File Operations in Java

Stream

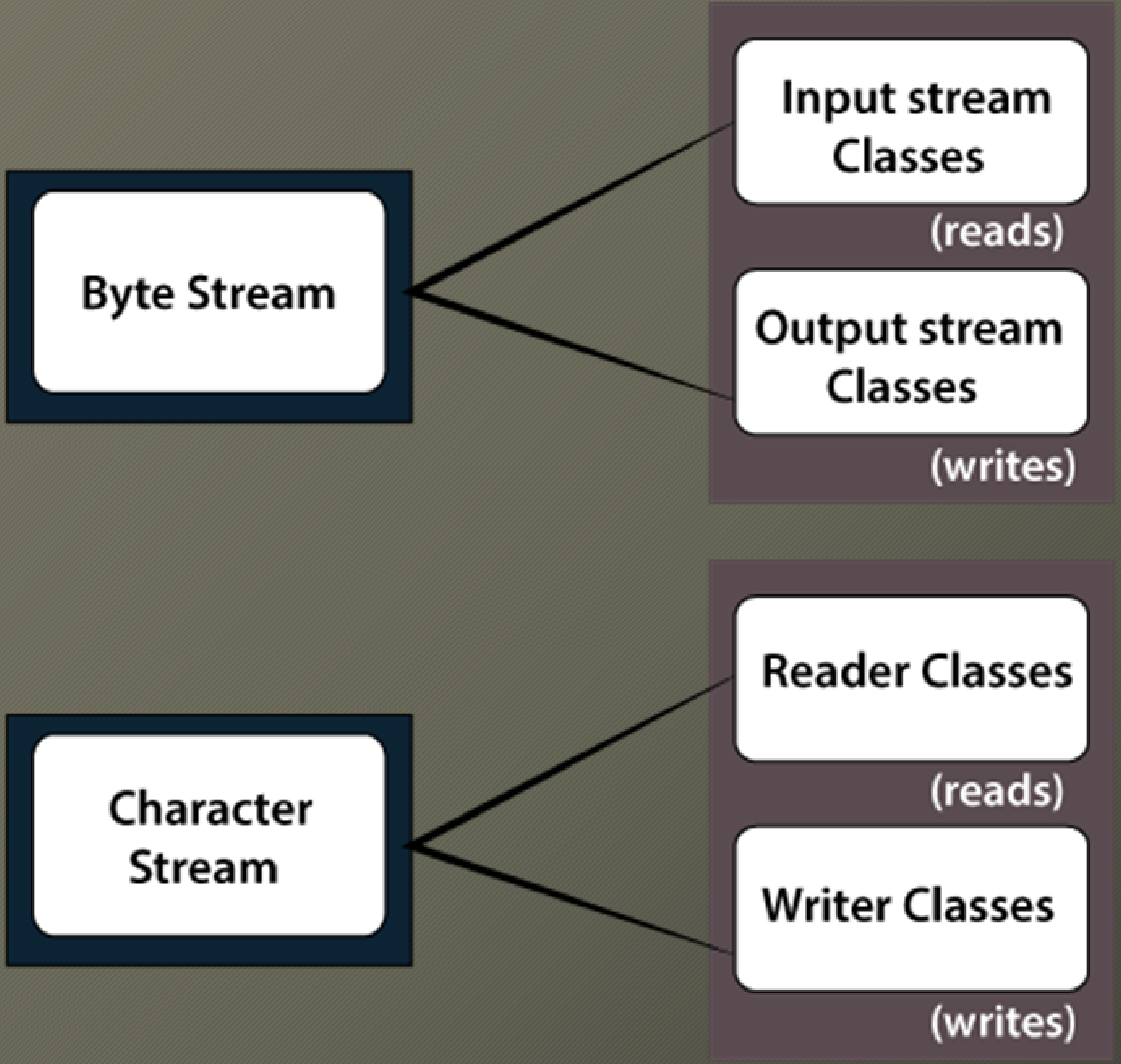
- A series of data is referred to as **a stream**. In Java, **Stream** is classified into two types, i.e., **Byte Stream** and **Character Stream**.

Byte Stream

- **Byte Stream** is mainly involved with byte data. A file handling process with a byte stream is a process in which an input is provided and executed with the byte data.

Character Stream

- **Character Stream** is mainly involved with character data. A file handling process with a character stream is a process in which an input is provided and executed with the character data.



