

File Input and Output

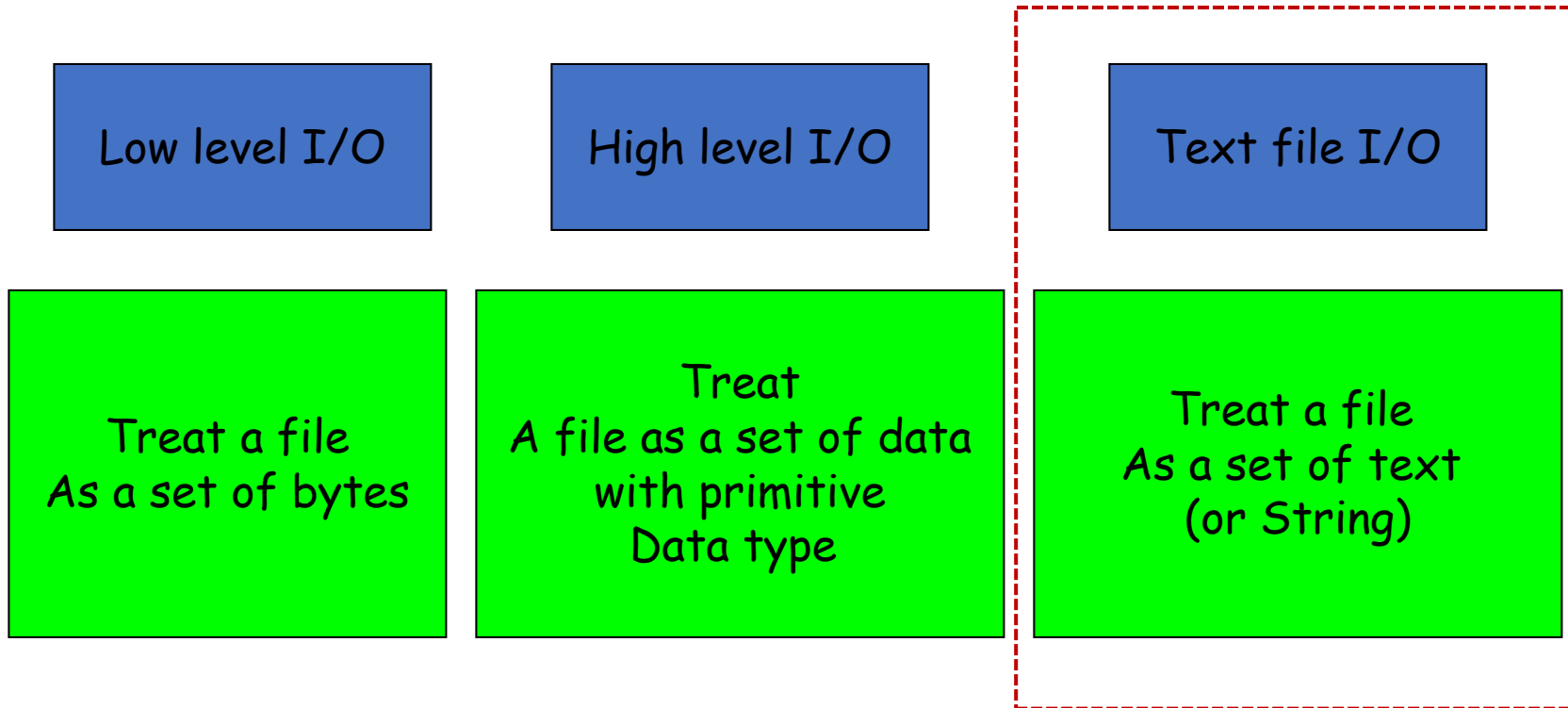
Sungchul Lee

Learning Object

- Built-in Package for Text File Input and Output
 - Java.IO package
 - Write
 - ❖ FileOutputStream and PrintWriter
 - Read
 - ❖ FileReader and BufferedReader

Type of File I/O

➤ Three type of file I/O



Built-in Package for Text File I/O

- java.io package (`import java.io.*;`)
- Instead of storing primitive data values as binary data in a file, we can convert and store them as a **string** data
 - ❑ View the file content using any text editor
- Write data on File
 - ❑ **FileOutputStream** is used to handle raw binary data.
 - ❑ **PrintWriter** is used to send characters to a text file.
- Read data from File
 - ❑ java.io.BufferedReader and java.io.FileReader class
 - ❑ **FileReader** is used to read data from the file
 - ❑ **BufferedReader** is used to read the text from a character-based input stream.

