**Федеральное агентство связи ордена Трудового Красного Знамени федеральное государственное бюджетное общеобразовательное учреждение высшего образования «Московский технический университет связи и информатики»**

Кафедра Информационная безопасность

Отчет по Антивирусу

По дисциплине «Защита информации от вредоносного программного обеспечения»

Выполнил: студент группы БВТ1801

Клюшкин Дмитрий Алексеевич

Проверил:

Барков Вячеслав Валерьевич

Москва 2021

Содержание

[Введение 3](#_Toc67684604)

[Динамическая библиотека классов 4](#_Toc67684605)

[SearchFile 4](#_Toc67684606)

[ScanObject 4](#_Toc67684607)

[ScanFile 5](#_Toc67684608)

[FileObserver 5](#_Toc67684609)

[DateValue 5](#_Toc67684610)

[DBManager 5](#_Toc67684611)

[MCF и IMCF 7](#_Toc67684612)

[Служба 8](#_Toc67684613)

[Клиент 11](#_Toc67684614)

[Сканирование 11](#_Toc67684615)

[Мониторинг директорий 12](#_Toc67684616)

[Карантин 13](#_Toc67684617)

[Отчетность 13](#_Toc67684618)

[Расписание 14](#_Toc67684619)

# Введение

При разработке собственного антивируса, следует определиться с важными моментами как: язык разработки службы и клиентского приложения, способ общения между ними, реализация хранения антивирусной сигнатуры, представление результатов сканирования и т.д. Немалую роль играет архитектура антивирусного комплекса. Важно придерживаться проектирования с применением принципов дизайна классов (S.O.L.I.D.) и принципов организации компонентов. Это придаст коду большую читабельность и позволит избежать возникновения различных проблем.

# Динамическая библиотека классов

В проекте имеется динамическая библиотека классов AntiLib.dll. Она содержит все необходимые методы для работы сервиса. В неё входят:

* SearchFile – рекурсивный поиск PE файлов;
* ScanObject – извлечение секции .text из PE файла;
* ScanFile – поиск сигнатуры в экземпляре класса ScanObject;
* FileObserver – создание события на мониторинг директории;
* DateValue – содержит глобальные переменные для создания отчета и таймер для запуска сканирования по расписанию;
* DBManager – содержит все необходимые запросы;
* MCF и IMCF – класс для работы WCF.

## SearchFile

Статический класс, который содержит два публичных метода: SearchFileDirectory и GetTypeFile.

SearchFileDirectory принимает на вход путь до директории/файла и операцию, которую надо будет совершить над файлом, если он окажется заражен. Операция над файлом не потребуется в текущем методе, но будет передана последующим. Этот метод может завершить сканирование, если все файлы были отсканированы и все объекты для сканирования были переданы. Найденные файлы передаются в ScanObject. Данный метод работает в совокупности с GetTypeFile.

GetTypeFile принимает на вход путь до файла, считывает первый байт и если имеется начало PE файла, отправляет обратно значение для уведомления.

## ScanObject

Класс, который содержит конструктор для извлечения секции .text из PE файла. Считывает расположение .text и его размер и извлекает его в массив байтов. Используется для передачи классу ScanFile.

## ScanFile

Статический класс, использующий пул потоков для поиска сигнатуры. Содержит конструктор, который ограничивает максимально количество потоков до трех, метод AddTask и ThreadProc.

AddTask принимает на вход экземпляр класса ScanObject и помещает задачу в пул потоков.

ThreadProc, метод, который производит поиск сигнатуры, производя запросы к базе данных. При обнаружении зловредной сигнатуры, файл удаляется или помещается в карантин путем замены первого байта на ‘Q’. Если количество просканированных файлов совпадает с количеством найденных файлов и поиск завершен, то происходит завершение сканирования и создание отчета.

## FileObserver

Класс, который подписывается на события операционной системы об изменении внутри указанной директории. Конструктор класса принимает путь и операцию, которую требуется совершить с файлов, если в папке были сделаны изменения содержимого файлов. Содержит метод отписки от событий.