Brief classification of I/O streams

File Operations

01 Create a File

02 Get File Information

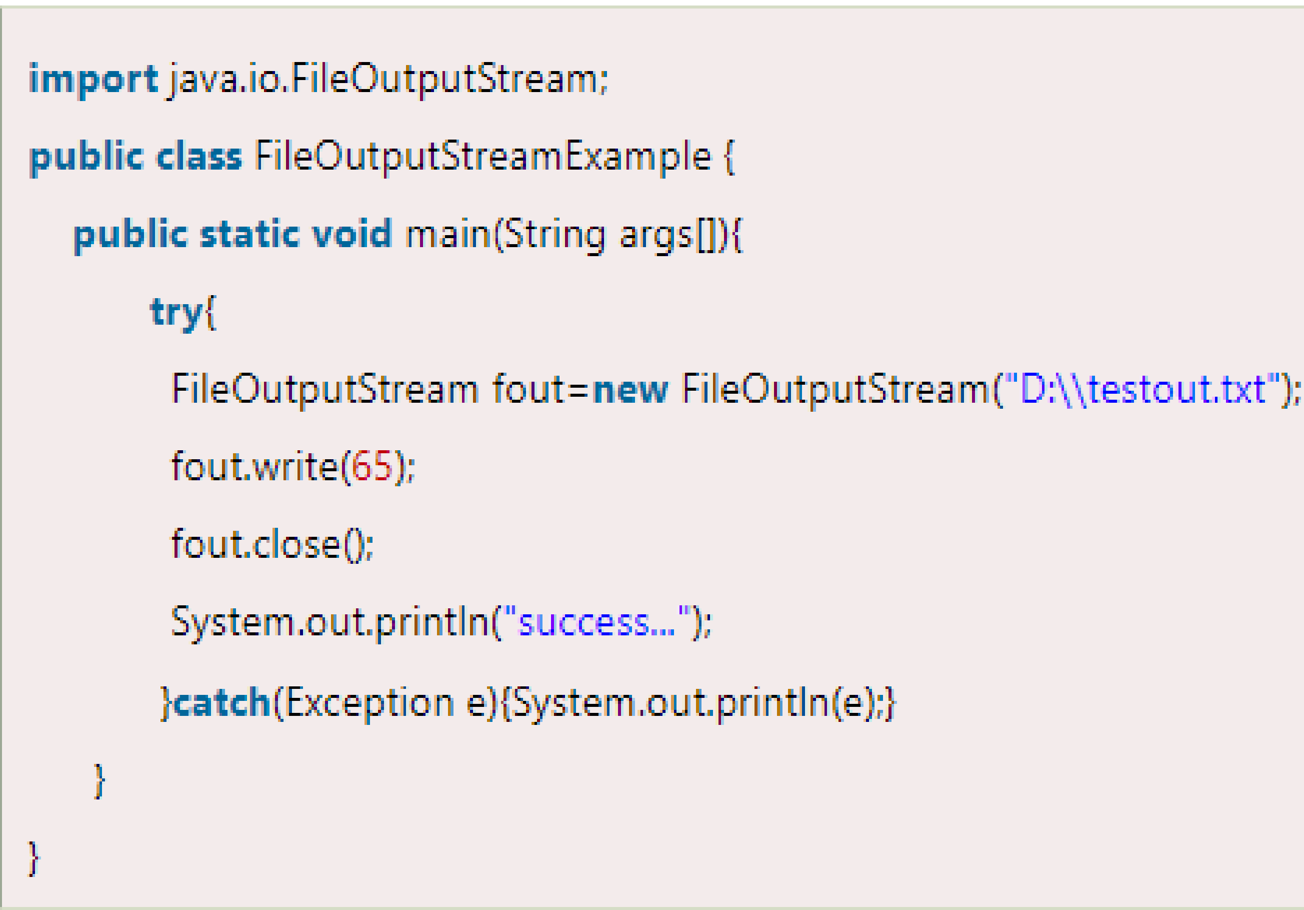
03 Write to a File

04 Read From a File

05 Delete a File

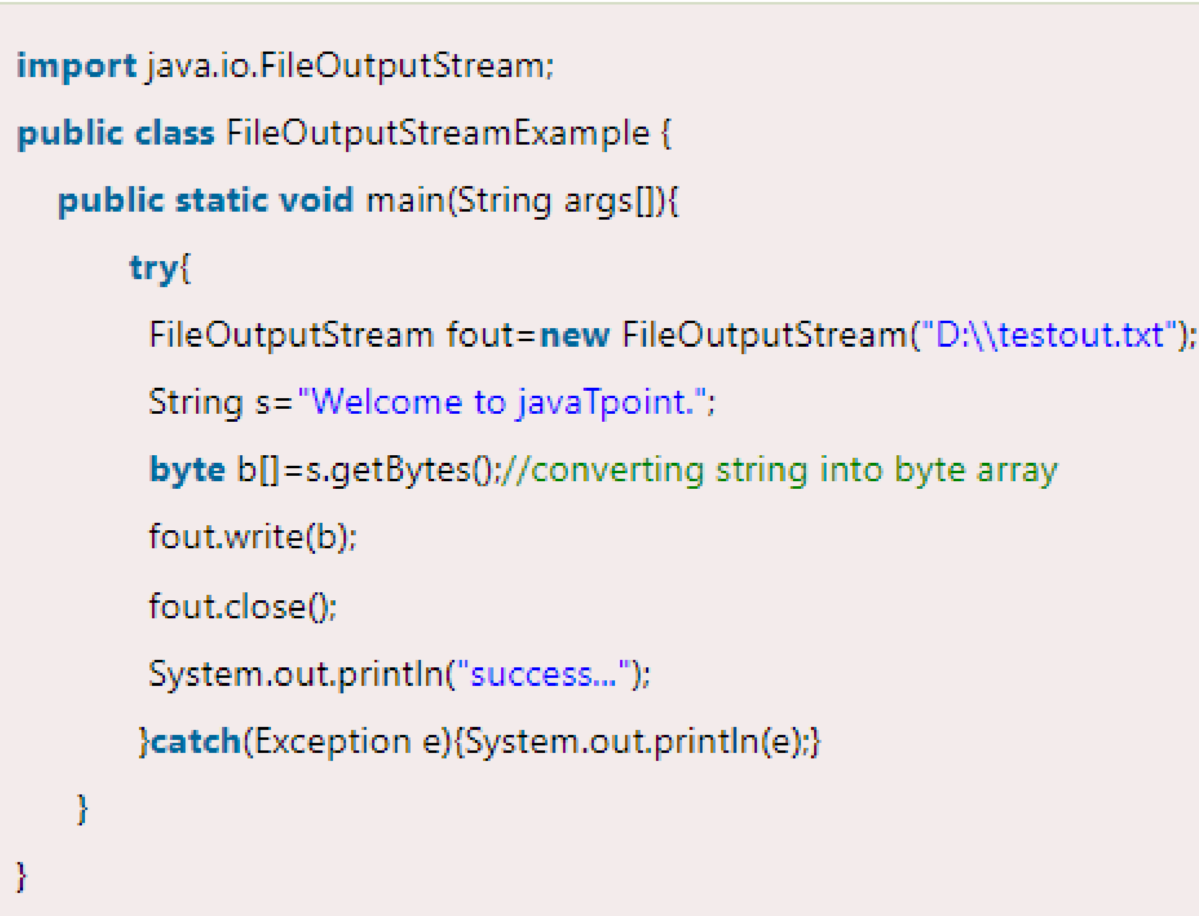
Java FileOutputStream (Byte Stream)