How to Write Data to a Text File

Step 1: Create a File object

- ❑ File class with file name

Step 2: Create a FileOutputStream object

- ❑ File object and FileOutputStream class

Step 3: Create a PrintWriter object

- ❑ FileOutputStream Object and PrintWrite class

Step 4: Write line(s)

Step 5: Close the file

FileOutputStream and PrintWriter

➤ Syntax for FileOutputStream Object:

```
FileOutputStream <variable_name> = new  
    FileOutputStream(<name of a File object>);
```

➤ Syntax for PrintWriter Object:

```
PrintWriter <variable_name> = new  
    PrintWriter(<name of a FileOutputStream object>);
```

➤ PrintWriter Method for writing:

```
<A print writer object>.println(<string object name>);
```

➤ <https://docs.oracle.com/javase/8/docs/api/java/io/PrintWriter.html>

How to Read Data from a Text File

Step 1: Create a File object

- ❑ File class and file name

Step 2: Create a [FileReader](#) object

- ❑ File object and FileReader class

Step 3: Create a [BufferedReader](#) object

- ❑ FileReader Object and BufferedReader class

Step 4: Read **line by line**

Step 5: Convert String object to primitive data type as necessary

Step 6: Close the file

FileReader and BufferedReader

- Syntax for FileReader Object:
`FileReader <variable_name> = new FileReader(<name of a File object>);`
- Syntax for BufferedReader object:
`BufferedReader <variable_name> = new
BufferedReader(<name of a FileReader object>);`
- BufferedReader method for reading
`<bufferedReader object name>.readLine();`
- <https://docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html>

Practice

1. Make a new project (Reference: Create Project and Class File)
 - ❑ Project name: Text_IO
2. Create a new Class File
 - ❑ Class name: Main
3. Coding:
 - ❑ `import java.io.*;`
 - ❖ File
 - ❖ `FileOutputStream` and `PrintWriter`
 - ❖ `FileReader` and `BufferedReader`

Practice – code (Write)

```
import java.io.*;
public class Main {
    public static void main(String[] args) {
        File file = new File("output.dat");//File Object
        //create FileOutputStream using file object
        FileOutputStream fileStream = new FileOutputStream(file);
        //create printWriter object using FileOutputStream
        PrintWriter printWriter = new PrintWriter(fileStream);
        int number[] = new int[10];
        for (int i=0;i<number.length; i++) {
            number[i] = i+1;
            printWriter.println("Line Number:"+ number[i]);
        }
```

```
//output done,
printWriter.flush();
printWriter.close();
```

Practice – code (Read)

```
//create FileReader using file object
FileReader fileReader = new FileReader(file);
//create BufferedReader object using FileReader
BufferedReader bufferReader = new BufferedReader(fileReader);
String inputStr = "";
inputStr = bufferReader.readLine(); //read line
while (inputStr != null){
    System.out.println(inputStr);
    inputStr = bufferReader.readLine(); //read line
} // end of while
bufferReader.close();
}
```

Practice – Code

```
Main.java
1 import java.io.*;
2 public class Main {
3     public static void main(String[] args) throws IOException {
4         // TODO Auto-generated method stub
5         //create file object
6         File file = new File("output.dat");
7
8         //create FileOutputStream using file object
9         FileOutputStream fileStream = new FileOutputStream(file);
10
11        //create PrintWriter object using FileOutputStream
12        PrintWriter printWriter = new PrintWriter(fileStream);
13
14        int number[] = new int[10];
15        for (int i=0;i<number.length; i++) {
16            number[i] = i+1;
17            printWriter.println("Line Number:"+ number[i]);
18        }
19        printWriter.flush();
20        printWriter.close();
21
22        //create FileReader using file object
23        FileReader fileReader = new FileReader(file);
24
25        //create BufferedReader object using FileReader
26        BufferedReader bufferReader = new BufferedReader(fileReader);
27        String inputStr = "";
28        //read line
29        inputStr = bufferReader.readLine();
30        while (inputStr != null){
31            System.out.println(inputStr);
32            inputStr = bufferReader.readLine();
33
34        } // end of while
35        bufferReader.close();
36    }
37 }
```