## DateValue

Статический класс, содержащий информацию о сканирование, а также метод для создания таймера. Описание переменных:

* bool isScaning – состояние сканирования;
* bool isSearchFile – состояние поиска;
* int countCheckFile – количество проверенных файлов;
* int countFile – количество найденных файлов для сканирования;
* int countVirusFile – количество вирусов;
* int idReport – следующий id для таблицы report;
* string dateStart – дата и время старта сканирования;
* string path – начальная директория сканирования.

## DBManager

Статический класс для работы с базой данных. Содержит методы:

* GetObserver – возвращает все строки из таблицы observer;
* UpdateObserver – принимает старый, новый путь и операцию выполняемую над файлом. Изменяет директория мониторинга и действие над файлом на указанное значение;
* AddObserver – добавляет директорию для мониторинга. Принимает путь и операцию над файлом;
* DeleteObserver – удаляет все строки с указанной директорией;
* GetQuarantine – возвращает все файлы находящиеся в карантине;
* AddQuarantine – добавляет файл в карантин. На вход принимает дату, путь и тип вируса;
* DeleteQuarantine – удаляет файл из карантина. На вход принимает имя файла;
* AddSignature – добавляет сигнатуру. На вход принимает сигнатуру в виде строки байтов в hex представлении, смещение слева, смещение справа и тип вируса;
* DeleteSignature – удаление сигнатуры;
* SearchSignature – поиск сигнатуры, которая начинается с казанных байтов;
* GetTypeSignature – возвращает тип вируса по сигнатуре;
* GetTime – возвращает все строки из таблицы time;
* AddTime – добавляет расписание. На вход принимает время, путь и что требуется сделать с зараженными файлами;
* IsTime – проверка на наличие указанного времени;
* DeleteTime – удаляет время. На вход принимает время и директорию;
* GetFile – возвращает все просканированные PE файлы;
* AddFile – добавляет файл в базу. На вход принимает id отчета, путь, совершенную операцию над файлом и тип вируса;
* GetReport – возвращает все отчеты;
* GetNextIdReport – возвращает следующий id отчета;
* AddReport – добавляет отчет в базу

## MCF и IMCF

MCF, класс реализующий интерфейс IMCF, который в свою очередь содержит методы для общения клиента с сервером. Часть методов это вызов методов из класса DBManager. Другую часть составляют методы запуска сканирования, статуса, запуска мониторинга и расписания.

# Служба

Служба создана в Visual studio «Служба Windows (.Net Framework)». Для общения с клиентом добавлен класс AntivirusServiceProxy. Он позволяет клиенту получать данные, используя методы, реализованные в классе MCF динамической библиотеки. При запуске, запрашивает и создает объекты класса FileObserver для мониторинга и инициализирует таймер для сканирования по расписанию.

Установка производится запуском InstallService.bat с правами администратора. После запуска, служба находится в отключенном состоянии. Если требуется удалить службу, то запускается UninstallService.bat с правами администратора. В комплекте со службой идет база данных AntiDB.db. В ней содержатся следующие таблицы:

1. Карантин. Содержит сигнатуру в hex представлении, её длину, смещение слева и справа, а также наименование вируса.

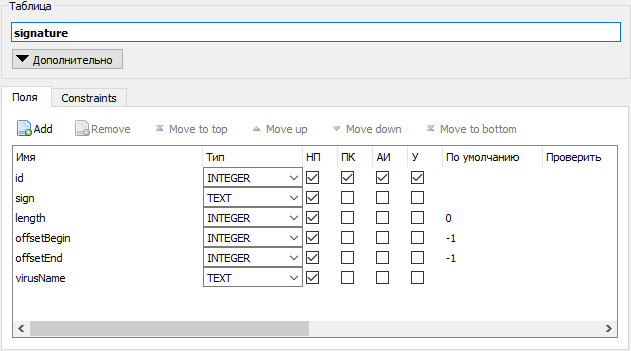


Рисунок 1 - Таблица signature

1. Отчеты. Содержит информацию о сканировании: дату начала и окончания сканирования, директорию, сколько всего было просканировано файлов и количество найденных вирусов.