- write byte



```
import java.io.FileOutputStream;
public class FileOutputStreamExample {
  public static void main(String args[]){
        try{
          FileOutputStream fout=new FileOutputStream("D:\\testout.txt");
          fout.write(65);
          fout.close();
          System.out.println("success...");
         }catch(Exception e){System.out.println(e);}
    }
}
```

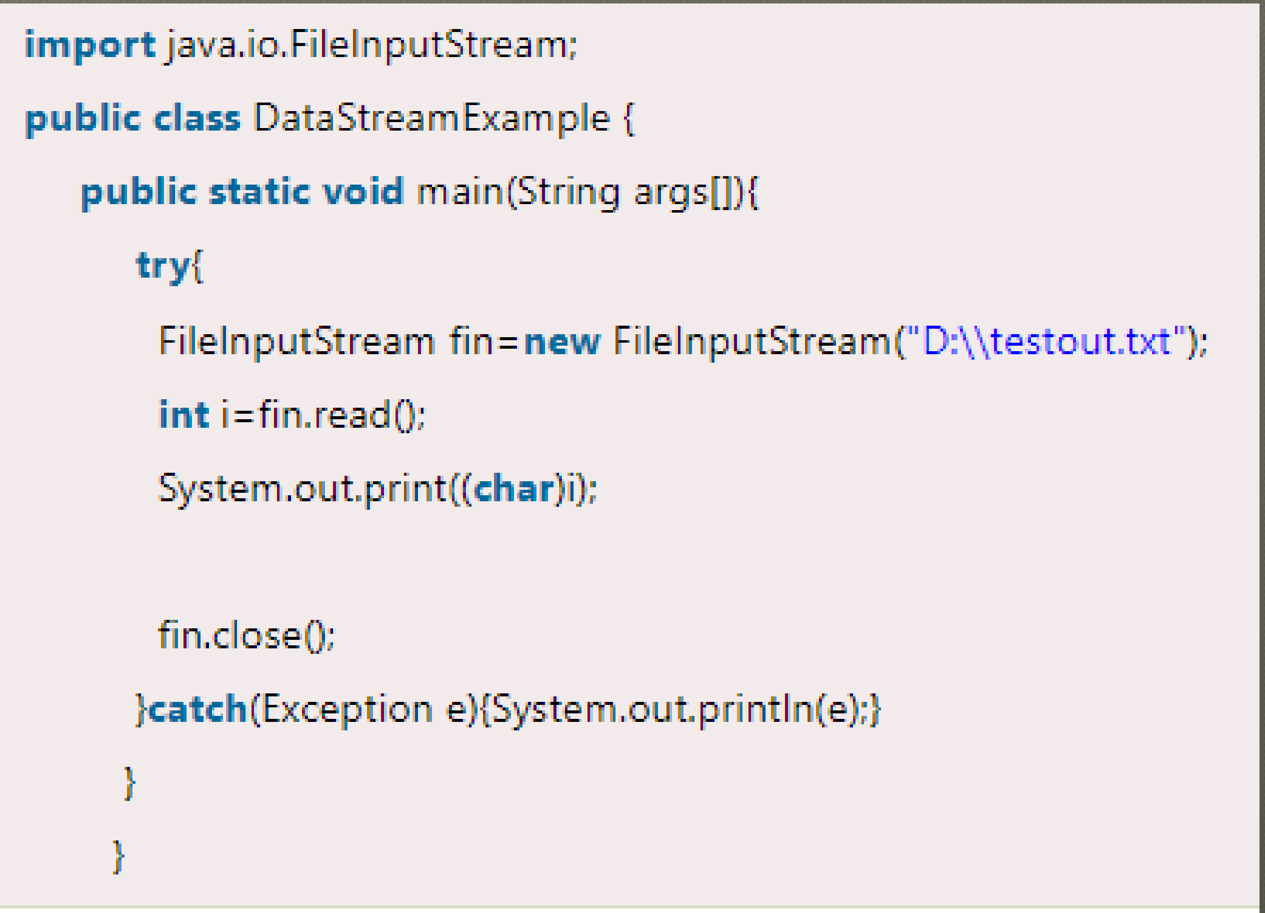
- write string



```
import java.io.FileOutputStream;
public class FileOutputStreamExample {
  public static void main(String args[]){
       try{
        FileOutputStream fout=new FileOutputStream("D:\\testout.txt");
        String s="Welcome to javaTpoint.";
        byte b[]=s.getBytes();//converting string into byte array
        fout.write(b);
        fout.close();
        System.out.println("success...");
       }catch(Exception e){System.out.println(e);}
    }
}
```

