Software Development II

File Handling in Java

Self Guided Tutorial

[Engagement Week Activity with Video Demonstration]

Outline

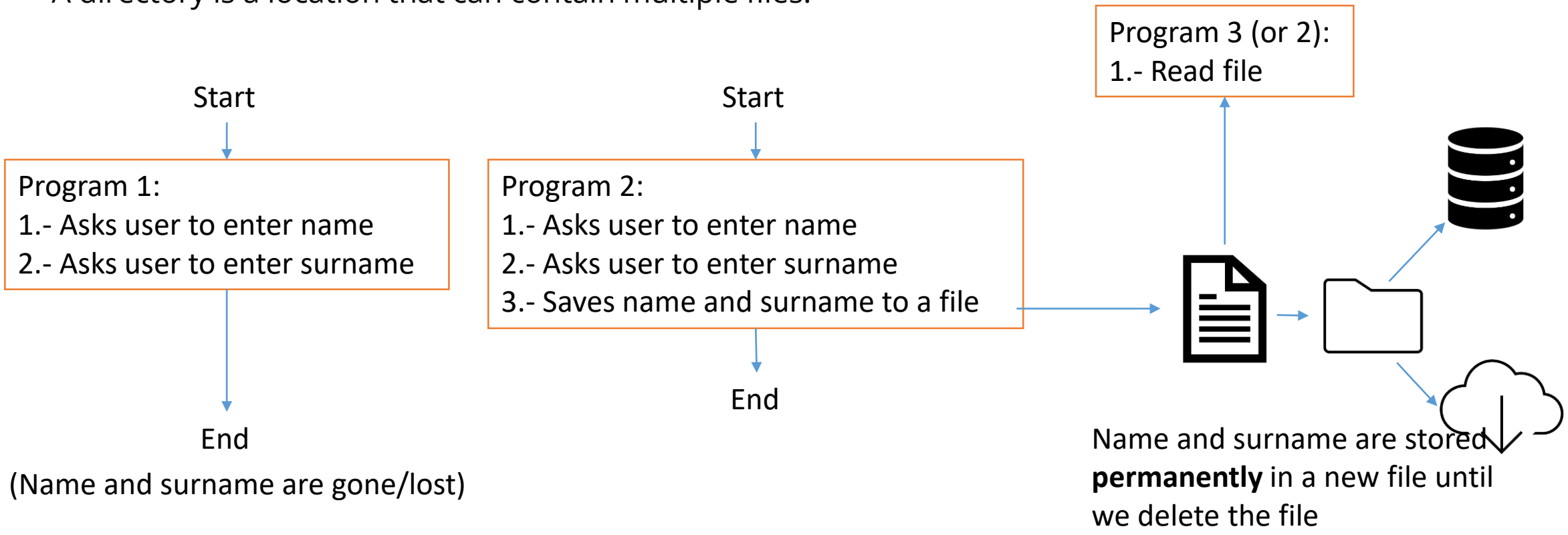
Files

- What is a file
- How to create a file
- How to write in a file
- How to read a file
- How to rename a file
- How to delete a file
- Directories

Video Demonstration Link

What is a file?

- A file is a named location (document, folder) where you can store information.
- There are many file types: text file, Word document, picture, voice, etc.
- You can create, read, update and delete files (and directories) in Java.
- A directory is a location that can contain multiple files.



Class File

- `import java.io.File;` ← Import the *File* class
- The *File* class has several methods to work with files

Method	Returns	Description	Example
`canRead()`	Boolean	Checks if the file is readable	`boolean readable = file.canRead();`
`canWrite()`	Boolean	Checks if the file is writable	`boolean writable = file.canWrite();`
`createNewFile()`	Boolean	Creates an empty file	`boolean var = file.createNewFile();`
`delete()`	Boolean	Deletes a file	`boolean deleted = file.delete();`
`exists()`	Boolean	Checks if the file exists	`boolean exists = file.exists();`
`getName()`	String	Returns the name of the file	`String name = file.getName();`
`getAbsolutePath()`	String	Returns the absolute pathname	`String path = file.getAbsolutePath();`
`length()`	Long	Returns the size of the file in bytes	`long length = file.length();`
`list()`	String[]	Returns a list of the files in a directory	`String[] list = file.list();`
`mkdir()`	Boolean	Creates a directory	`bool ok = file.mkdir();`

Use of Try Catch Block

- The try statement allows you to define a block of code to be tested for errors while it is being executed.
- The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

```
try {
  //  Block of code to try
}
catch(Exception e) {
  //  Block of code to handle errors
}
```

- Watch : Use of Try Catch Block

Creating a File

1. The File class from the java.io package, allows us to work with files.

```
import java.io.File;  // Import the File class (or
java.io.*
```

2. To use the File class, create an object of the class, and specify the filename or directory name: (text.txt is the name of the file we create

```
File file = new File("text.txt");
```

3. Attempt to create the file. Store the status of file creation in a Boolean variable

```
boolean file_created = file.createNewFile();
```