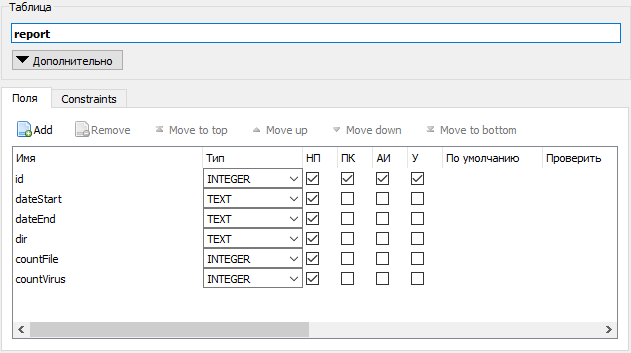


Рисунок 2 - Таблица report

1. Файлы. Содержит все файлы, которые были помещены в карантин или удалены.

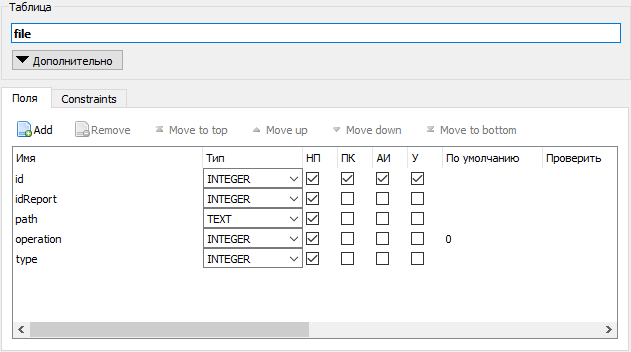


Рисунок 3 - Таблица file

1. Карантин. Содержит все файлы, находящиеся в карантине.

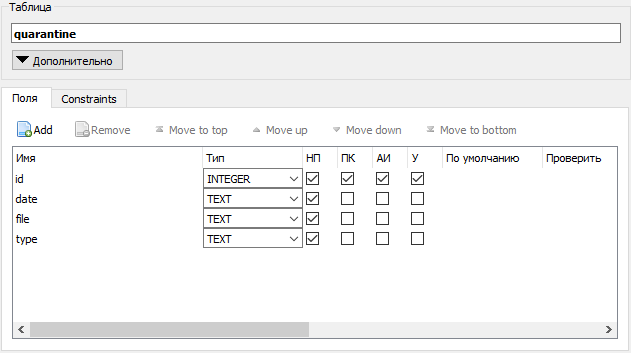


Рисунок 4 – Таблица quarantine

1. Мониторинг. Содержит директории для мониторинга.

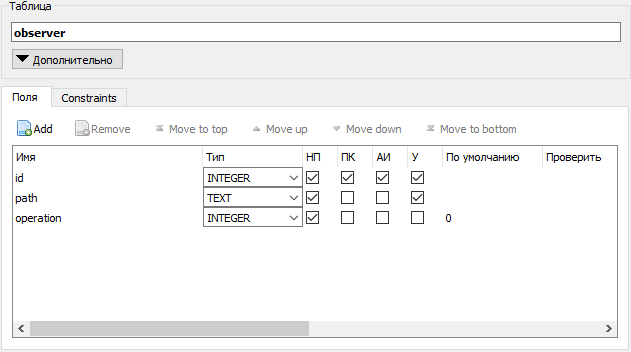


Рисунок 5 – Таблица observer

1. Расписание. Содержит директории для сканирования по расписанию.

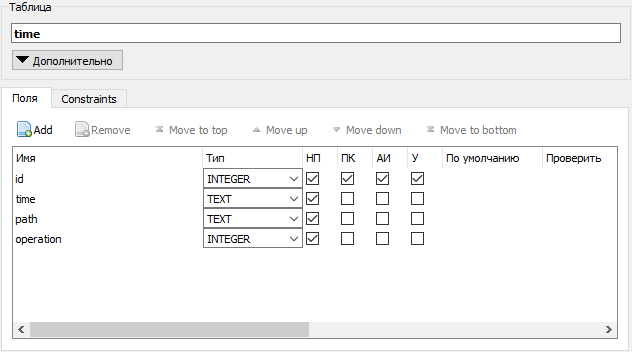


Рисунок 6 – Таблица time

# Клиент

Содержит в себе пять пунктов основного меню: сканирование, мониторинг директорий, карантин, отчетность, расписание.