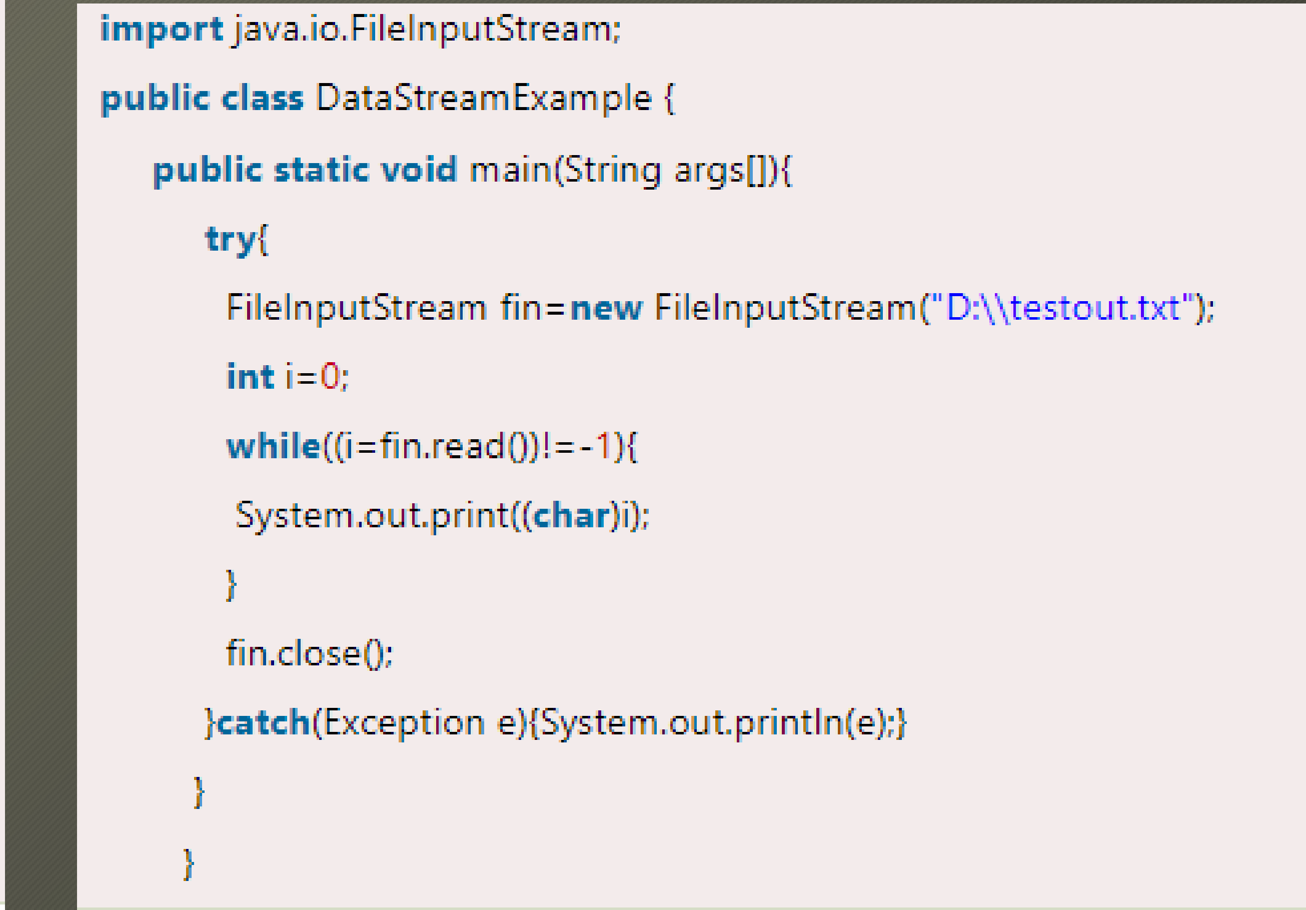
Java FileInputStream (Byte Stream)

