumar Ojha
Asst. Prof., CSE, Silicon University

BufferedOutputStream class

- Java BufferedOutputStream class uses an internal buffer to store data. It adds more efficiency than to write data directly into a stream.
- In this example, we are writing the textual information in the BufferedOutputStream object which is connected to the FileOutputStream object.
- The flush() flushes the data of one stream and send it into another. It is required if you have connected the one stream with another.

```
import java.io.*;
class Test{
 public static void main(String args[])throws Exception{
   FileOutputStream fout=new FileOutputStream("f1.txt");
   BufferedOutputStream bout=new BufferedOutputStream(fout);
   String s="Sachin is my favourite player";
   byte b[]=s.getBytes();
   bout.write(b);

   bout.flush();
   bout.close();
   fout.close();
   System.out.println("success");
 }
}
```

BufferedInputStream class

- Java BufferedInputStream class is used to read information from stream.
- It internally uses buffer mechanism to make the performance fast

```
import java.io.*;
class SimpleRead{
 public static void main(String args[]){
  try{
    FileInputStream fin=new FileInputStream("f1.txt");
    BufferedInputStream bin=new BufferedInputStream(fin);
    int i;
    while((i=bin.read())!=-1){
     System.out.println((char)i);
    }
    bin.close();
    fin.close();
  }catch(Exception e){system.out.println(e);}
 }
}
```

O/P: Sachin is my favourite player

FileWriter and FileReader

- Java FileWriter and FileReader classes are used to write and read data from text files. These are character-oriented classes, used for file handling in java.

Java FileWriter class

- Java FileWriter class is used to write character-oriented data to the file.

Constructors of FileWriter class

Constructor	Description
FileWriter(String file)	creates a new file. It gets file name in string.
FileWriter(File file)	creates a new file. It gets file name in File object.

Methods of FileWriter class

Method	Description
public void write(String text)	writes the string into FileWriter.
public void write(char c)	writes the char into FileWriter.
public void write(char[] c)	writes char array into FileWriter.
public void flush()	flushes the data of FileWriter.
public void close()	closes FileWriter.

Java FileWriter Example

```
import java.io.*;
class Simple{
 public static void main(String args[]){
  try{
   FileWriter fw=new FileWriter("abc.txt");
   fw.write("my name is sachin");
   fw.close();
  }catch(Exception e){System.out.println(e);}
  System.out.println("success");
 }
}
```

O/P: success…

Java FileReader class

- Java FileReader class is used to read data from the file. It returns data in byte format like FileInputStream class.

Constructors of FileWriter class

Constructor	Description
FileReader(String file)	It gets filename in string. It opens the given file in read mode. If file doesn't exist, it throws FileNotFoundException.
FileReader(File file)	It gets filename in file instance. It opens the given file in read mode. If file doesn't exist, it throws FileNotFoundException.

Methods of FileReader class

Method	Description
public int read()	returns a character in ASCII form. It returns -1 at the end of file.
public void close()	closes FileReader.

Java FileReader Example

```
import java.io.*;
class Simple{
 public static void main(String args[])throws Exception{
  FileReader fr=new FileReader("abc.txt");
  int i;
  while((i=fr.read())!=-1)
  System.out.println((char)i);

  fr.close();
 }
}
```

O/P: my name is sachin

CharArrayWriter class

- The CharArrayWriter class can be used to write data to multiple files. This class implements the Appendable interface.
- Its buffer automatically grows when data is written in this stream.
- Calling the close() method on this object has no effect.

CharArrayWriter class Example

```
import java.io.*;
class Simple{
 public static void main(String args[])throws Exception{
  CharArrayWriter out=new CharArrayWriter();
  out.write("my name is");
  FileWriter f1=new FileWriter("a.txt");
  FileWriter f2=new FileWriter("b.txt");
  FileWriter f3=new FileWriter("c.txt");
  FileWriter f4=new FileWriter("d.txt");
  out.writeTo(f1);
  out.writeTo(f2);
  out.writeTo(f3);
  out.writeTo(f4);
  f1.close();
  f2.close();
  f3.close();
  f4.close();
 }
}
```

Reading Data From Keyboard

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

- InputStreamReader
- Console
- Scanner
- 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

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

Reading data from keyboard

```
import java.io.*;
class G5{
public static void main(String args[])throws Exception{

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);
 }
}
```

```
Output:Enter your name
Amit
Welcome Amit
```

Reading data from keyboard

```
import java.io.*;
class G5{
public static void main(String args[])throws Exception{
 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();
 }
}
```

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.

Example:

- String text=System.console().readLine();
- System.out.println("Text is: "+text);

Methods of Console class

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

```
public static Console console(){}
```

Example:

```
Console c=System.console();
```

Console Example to read password

```
import java.io.*;
class ReadPasswordTest{
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
```

Scanner class

- 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.

Methods of Scanner class

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 int nextInt()	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.

Scanner Example

```
import java.util.Scanner;
class ScannerTest{
 public static void main(String args[]){
   Scanner sc=new Scanner(System.in);
   System.out.println("Enter your rollno");
   int rollno=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("Rollno:"+rollno+" name:"+name+" fee:"+fee);
   sc.close();
 }
}
```

Scanner Example

```
import java.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();
}}
```

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.

PrintStream Example

```
import java.io.*;
class PrintStreamTest{
 public static void main(String args[])throws Exception{
   FileOutputStream fout=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();
 }
}
```

Thank you