## Сканирование

Пункт меню «Сканирование» позволяет указать файл или папку для сканирования, действие над файлами. Задание директории возможно через прямой ввод пути, через перетаскивание его в специальное поле или с помощью кнопки обзора «…». После нажатия на кнопку «Начать сканирование» появляется прогресс сканирования.

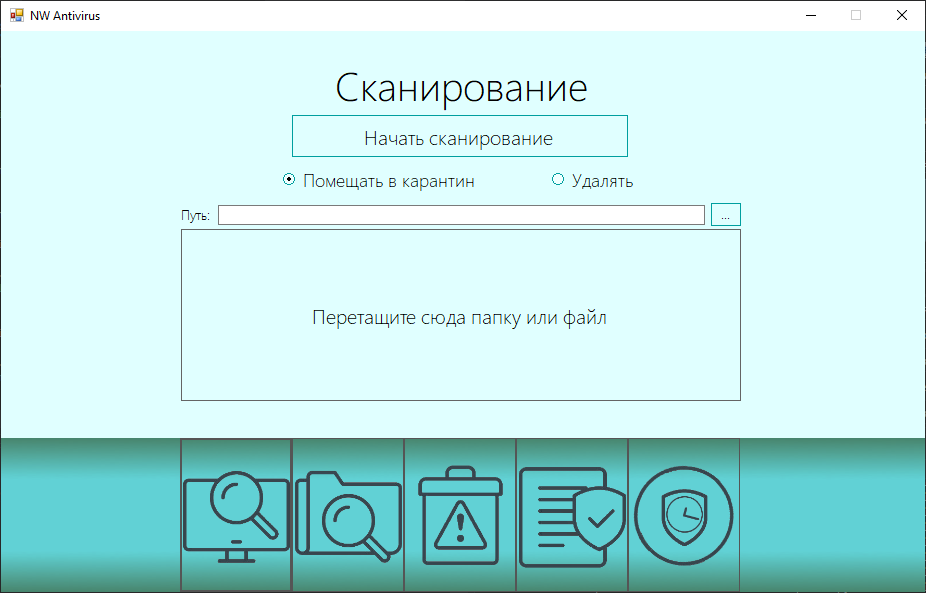


Рисунок 7 – пункт меню «Сканирование»

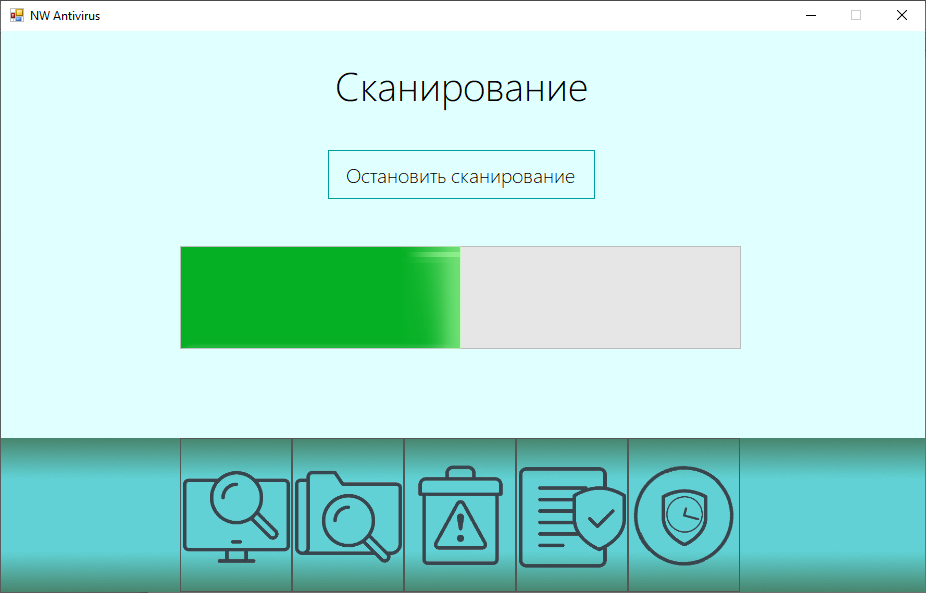


Рисунок 8 - Прогресс сканирования

## Мониторинг директорий

Пункт меню «Мониторинг директорий» позволяет добавить, удалить или отредактировать задачу на мониторинг директории. При нажатии на «Добавить» или «Редактировать» появится дополнительная панель для указания нового и пути и операции совершаемой над файлом.