- read single character



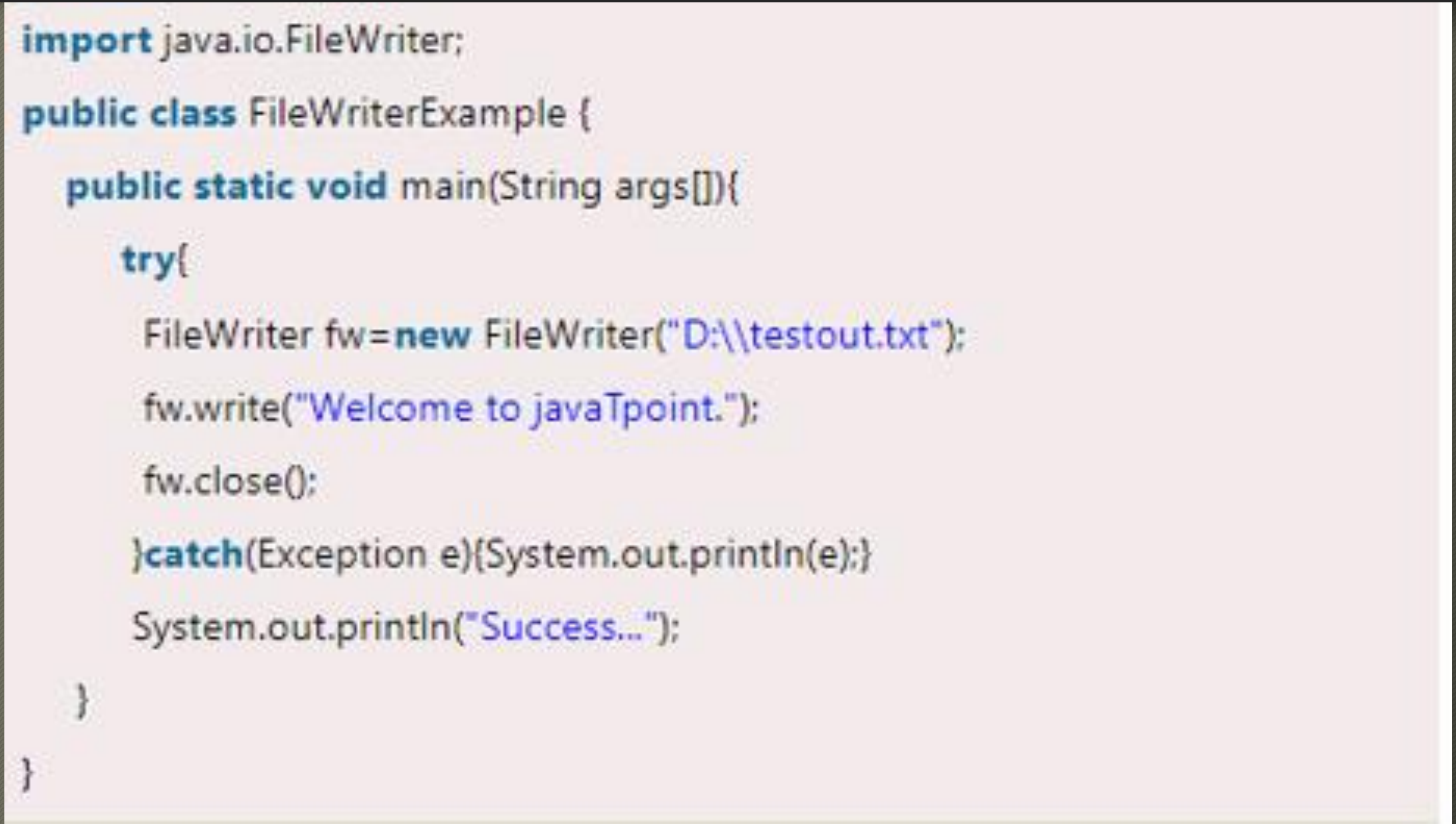
```
import java.io.FileInputStream;
public class DataStreamExample {
    public static void main(String args[]){
        try{
          FileInputStream fin=new FileInputStream("D:\\testout.txt");
          int i=fin.read();
          System.out.print((char)i);

          fin.close();
        }catch(Exception e){System.out.println(e);}
       }
      }
```

