ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ Федеральное государственное бюджетное образовательное учреждение высшего образования «МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ» Кафедра «Информационная безопасность»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

по дисциплине

«Защита информации от вредоносного программного обеспечения»

Разработка модуля мониторинга файловой системы

Выполнила:

студентка группы БВТ1601 Фурман С.И.

Проверил:

ассистент кафедры ИБ

Барков В. В.

**Разработка модуля мониторинга файловой системы**

Для своевременного обнаружения заражённых файлов необходимо отслеживать изменения в определённых каталогах файловой системы. При изменении файлов или создании новых файлов необходимо передать пути к новым или изменённым файлам модулю подготовки файлов для сканирования.

***Задание***

Разработать класс, который отслеживает изменения в заданных директориях. При обнаружении изменений передать файлы для сканирования модулю сканирования. Для взаимодействия с модулем подготовки файлов для сканирования использовать паттерн «Наблюдатель»

**Текст модулей**

FileWatcher - класс, отслеживающий изменения в директориях.

***FileWatcher***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.IO;

using Microsoft.Win32;

using System.Collections;

using System.ComponentModel;

using System.Windows.Forms;

using System.Data;

using System.Configuration;

using Microsoft.Win32;

namespace AntiVirus

{

class FileWatcher : ISubject

{

// --------------------------------Наблюдатель------------------------------------------------------

private List<IObserver> \_observers = new List<IObserver>();

// Методы управления подпиской.

public void Attach(IObserver observer)

{

Console.WriteLine("FileWatcher: Добавлен наблюдатель.");

this.\_observers.Add(observer);

}

public void Detach(IObserver observer)

{

this.\_observers.Remove(observer);

Console.WriteLine("FileWatcher: Удаление наблюдателя.");

}

public void Notify()

{

Console.WriteLine("FileWatcher: Уведомление наблюдателей...");

foreach (var observer in \_observers)

{

observer.Update(this);

}

}

// --------------------------------Наблюдатель------------------------------------------------------

FileSystemWatcher FileMonitor = new FileSystemWatcher();

RegistryKey key\_create = Registry.CurrentUser.CreateSubKey(@"Software\Antivirus");

public string Put {get;set;}

private string Filter = "\*.\*";

private bool IncludeSubs = true;

private System.ComponentModel.Container components = null;

public string FullPath\_Create { get; set; }

public string ShortPath\_Create { get; set; }

public bool BoolPath\_Create { get; set; }

public string FullPath\_Deleted { get; set; }

public string ShortPath\_Deleted { get; set; }

public bool BoolPath\_Deleted { get; set; }

// Изменение названия

private void FileMonitor\_OnRenamed(object source, RenamedEventArgs e)

{

//Старое имя

string originalname = e.OldFullPath;

//Новое имя

string renamed = e.FullPath;

//Console.WriteLine("File: " + originalname + " renamed to " + renamed, e.OldName + " Renamed");

string line;

line = String.Format("File: {0} renamed to {1}, {2} Renamed", originalname, renamed, e.OldName);

Program.fm1.Filewatcher\_listbox(line);

}

//Выбор изменений произошедших с файлом

private void FileMonitor\_Changed(object sender, System.IO.FileSystemEventArgs e)

{

string ChangeType = e.ChangeType.ToString();

//Добавлен

if (ChangeType == "Created")

{

Console.WriteLine("File: " + e.FullPath + " " + e.ChangeType, e.Name + " Created");

FullPath\_Create = e.FullPath;

ShortPath\_Create = e.Name;

BoolPath\_Create = true;

this.Notify();

BoolPath\_Create = false;

}//Удален

else if (ChangeType == "Deleted")

{

Console.WriteLine("File: " + e.FullPath + " " + e.ChangeType, e.Name + " Deleted");

FullPath\_Deleted = e.FullPath;

ShortPath\_Deleted = e.Name;

BoolPath\_Deleted = true;

this.Notify();

BoolPath\_Deleted = false;

}//

else if (ChangeType == "Changed")

{

Console.WriteLine("File: " + e.FullPath + " " + e.ChangeType, e.Name + " Changed");

}

}

//Начало слежения

public void FileMonitor\_Start()

{

FileMonitor.Deleted += new System.IO.FileSystemEventHandler(FileMonitor\_Changed);

FileMonitor.Renamed += new System.IO.RenamedEventHandler(FileMonitor\_OnRenamed);

FileMonitor.Changed += new System.IO.FileSystemEventHandler(FileMonitor\_Changed);

FileMonitor.Created += new System.IO.FileSystemEventHandler(FileMonitor\_Changed);

Put = key\_create.GetValue("filewatcher").ToString();

//Путь до папки из textbox

if (Put == "")

{

Console.WriteLine("Директория не указана!");

return;

}

//Настройка параметров для монитора

//Путь до папки

//FileMonitor.Path = Path.ToString();

FileMonitor.Path = Put.ToString();

//Фильтр

FileMonitor.Filter = Filter.ToString();

//Вкл/Выкл поддирикторий

FileMonitor.IncludeSubdirectories = IncludeSubs;

FileMonitor.NotifyFilter = NotifyFilters.LastAccess | NotifyFilters.LastWrite

| NotifyFilters.FileName | NotifyFilters.DirectoryName;

//Начать мониторингь

FileMonitor.EnableRaisingEvents = true;

Program.fm1.Filewatcher\_listbox("FileWatcher Работает");

}

// Остановка слежения

public void FileMonitor\_Stop()

{

FileMonitor.EnableRaisingEvents = false;

Program.fm1.Filewatcher\_listbox("FileWatcher Отключен");

}

}

}