Lesson 30 - File Handling

Writing to Files

The `System.IO` namespace has various classes that are used for performing numerous operations with files, such as creating and deleting files, reading from or writing to a file, closing a file, and more. The `File` class is one of them.

```
string str = "Hello File!";
File.WriteAllText("hello.txt", str);
```

The `WriteAllText()` method creates a file with the specified path and writes the content to it. If the file already exists, it is **overwritten**. The code above creates a file `test.txt` and writes the contents of the `str` string into it.

> To use the `File` class you need to use the `System.IO` namespace: `using System.IO;`

Reading from Files

You can read the content of a file using the `ReadAllText()` method of the `File` class.

```
string txt = File.ReadAllText("hello.txt");
Console.WriteLine(txt);
```

Other Methods of the File class

`AppendAllText()`

Appends text to the end of the file.

```
string path = "C:/temp/MyTest.txt";
string appendText = "This is extra text";
File.AppendAllText(path, appendText);
```

`Create()`

Creates a file in the specified location.

```
string path = "C:/temp/MyTest.txt";
File.Create(path);
```

`Delete()`

Deletes the specified file.

```
string path = "C:/temp/MyTest.txt";
File.Delete(path);
```

`Exists()`

Determines whether the specified file exists.

```
string path = "C:/temp/MyTest.txt";
Console.WriteLine(File.Exists(path) ? "File exists." : "File does not exist.");
```

`Copy()`

Copies a file to a new location.

```
string path = "C:/temp/MyTest.txt";
string pathCopied = "C:/temp/Copied.txt";
File.Copy(path, pathCopied);
```

`Move()`

Moves a specified file to a new location.

```
string path = "C:/temp/MyTest.txt";
string pathMoved = "C:/tempMove/MyText.txt";
File.Move(path, pathMoved);
```

> All methods automatically close the file after performing the operation.

```
// --------- //
// FILE INFO //
// --------- //

string str = "Hello World!";
string path = "C:/temp/";
string fileName = "hello.txt";

// ------------- //
// WRITE TO FILE //
// ------------- //
```

```
File.WriteAllText(path + fileName, str);
Console.WriteLine(str);

// -------------- //
// READ FROM FILE //
// -------------- //

string txt = File.ReadAllText(path + fileName);
Console.WriteLine(txt);

// -------------- //
// APPEND TO FILE //
// -------------- //

string appendText = DateTime.Now.ToString();
File.AppendAllText(path + fileName, appendText + "\n");

// --------------- //
// CREATE NEW FILE //
// --------------- //

File.Create(path + "newFile.txt");

// ------------- //
// DELETE A FILE //
// ------------- //

File.Delete(path + "newFile.txt");

// ------------------ //
// FILE EXISTS OR NOT //
// ------------------ //

Console.WriteLine(File.Exists(path + fileName)
    ? "File exists."
    : "File does not exist.");

// --------- //
// COPY FILE //
// --------- //

File.Copy(path + fileName, path + "copiedFile.txt");

// --------- //
// MOVE FILE //
// --------- //

File.Move(
    path + "move/" + fileName,
    path + "movedFile.txt");
```

ConfigHandler.cs

```
namespace FileHandlingLesson30
{
    public enum Theme { LIGHT, DARK }

    public static class ConfigHandler
    {
        private static string username;
        private static int age;
        private static Theme mode;
        private static bool configLoaded = false;

        private static string ConfigPath { get; set; }

        public static string Username
        {
            get { CheckConfig(); return username; }
            set { username = value; WriteConfig(); }
        }

        public static int Age
        {
            get { CheckConfig(); return age; }
            set { age = value; WriteConfig(); }
        }

        public static Theme Mode
        {
            get { CheckConfig(); return mode; }
            set { mode = value; WriteConfig(); }
        }

        public static void LoadConfig(string configPath)
        {
            ConfigPath = configPath;
            configLoaded = true;
            List<string> config = File.ReadLines(configPath).ToList();

            for (int i = 0; i < config.Count; i++)
            {
                if (config[i].Equals("USER"))
                    Username = config[i+1];

                if (config[i].Equals("AGE"))
```

```
                    Age = int.Parse(config[i+1]);

                if (config[i].Equals("MODE"))
                    Mode = (Theme)int.Parse(config[i + 1]);
            }
        }

        private static void WriteConfig()
        {
            CheckConfig();

            string config = "USER\n" + Username + "\n\n";
            config += "AGE\n" + Age + "\n\n";
            config += "MODE\n" + ((int)Mode) + "\n\n";

            File.WriteAllText(ConfigPath, config);
        }

        private static void CheckConfig()
        {
            try
            {
                if (!configLoaded)
                    throw new ConfigNotLoadedException();
            }
            catch (Exception ex)
            {
                Logger.LogError(ex);
            }
        }
    }

    public class ConfigNotLoadedException : Exception
    {
       public ConfigNotLoadedException()
            : base("Configuration File is not Loaded.") { }
    }
}
```

`Logger.cs`

```
namespace FileHandlingLesson30
{
    public static class Logger
    {
        private static string LogPath { get; } = "C:/temp/log.txt";

        public static void LogError(Exception ex, bool enableConsoleLogging =
true)
        {
            string logMessage = $"{DateTime.Now} | ERROR | " + ex.ToString() +
"\n\n";

            File.AppendAllText(LogPath, logMessage);
            if (enableConsoleLogging) LogConsole(logMessage);
        }

        public static void LogInfo(string message)
        {
            string logMessage = $"{DateTime.Now} | INFO | " + message + "\n\n";

            File.AppendAllText(LogPath, logMessage);
        }

        public static void LogConsole(string message)
            => Console.WriteLine(message);
    }
}
```

`Program.cs`

```
using FileHandlingLesson30;

string configPath = "C:/temp/config.ini";

ConfigHandler.LoadConfig(configPath);

try
{
    Console.WriteLine(ConfigHandler.Username);
    Console.WriteLine(ConfigHandler.Age);
    Console.WriteLine(ConfigHandler.Mode);
}
catch (Exception ex)
{
    Logger.LogError(ex, enableConsoleLogging: false);
}
```

`config.ini`

```
USER
Talha Salman


AGE
22
```

```
MODE
1
```