Dt : 6/11/2022(Sunday)

IO Streams and Files in Java:

IO Streams and Files in Java:

define Stream?

=>The contineous flow of data is known as Stream.

Types of streams in Java:

=>Streams in Java are categorized into two types:

1.Character Stream

2.Byte Stream

1.Character Stream:

=>The contineous flow of data in the form of 16-bits is known as Character Stream.

2.Byte Stream:

=>The contineous flow of data in the form of 8-bits is known as Byte Stream or Binary Stream.

---

define InputStream?

=>The Stream which is coming into Java application is known as InputStream

define OutputStream?

=>The Stream which is going out of Java application is known as OutputStream

Diagram:

----

define File Storage?

=>The smallest permanent Storage of Computer System which is controlled and managed by the OperatingSystem is known as File Storage.

Note:

=>when we want to establish communication b/w JavaProgram and File Storage, the JavaProgram must be constructed using Classes and Interfaces available from "java.io" package.

Diagram:

==========

**imp*

Classes related to Byte Stream:

(1)OutputStream:

(a)FileOutputStream

(b)ObjectOutputStream

(c)ByteArrayOutputStream

(d)PipedOutputStream

(e)FilteredOutputStream

(i)DataOutputStream

(ii)BufferedOutputStream

(iii)PrintStream

(2)InputStream:

(a)FileInputStream

(b)ObjectInputStream

(c)ByteArrayInputStream

(d)PipedInputStream

(e)FilteredInputStream

(i)DataInputStream

(ii)BufferedInputStream

(iii)PushBackInputStream

==========

Ex-program:

wap to copy the file from one loaction(Source) to another location(destination)?

Source path : C:\Images\Screenshot (1).png

Destination path : D:\Images\XYZ.png

Program : DemoFile1.java

```
package maccess;
import java.io.*;
```

```
import java.util.*;
public class DemoFile1 {
        public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    try(s;)
    {
        try
        {
        System.out.println("Enter fpath&fname:(Source)");
        String path1 = s.nextLine();
        FileInputStream fis = new FileInputStream(path1);//Con_Call
        System.out.println("Enter fPath&fName:(Destination)");
        String path2 = s.nextLine();
        FileOutputStream fos = new FileOutputStream(path2);
        int z;
        while((z=fis.read())!=-1)
        {
                fos.write(z);
        }//end of loop
        System.out.println("File copied Successfully...");
        fos.close();
        fis.close();
        }//end of try
        catch(Exception e)
```

```
            {
            e.printStackTrace();
            }
      }//end of try with resource
          }
}
```

o/p:

Enter fpath&fname:(Source)

C:\Images\Screenshot (1).png

Enter fPath&fName:(Destination)

D:\Images\XYZ.png

File copied Successfully...

==========

Define FileInputStream?

=>FileInputStream class will find the file and opens the file to read byte stream data.

syntax:

FileInputStream fis = new FileInputStream(path);

Define FileOutputStream?

=>FileOutputStream class will create a new file and opens the file to write byte stream data.

syntax:

FileOutputStream fos = new FileOutputStream(path);

==========

**imp*

Object Stream:

=>when we want to store Object onto file Storage or Database Storage or Network,the Object must be available in the form Stream.(Byte Stream)

define Serialization process?

=>The process of converting Object into Stream is known as Serialization process.

=>we use writeObject() method from "java.io.ObjectOutStream" class to perform Serialization process.

syntax:

ObjectOutputStream oos = new ObjectOutputStream(fos);

oos.writeObject(obj_name);

define DeSerialization process?

=>The process of converting Stream into Object is known as DeSerialization process.

=>we use readObject() method from "java.io.ObjectInputStream" class to perform DeSerialization process.

syntax:

ObjectInputStream ois = new ObjectInputStream(fis);

Object o = ois.readObject();

Note:

=>To Perform Serialization and DeSerialization process,the class must be implemented from "java.io.Serializable" interface.

=>"java.io.Serializable" interface is empty interface and which is also known as Marker interface or Tagging interface.

Ex-program:

TransLog.java

```
package test;
import java.io.Serializable;
import java.util.Date;
//Immutable Class
@SuppressWarnings("serial")
public final class TransLog implements Serializable
{
   private final long hAccNo,bAccNo;
   private final double amt;
   private final Date dateTime;
   public TransLog(long hAccNo,long bAccNo,double amt,Date dateTime)
```

```
  {
        this.hAccNo=hAccNo;
        this.bAccNo=bAccNo;
        this.amt=amt;
        this.dateTime=dateTime;
  }
        public final long gethAccNo() {
              return hAccNo;
        }
        public final long getbAccNo() {
              return bAccNo;
        }
        public final double getAmt() {
              return amt;
        }
        public final Date getDateTime() {
              return dateTime;
        }
}
```

Serialization.java(MainClass)

```
package maccess;
import java.io.*;
import java.util.*;
```

```
import test.TransLog;
public class Serialization {
        public static void main(String[] args) {
      Scanner s = new Scanner(System.in);
      try(s;)
      {
          try
          {
                  System.out.println("Enter the hAccNo:(Home)");
                  long hAccNo = s.nextLong();
                  System.out.println("Enter the bAccNo:(benefieciery)");
                  long bAccNo = s.nextLong();
                  System.out.println("Enter the amt to be transferred:");
                  double amt = s.nextDouble();
                  TransLog ob1 = new TransLog(hAccNo,bAccNo,amt,new Date());
                          //Immutable object
                  String path = "D:\\Images\\Obj.txt";
                  FileOutputStream fos = new FileOutputStream(path);
                  ObjectOutputStream oos = new ObjectOutputStream(fos);
                  oos.writeObject(ob1);//Serialization
                  System.out.println("Object Stored in file Successfully...");
                  oos.close();
                  fos.close();
          }//end of try
```

```
            catch(Exception e)
            {
                    e.printStackTrace();
            }
        }//end of try
            }
}
```

o/p:

```
Enter the hAccNo:(Home)
6123456
Enter the bAccNo:(benefieciery)
313131
Enter the amt to be transferred:
6000
Object Stored in file Successfully...
```

DeSerialization.java(MainClass)

```
package maccess;
import java.io.*;
import test.TransLog;
public class DeSerialization {
        public static void main(String[] args) {
    try
    {
```

```
        String path = "D:\\Images\\Obj.txt";
        FileInputStream fis = new FileInputStream(path);
        ObjectInputStream ois = new ObjectInputStream(fis);
        TransLog ob2 = (TransLog)ois.readObject();//DeSerialization process
        System.out.println("====Transaction details====");
        System.out.println("HAccNo:"+ob2.gethAccNo());
        System.out.println("BAccNo:"+ob2.getbAccNo());
        System.out.println("Amt:"+ob2.getAmt());
        System.out.println("DataTime:"+ob2.getDateTime());
        ois.close();
        fis.close();
    }//end of try
    catch(Exception e)
    {
        e.printStackTrace();
    }
      }
}
```

o/p:

```
====Transaction details====
HAccNo:6123456
BAccNo:313131
Amt:6000.0
DataTime:Sun Nov 06 11:41:01 PST 2022
```

==========