Practice –Result

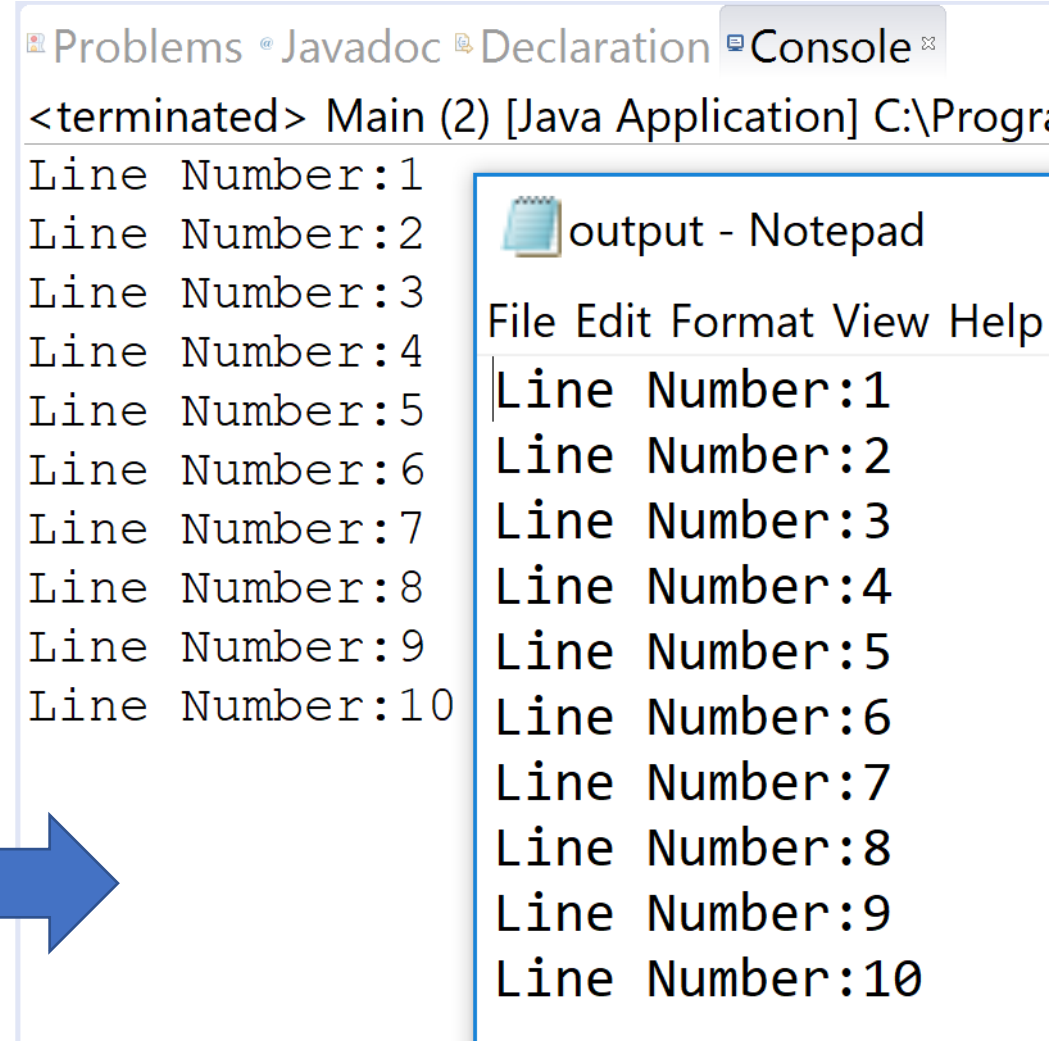
➤output.txt

- ☐ In Project folder

- ☐ String data type

- ❖ Any text editor can read

Result



The screenshot shows an IDE interface with a console window and a Notepad window. The console window displays the output of a Java application, showing 10 lines of text. The Notepad window, titled 'output - Notepad', contains the same 10 lines of text.

```
<terminated> Main (2) [Java Application] C:\Progra
Line Number:1
Line Number:2
Line Number:3
Line Number:4
Line Number:5
Line Number:6
Line Number:7
Line Number:8
Line Number:9
Line Number:10
```

output - Notepad

File Edit Format View Help

```
Line Number:1
Line Number:2
Line Number:3
Line Number:4
Line Number:5
Line Number:6
Line Number:7
Line Number:8
Line Number:9
Line Number:10
```

Summary

➤ Write

□ FileOutputStream and PrintWriter

➤ Read

□ FileReader and BufferedReader

```
File file = new File("output.dat");

//create FileOutputStream using file object
FileOutputStream fileStream = new FileOutputStream(file);

//create printWriter object using FileOutputStream
PrintWriter printWriter = new PrintWriter(fileStream);

int number[] = new int[10];
for (int i=0;i<number.length; i++) {
    number[i] = i+1;
    printWriter.println("Line Number:"+ number[i]);
}
printWriter.flush();
printWriter.close();

//create FileReader using file object
FileReader fileReader = new FileReader(file);

//create BufferedReader object using FileReader
BufferedReader bufferReader = new BufferedReader(fileReader);
String inputStr = "";
//read line
inputStr = bufferReader.readLine();
while (inputStr != null){
    System.out.println(inputStr);
    inputStr = bufferReader.readLine();
} // end of while
bufferReader.close();
```