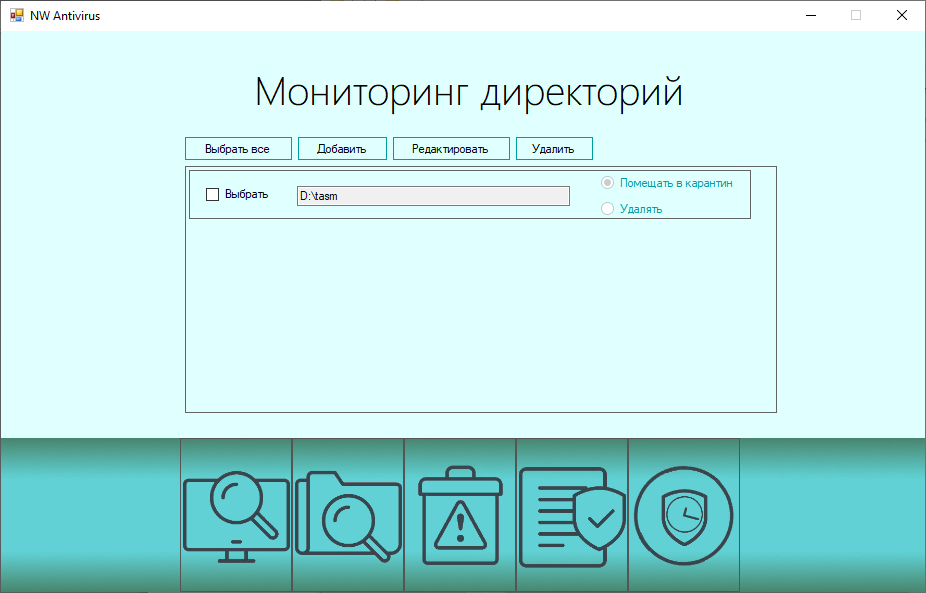


Рисунок 9 – пункт меню «Мониторинг директорий»

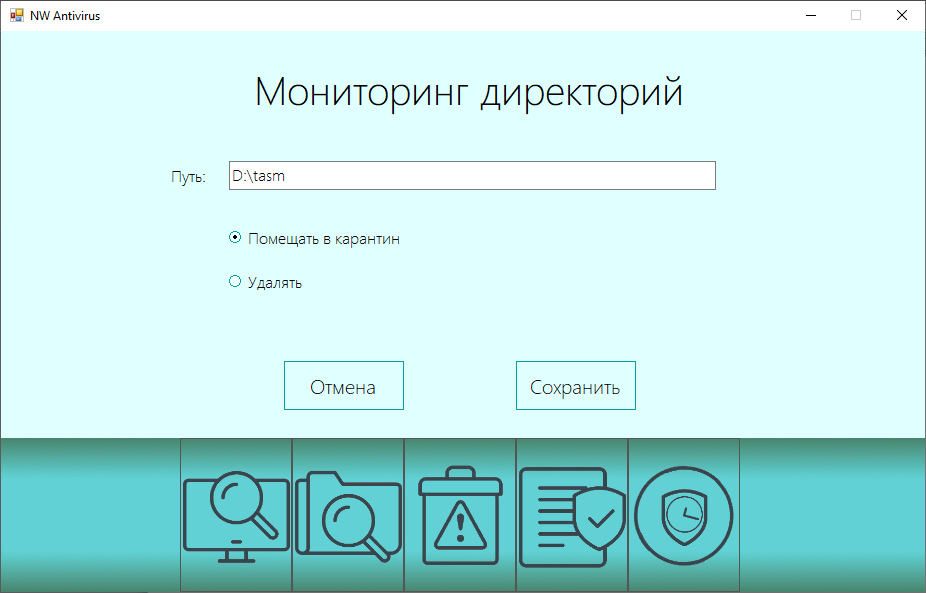


Рисунок 10 – панель добавления/редактирования мониторинга

## Карантин

Пункт меню «Карантин» позволяет восстановить или удалить файл, который считается зараженным.

