Практикум. Перечисление и мониторинг файлов

Упражнение 1. Перечисление файлов в папке

Сейчас вы "перечислите файлы на выбранном диске.

1. Создайте новое консольное приложение с именем ShowFilesDemo.
2. Добавьте директиву using для импорта в новый проект пространства имен System.IO.
3. Добавьте метод ShowDirectory, принимающий в качестве параметра объект Directorylnfo.
4. В добавленном методе напишите цикл для перебора всех файлов в папке и вывода их имен на консоль, Полученный код выглядит примерно так:

//C#

static void ShowDirectory(Directorylnfo dir)

{

// Показать все файлы

foreach (Filelnfo file in dir.GetFilesO)

{

Console.WriteLine("File: {0}", file.FullName);

}

}

5. В теле метода ShowDirectory переберите в цикле все вложенные папки, вызывая метод ShowDirectory. Рекурсивно вызываемый метод ShowDirectory найдет все файлы в  
каждой папке. Полученный код выглядит примерно так:

// C#

// Рекурсивный перебор // вложенных папок

foreach (Directorylnfo subDir in dir.GetDirectoriesO) {

ShowDirectory(subDir);

}

6. Поместите в тело метода Main код, создающий новый экземпляр объекта Directorylnfo  
для каталога Windows, и вызовите новый метод ShowDirectory, передав ему этот  
объект в качестве параметра. Ниже приводится соответствующий пример кода\*

// C#

Directorylnfo dir = new Directorylnfo(Environment.SystemDirectory); ShowDirectory(dir);

7. Соберите проект и исправьте ошибки. Убедитесь, что консольное приложение пра-  
вильно перечисляет все файлы в системном каталоге (Environment.SystemDirectory).

Упражнение 2. Мониторинг изменений файловой системы

Ваша задача — отследить изменения всех файлов с расширением .ini.

1 Создайте новое консольное приложение с именем FileWatchingDemo.

1. Импортируйте пространство имен System.IO namespace в новый файл.
2. Создайте новый экземпляр класса FileSystem Watcher для системного каталога. Напри­мер, это можно сделать так:

// C#

FileSystemWatcher watcher =

new FileSystemWatcher(Environment.SystemDirectory);

4. Установите свойства созданного экземпляра объекта-наблюдателя за файловой системой, чтобы он контролировал только .ini-файлы во всех вложенных папках и уведомлял только об изменении атрибутов или размера файла. Полученный код может  
выглядеть примерно так:

// C#

watcher.Filter = "\*.ini";

watcher.IncludeSubdirectories = true;

watcher.NotifyFilter = NotifyFilters.Attributes | NotifyFilters.Size;

5. Чтобы отслеживать изменения, добавьте к объекту-наблюдателю обработчик события Changed. Например, это можно сделать так:

watcher.Changed +=

new FileSystemEventHandler(watcher\_Changed);

6. Далее потребуется создать метод, обрабатывающий событие Changed. Добавьте в тело  
этого метода код, выводящий на консоль имя измененного файла. Полученный код  
выглядит примерно так:

// C#

static void watcher\_Changed(object sender, FileSystemEventArgs e)

{

Console.WriteLine("Changed: {0}", e.FullPath);

}

1. Чтобы объект-наблюдатель начал генерировать события, установите свойство EnabhngRaisingEvents равным true.
2. Соберите проект и исправьте ошибки. Убедитесь, что при изменениях атрибутов или размера любого .ini файла консольное приложение выдает сообщение.