Creating a File

4. If file is successfully created, retrieve the name of the file and display it.

```
if (file_created){ // name of the bool variable
      System.out.println("File created:"+ file.getName());
           }
```

5. If file is not created, check weather the given file already exist.

```
if (file.exists()){
      System.out.println("File already exists.");
}
```

Creating a File- Complete code (with TryCatch)

```
import java.io.File;
import java.io.IOException;

public class Files {
 public static void main(String[] args) {

  try{
    File file = new File("text.txt");
    boolean file_created = file.createNewFile();
    if (file_created){
      System.out.println("File created: " + file.getName());
    }
else{
        System.out.println("Error while creating file: " + file.getName());
      }
    }
  }
  catch(IOException e){
    e.printStackTrace();
  }
 }
}
```

Reading from a file

1. The File class from the java.io package, allows us to work with files.

```
import java.io.File;  // Import the File class (or
java.io.*
```

2. To use the File class, create an object of the class, and specify the filename or directory name:

(Make sure a file already exist in stated folder before reading)

```
File inputFile = new File("filename.txt"); // Specify the filename (if
the file is stored in current folder
```

OR

```
File inputFile= new File("K:\\temp\\Jcode.txt"); // Provide the path
of the file if the file is from a different folder
```

Reading from a file - contd

3. Similar to reading from keyboard, user scanner class to read input from file.Make sur
you `import java.util.Scanner;` package to use Scanner class.

```
Scanner rf = new Scanner(inputFile);
```

4. Start reading from the file single line at a time

```
String fileLine;
        while ( rf.hasNext())
        {
                fileLine = rf.nextLine(); //or use next() to get word
                System.out.println(lineCount + " "+ fileLine);
        }
```

5. Close the file scanner

```
rf.close();
```

Read Data from File - Complete code (Tryout)

```
import java.io.File;
import java.io.IOException;
import java.util.Scanner;

public class Files {
 public static void main(String[] args) {

    try {
      File file = new File("text.txt");
      Scanner file_reader = new Scanner(file);
      while (file_reader.hasNextLine()) {
        String text = file_reader.nextLine();
        System.out.println(text);
      }
      file_reader.close();
    } catch (IOException e) {
      System.out.println("Error while reading a file.");
      e.printStackTrace();
    }
 }
}
```

Write data to a File

1. Verify whether io package is already imported

```
import java.io.File;
```

2. create an object of the *FileWriter* class, and specify the filename or directory name: (which you want to create the file)

```
FileWriter myWriter = new FileWriter("filename.txt");
```

3. Specify the content to be written to the file

```
myWriter.write("This is what I want to write into the file !");
```

4. Close the file writer object.

```
myWriter.close();
```

Write Data to the file – Full code with Try Catch

```
import java.io.FileWriter;   // Import the FileWriter class
import java.io.IOException;  // Import the IOException class to handle errors or(java.io.*)
public class WriteToFile
{
  public static void main(String[] args)
  {
    try {
         FileWriter myWriter = new FileWriter("filename.txt");
         myWriter.write("This is what I want to write into the file");
         myWriter.close();
         System.out.println("Successfully wrote to the file.");
        }
    catch (IOException e)
        {
        System.out.println("An error occurred.");
        }
  }
}
```

Rename a file

- You can rename a file by creating two variables File:

```
File file = new File("MyFile.txt");
```
← Create a *File* with the old name

```
File file_newName = new File("MyNewNameFile.txt");
```
← Create a *File* with the new name

```
file.renameTo(file_newName);
```
← Call the `renameTo` method on the first *File* with the second *File* as parameter.

Files: Delete a file

You can delete files:

```
import java.io.File;
import java.io.IOException;

public class Files {
 public static void main(String[] args) {

    File file = new File("MyFile.txt");
    if (file.exists()){
        file.delete();
        System.out.println("File deleted.");
    }
    else{
        System.out.println("File does not exist.");
    }
 }
}
```

Check if file exists

Delete file

Create a directory (folder)

- To create a directory:

```
File myDirectory = new File("/path/directory");

if (!myDirectory.exists()){
        myDirectory.mkdirs();
}
```

Create a *File* with the directory we want to create

Check if the directory exists

Create the directory

List files in a directory

- You can list all the files in a directory

```
File directory = new File("./");

String[] files = directory.list();

for(int i = 0; i < files.length; i++){
        System.out.println(files[i]);
}
```

Create a *File* with the directory we want to list the files

We obtain the list of filenames as Strings

Print file name

- Once you have a list of filenames, you can:
  - Open them individually
  - Delete them
  - Rename them
  - Etc.

Files: Delete a directory

You can also delete folders:

```
import java.io.File;
import java.io.IOException;

public class Files {
 public static void main(String[] args) {

    File folder = new File("C:\\Users\\user_name\\folder");
    if (folder.exists()){
        folder.delete();
       System.out.println("Folder deleted.");
    }
    else{
        System.out.println("Folder does not exist.");
    }
 }
}
```

Folder name instead of file name

Check if folder exists

Delete folder

Exceptions Which may occur in File Handling

- Read
- https://www.tutorialspoint.com/java/java_exceptions.htm