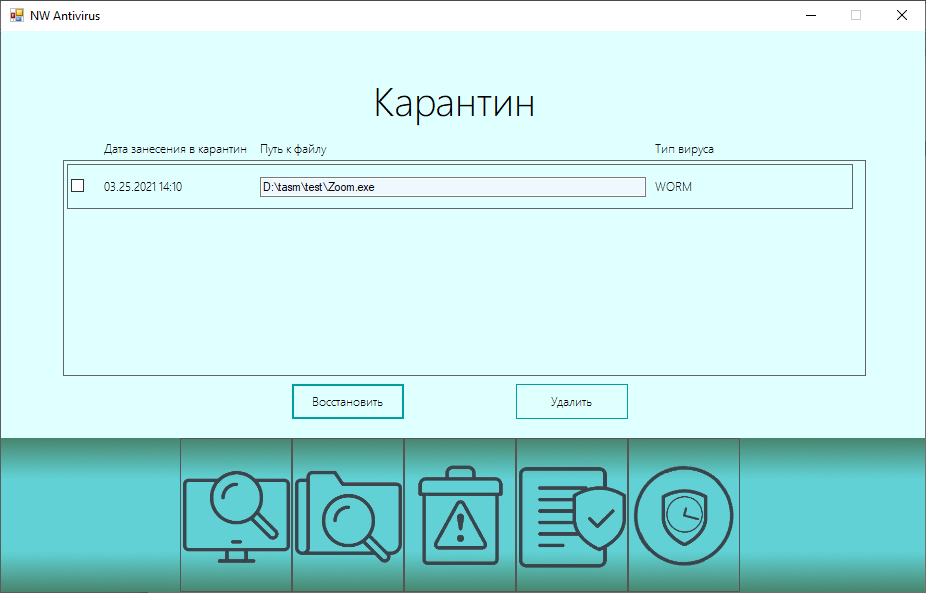


Рисунок 11 – пункт меню «Карантин»

## Отчетность

Пункт меню «Отчетность» отображает информацию о прошедших сканированиях, а при нажатии кнопки «Показать», появляется информация о найденных файлах. Для выхода требуется нажать ЛКМ за пределами панели информации о файлах.

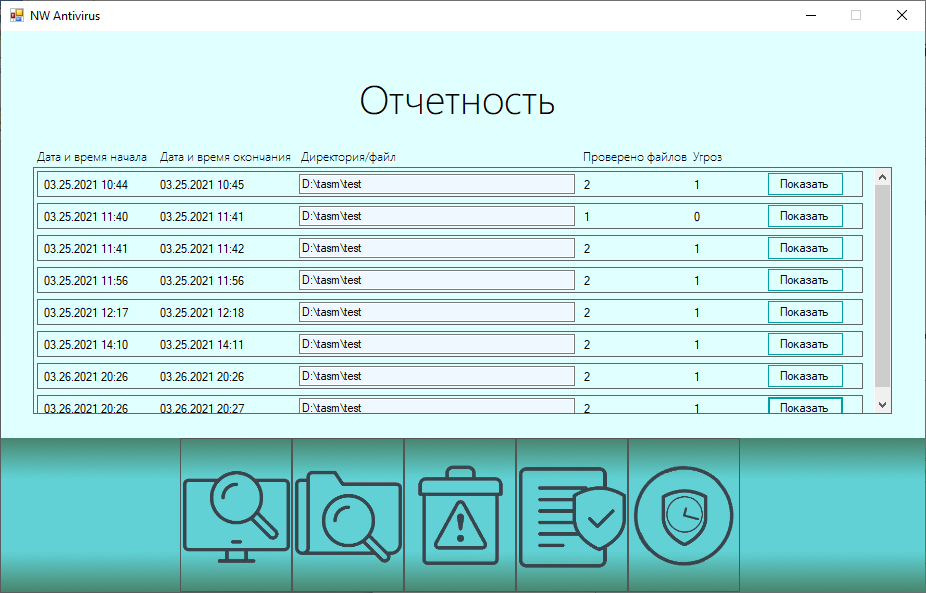


Рисунок 12 – пункт меню «Отчетность»

