

FileSystemWatcher

FileSystemWatcher - это класс в пространстве имен System.IO, который ожидает уведомления файловой системы об изменениях и инициирует события при изменениях каталога или файла в каталоге.

Постановка задачи: напишите консольное приложение, которое будет «наблюдать» за папкой и выводить сообщения об основных изменениях в ней. Все текстовые файлы (при добавлении, изменении, переименовании) копируйте в «backup» папку вместе с содержимым этих файлов. При конфликте имен в папке «backup» перезаписывать старый файл новым.

Решение: концептуально решение имеет следующий вид

1. Создание папок для работы программы
2. Запуск FileSystemWatcher
3. Ожидание от пользователя ввода символа, чтобы консольное приложение не прекратило свою работу моментально

```
static void Main(string[] args)
{
    SecureDirectories();
    StartWatcher();
    Console.WriteLine("Для окончания программы нажмите любую клавишу...");
    Console.ReadKey();
}
```

1. Создание папок для работы программы

Для создания корректных путей к папкам, которые будут работать на любых платформах воспользуемся классами Environment и File.

```
static string initialDir = "Files";
static string backupDir = "Backup";

private static void SecureDirectories()
{
    initialDir = Environment.CurrentDirectory + Path.DirectorySeparatorChar + initialDir;
    backupDir = Environment.CurrentDirectory + Path.DirectorySeparatorChar + backupDir;

    CreateDirectory(initialDir);
    CreateDirectory(backupDir);
}

private static void CreateDirectory(string directory)
{
    if (!Directory.Exists(directory))
        Directory.CreateDirectory(directory);
}
```

2. Создание и запуск FileSystemWatcher

Сначала создадим экземпляр класса `FileSystemWatcher`. Укажем для него папку, за которой следует «наблюдать». Далее подпишем наши методы на события этого класса. И запустим уведомления от этих событий.

```
private static void StartWatcher()
{
    FileSystemWatcher watcher = new FileSystemWatcher();
    watcher.Path = initialDir;
    watcher.Changed += OnChanged;
    watcher.Created += OnCreated;
    watcher.Deleted += OnDeleted;
    watcher.Renamed += OnRenamed;
    watcher.Error += OnError;
    watcher.EnableRaisingEvents = true;
}
```

Теперь нужно написать методы, которые будут срабатывать при получении уведомления от `FileSystemWatcher`.

```
private static void OnChanged(object sender, FileSystemEventArgs e)
{
    if (e.ChangeType == WatcherChangeTypes.Changed)
    {
        Console.WriteLine($"Файл был изменен: {e.Name}");
        CopyFile(e.Name);
    }
}

// Перенос файла в папку считается за создание
private static void OnCreated(object sender, FileSystemEventArgs e)
{
    Console.WriteLine($"Новый файл был создан: {e.Name}");
    CopyFile(e.Name);
}
```

Также напишем метод, который будет копировать файл в «backup» папку.

```
// Будет создавать копии всех текстовых файлов в папке Backup
private static void CopyFile(string filename)
{
    if (filename.EndsWith(".txt"))
    {
        File.Copy(Path.Combine(initialDir, filename), Path.Combine(backupDir, filename), true);
        Console.WriteLine($"Файл был скопирован: {filename}");
    }
}
```

Методы `OnDeleted`, `OnRenamed`, `OnError` предлагается написать самим. Они есть в прикреплённом решении для преподавателя.