Pranav Patel

Directory & File Operations Using try-catch for exception handling

Kailas Patil • 15:27

10 points

---

that uses try-catch to demonstrate . :

-
- Creating directories & Files
- Checking existence of Directory & Files
- Listing all files and subdirectories
- Creating, writing, reading files- Write Text,Append Text
- Copying, moving, deleting, rename files and directories
- Using try-catch for exception handling

Code:

```
using System;
using System.IO;

class FileDirectoryDemo
{
    static void Main()
    {
        try
        {
            string dirPath = @"C:\TestDir";
            string filePath = Path.Combine(dirPath, "file.txt");
            string copyPath = Path.Combine(dirPath, "file_copy.txt");

            // 1. Creating Directory
            if (!Directory.Exists(dirPath))
            {
                Directory.CreateDirectory(dirPath);
                Console.WriteLine("Directory created: " + dirPath);
```

```
            }
            else
            {
                Console.WriteLine("Directory already exists.");
            }

            // 2. Creating File
            if (!File.Exists(filePath))
            {
                File.WriteAllText(filePath, "Hello, this is first line.\n");
                Console.WriteLine("File created: " + filePath);
            }
            else
            {
                Console.WriteLine("File already exists.");
            }

            // 3. Checking existence
            Console.WriteLine("Directory exists? " +
Directory.Exists(dirPath));
            Console.WriteLine("File exists? " + File.Exists(filePath));

            // 4. Listing all files and subdirectories
            Console.WriteLine("\nFiles in directory:");
            foreach (string file in Directory.GetFiles(dirPath))
            {
                Console.WriteLine(" - " + file);
            }

            // 5. Writing, Reading, Appending
            File.AppendAllText(filePath, "This is new content.\n");
            Console.WriteLine("Text appended.");

            File.AppendAllText(filePath, "Appended line.\n");
            Console.WriteLine("Text appended.");

            string content = File.ReadAllText(filePath);
            Console.WriteLine("\nFile Content:\n" + content);

            // 6. Copying file
            File.Copy(filePath, copyPath, true);
            Console.WriteLine("File copied to: " + copyPath);

            // 7. Renaming file (Move)
            string renamedPath = Path.Combine(dirPath, "renamed_file.txt");
            File.Move(copyPath, renamedPath, true);
            Console.WriteLine("File renamed to: " + renamedPath);
```

```
            // 8. Deleting file
            File.Delete(renamedPath);
            Console.WriteLine("Renamed file deleted.");
        }
        catch (UnauthorizedAccessException ex)
        {
            Console.WriteLine("Access Error: " + ex.Message);
        }
        catch (IOException ex)
        {
            Console.WriteLine("IO Error: " + ex.Message);
        }
        catch (Exception ex)
        {
            Console.WriteLine("General Error: " + ex.Message);
        }
    }
}
```