- read all characters



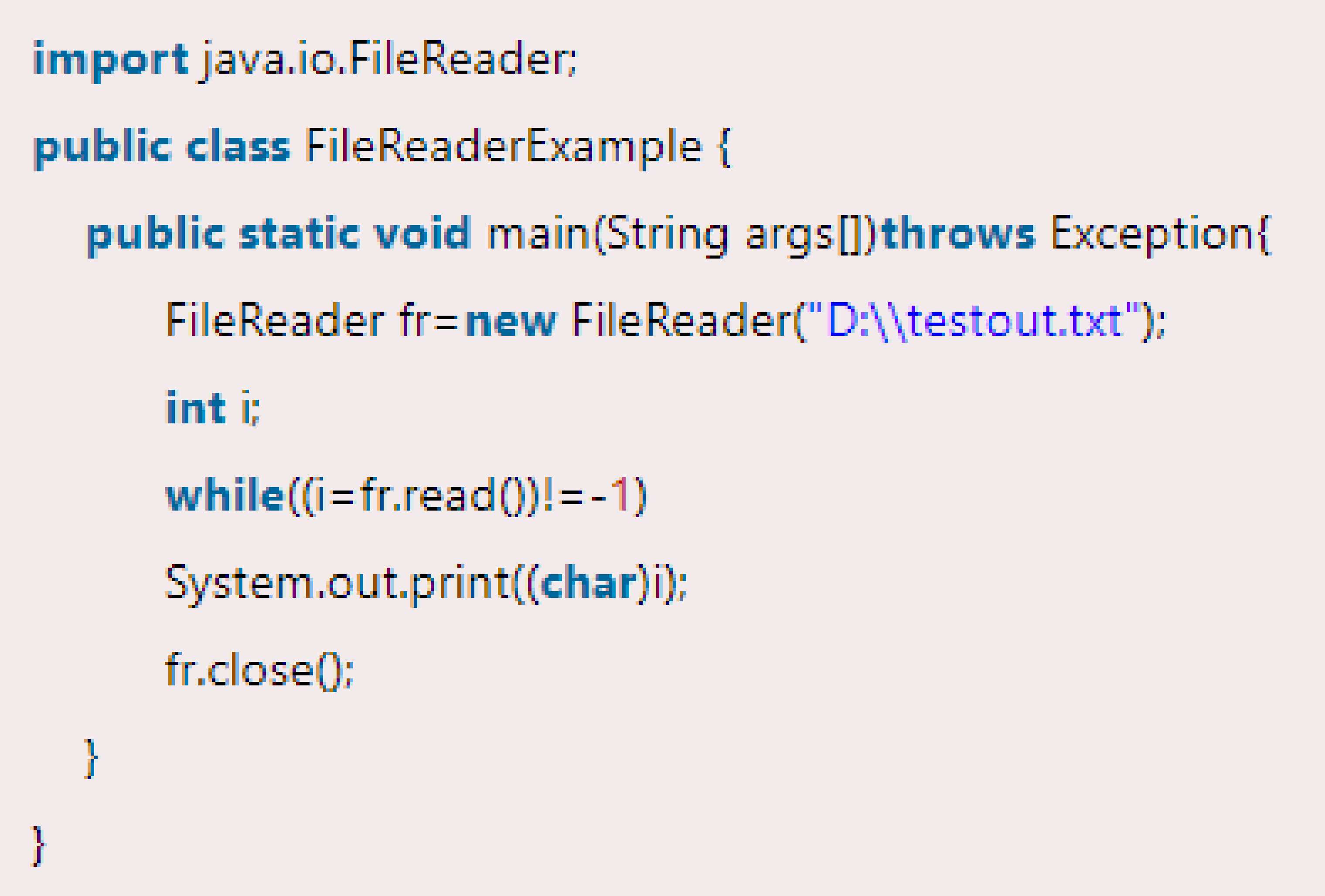
```
import java.io.FileInputStream;
public class DataStreamExample {
    public static void main(String args[]){
        try{
          FileInputStream fin=new FileInputStream("D:\\testout.txt");
          int i=0;
          while((i=fin.read())!=-1){
           System.out.print((char)i);
          }
          fin.close();
        }catch(Exception e){System.out.println(e);}
       }
      }
```

Java FileWriter (Character Stream)