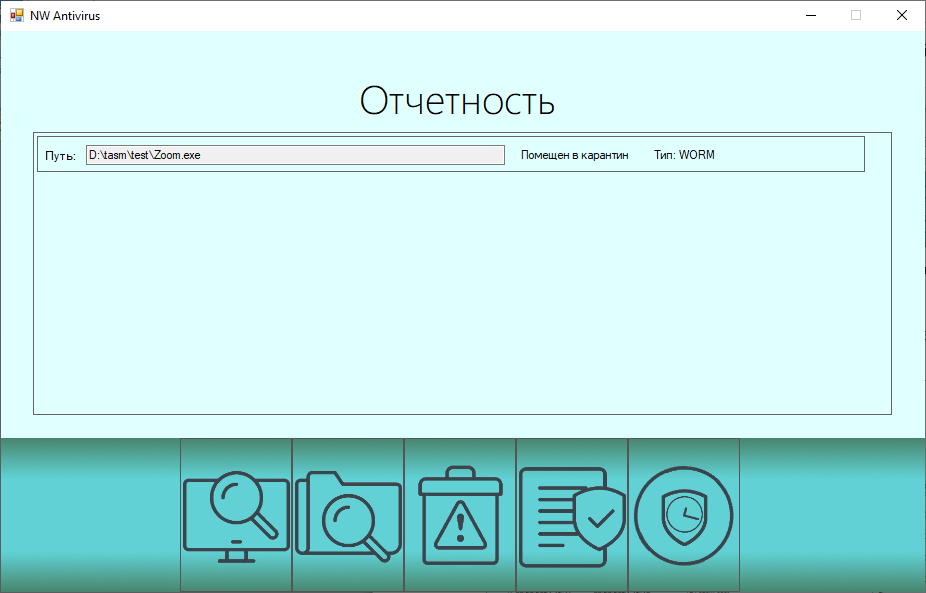


Рисунок 13 – информация о найденных файлах

## Расписание

Пункт меню «Расписание» отображает время и директорию для сканирования по расписанию.

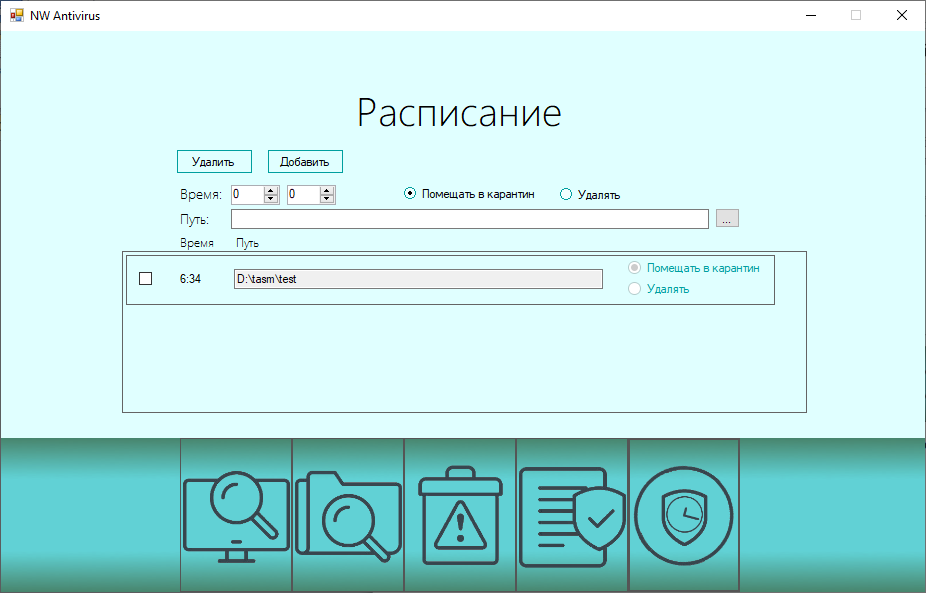


Рисунок 14 – пункт меню «Расписание»