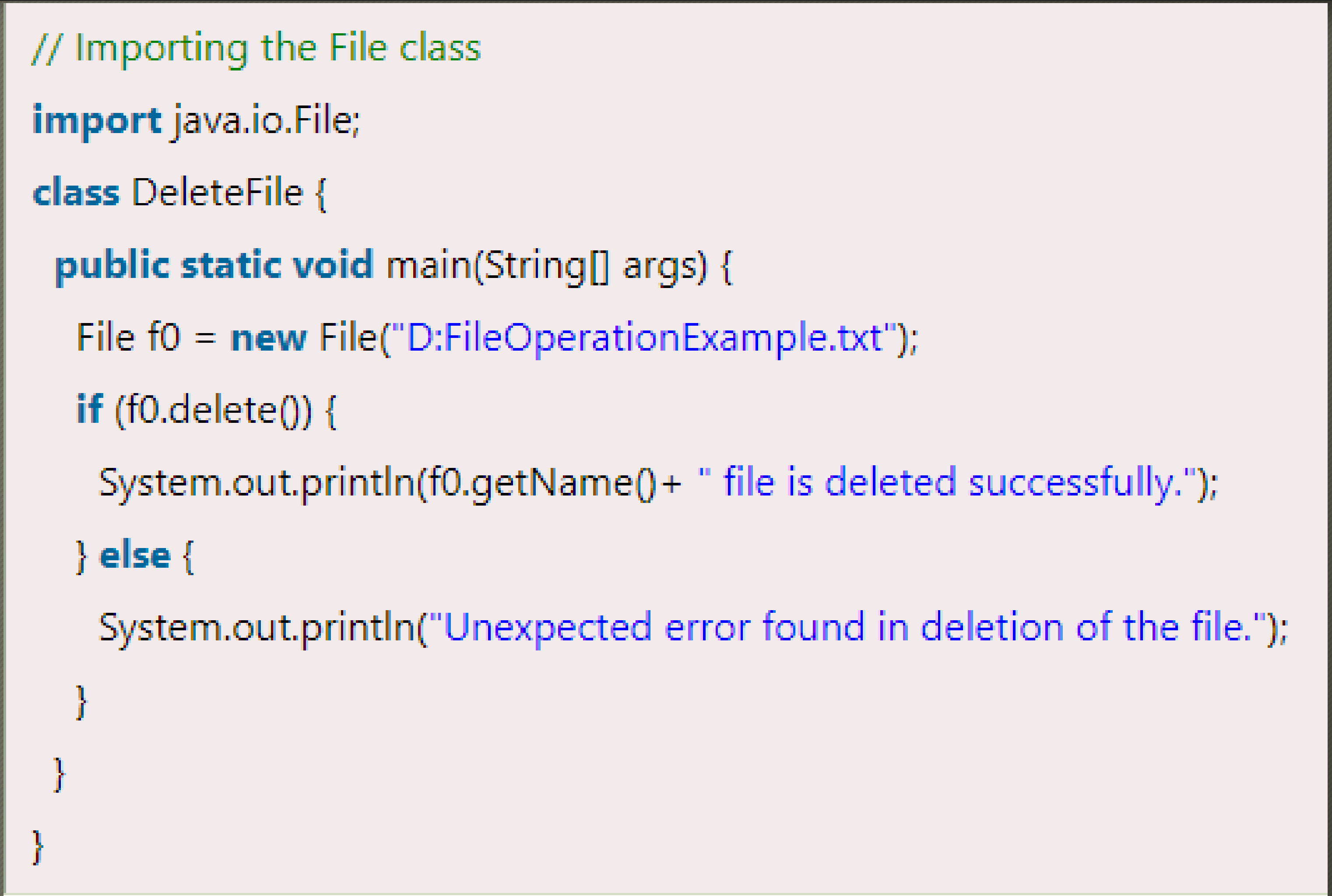
```
import java.io.FileWriter;
public class FileWriterExample {
  public static void main(String args[]){
       try{
         FileWriter fw=new FileWriter("D:\\testout.txt");
         fw.write("Welcome to javaTpoint.");
         fw.close();
        }catch(Exception e){System.out.println(e);}
        System.out.println("Success...");
   }
}
```

Java FileReader (Character Stream)



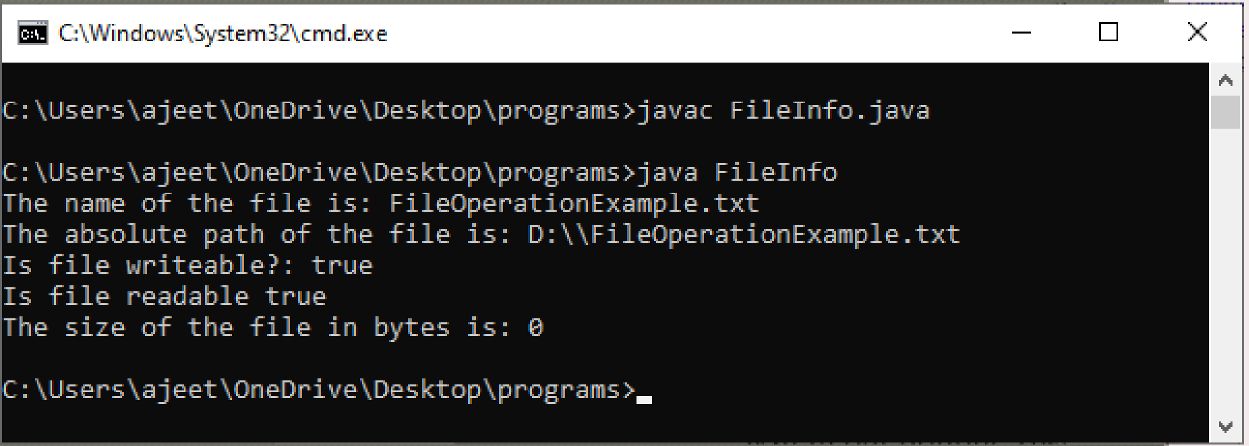
```
import java.io.FileReader;
public class FileReaderExample {
    public static void main(String args[])throws Exception{
          FileReader fr=new FileReader("D:\\testout.txt");
          int i;
          while((i=fr.read())!=-1)
          System.out.print((char)i);
          fr.close();
    }
}
```

Delete File



```java
// Importing the File class
import java.io.File;
class DeleteFile {
 public static void main(String[] args) {
  File f0 = new File("D:FileOperationExample.txt");
  if (f0.delete()) {
   System.out.println(f0.getName()+ " file is deleted successfully.");
  } else {
   System.out.println("Unexpected error found in deletion of the file.");
  }
 }
}
```

Get File Information



```
C:\Users\ajeet\OneDrive\Desktop\programs>javac FileInfo.java

C:\Users\ajeet\OneDrive\Desktop\programs>java FileInfo
The name of the file is: FileOperationExample.txt
The absolute path of the file is: D:\\FileOperationExample.txt
Is file writeable?: true
Is file readable true
The size of the file in bytes is: 0

C:\Users\ajeet\OneDrive\Desktop\programs>
```

```java
import java.io.File;
class FileInfo {
  public static void main(String[] args) {
    // Creating file object
    File f0 = new File("D:FileOperationExample.txt");
    if (f0.exists()) {
      // Getting file name
      System.out.println("The name of the file is: " + f0.getName());

      // Getting path of the file
      System.out.println("The absolute path of the file is: " + f0.getAbsolutePath());

      // Checking whether the file is writable or not
      System.out.println("Is file writeable?: " + f0.canWrite());

      // Checking whether the file is readable or not
      System.out.println("Is file readable " + f0.canRead());

      // Getting the length of the file in bytes
      System.out.println("The size of the file in bytes is: " + f0.length());
    } else {
      System.out.println("The file does not exist.");
    }
  }
}
```

1. Program to list the sub directories and files in a given directory and also search for a file name.

File Name : p1.java

```java
import java.io.File;
import java.io.*;
import java.util.*;

public class p1 {
    public static final String RESET = "\033[0m";
    public static final String RED = "\033[0;31m";
    public static final String TEXT_RESET = "\u001B[0m";
    public static final String TEXT_BLACK = "\u001B[30m";
    public static final String TEXT_RED = "\u001B[31m";

    static void RecursivePrint(File[] arr, int index, int level, String searchfor) {
        // exit condition
        if (index == arr.length)
            return;

        // space for internbal level
        for (int i = 0; i < level; i++)
            System.out.print("\t");

        if (arr[index].getName().toLowerCase().contains(searchfor))
            System.out.print(TEXT_RED);
        else
            System.out.print(RESET);
        // for files
        if (arr[index].isFile())
            System.out.println(arr[index].getName());

        else if (arr[index].isDirectory()) {
            System.out.println("[" + arr[index].getName() + "]");

            RecursivePrint(arr[index].listFiles(), 0, level + 1, searchfor);
        }
        RecursivePrint(arr, ++index, level, searchfor);
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the directory path");
        String maindirpath = scan.nextLine();
        System.out.println("Enter the file/directory name to search");
        String searchfor = scan.nextLine();
        File maindir = new File(maindirpath);
        if (maindir.exists() && maindir.isDirectory()) {
            File arr[] = maindir.listFiles();
            System.out.println("#################################################");
            System.out.println("Files from main directory" + maindir);
            System.out.println("#################################################");
            RecursivePrint(arr, 0, 0, searchfor.toLowerCase()); // array,index,level,search
        }
    }
}
```

2. **Write a program to write to a file, then read from the file and display the contents on the console.**
   File Name : read.java

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;

class read {
    public static void main(String[] args) {
        // initialize String
        String var = "";
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the text to create file : type ENTER key 3 times to stop");
        while (!var.endsWith("\n\n\n"))
            var = var + scan.nextLine() + "\n";
        try {
            // create file object
            File file = new File("output.txt");
            // create filewriter object
            FileWriter fw = new FileWriter(file);
            fw.write(var);
            fw.close();
            System.out.println("Reading File content");
            FileReader fr = new FileReader("output.txt");
            String str = "";
            int i;
            while ((i = fr.read()) != -1) {
                // Storing every character in the string
                str += (char) i;
            }
            System.out.println(str);
            fr.close();
        } catch (IOException e) {
            System.out.println("There are some exception");
        }
    }
}
```

3. **Write a program to copy one file to another.**
   File Name : copy.java

```java
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;

public class copy {
    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        System.out.println("Enter the source File Name");
        String source=scan.nextLine();
        try {
            FileReader fr=new FileReader(source);
            String str = "";
            int i;
            System.out.println("Reading from file "+source);
            while ((i = fr.read()) != -1) {
                // Storing every character in the string
                str += (char) i;
            }
            System.out.println(str);
            System.out.println("\n Enter the filename to copy");
            String destination=scan.nextLine();
            File file=new File(destination);
            FileWriter fw = new FileWriter(file);
            fw.write(str);
            fr.close();
            fw.close();
            System.out.println("Copied from "+source+" to "+destination+ " Successfully..!");
        } catch (Exception e) {
            //TODO: handle exception
            System.out.println("Exception Occured");
        }
    }
}
```

4. **Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.**
   File Name : oddeven.java

```java
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.*;
import java.util.*;
import java.io.File;

public class oddeven {
    public static void main(String[] args) {
        try {
            FileReader fr = new FileReader("numbers.txt");
            BufferedReader br = new BufferedReader(fr);

            File file1 = new File("oddnumbers.txt");
            FileWriter fw1 = new FileWriter(file1);
            File file2 = new File("evennumbers.txt");
            FileWriter fw2 = new FileWriter(file2);
            String num;
            while ((num = br.readLine()) != null) {
                if (Integer.parseInt(num) % 2 == 0) {
                    fw2.write(num + "\n");
                } else {
                    fw1.write(num + "\n");
                }

            }
            fw1.close();
            fw2.close();
        } catch (Exception e) {
            // TODO: handle exception
            System.out.println("Error");
        }
    }

}
```