Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Рязанский Государственный Радиотехнический Университет»

Кафедра ЭВМ

*К защите:*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

дата, подпись

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ**

по дисциплине

**«Операционные системы и оболочки»**

**Тема**

**Разработка системной вспомогательной многопоточной программы в системе программирования Visual Studio 2017 с использованием средств WinAPI**

Выполнил студент группы 540

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Данилов И.Д.

дата сдачи на проверку, подпись

Руководитель

доцент Засорин С.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

оценка дата защиты, подпись

Рязань 2018 г.

Оглавление

[**1. Введение 3**](#_Toc515255683)

[**Основная часть 3**](#_Toc515255684)

[**2. Техническое задание. 3**](#_Toc515255685)

[**2.1. Основания для разработки программы. 3**](#_Toc515255686)

[**2.2. Назначение разработки. 4**](#_Toc515255687)

[**2.3. Требования к программе. 4**](#_Toc515255688)

[**2.3.1. Требования к функциональным характеристикам. 4**](#_Toc515255689)

[**2.3.2. Требования к исходным кодам и языкам программирования. 4**](#_Toc515255690)

[**2.3.3. Требования к составу и параметрам технических средств 4**](#_Toc515255691)

[**2.4. Требования к надёжности 5**](#_Toc515255692)

[**2.5. Требования к программной документации 5**](#_Toc515255693)

[**2.6. Текст задания в соответствии с вариантом 5**](#_Toc515255694)

[**3. Описание программы. 5**](#_Toc515255695)

[**3.1. Общие сведения. 5**](#_Toc515255696)

[**3.2. Функциональное назначение 6**](#_Toc515255697)

[**3.2.1. Классы решаемых задач. 6**](#_Toc515255698)

[**3.2.2. Назначение программы. 6**](#_Toc515255699)

[**3.3. Описание логической структуры. 6**](#_Toc515255700)

[**3.3.1. Структура программы с описанием функций составных частей и связи между ними. 6**](#_Toc515255701)

# Введение

Целью данной работы является разработка, отладка и тестирование приложения в объектно-ориентированной системе программирования Visual Studio 2018 с использованием средств Windows API.

В приложении должен быть предусмотрен пользовательский интерфейс.

**Наименование программы**

“NTFS2Threads”

Для разработки приложения был выбран объектно-ориентированный язык программирования C#.

# Основная часть

# Техническое задание.

## Основания для разработки программы.

Разработка программы производится в соответствии с указанным в пункте 2.6 заданием.

## Назначение разработки.

Функциональным назначением программы NTFS2Threads является копирование структуры каталогов с одного логического диска на другой, а также анализ расположения группы файлов на диске по отрезкам.

## Требования к программе.

### Требования к функциональным характеристикам.

Программа NTFS2Threads должна обеспечивать возможность выполнения следующих функций:

* Выбор структуры каталогов на диске
* Выбор логического диска для вставки структуры каталогов
* Копирование выбранной структуры каталогов на другой логический диск
* Выдача сообщения о возможности копирования структуры
* Выдача сообщения о результате копирования
* Задание группы файлов
* Формирование таблицы номеров кластеров, занимаемых файлами в заданной группе
* Вывод данной таблицы номеров кластеров, занимаемых файлами в заданной группе

### Требования к исходным кодам и языкам программирования.

Программа должна быть разработана в объектно-ориентированной системе программирования с применением средств API используемой операционной системы.

### Требования к составу и параметрам технических средств

В состав технических средств должен входить персональный компьютер (ПЭВМ). Рекомендуемые технические характеристики:

* Операционная система: не ниже Windows® 7™
* Рекомендуемый процессор: не хуже Intel® Core™ i7-2630QM
* Не менее 256 мБ оперативной памяти
* Экран с разрешением не менее 800x600, 256 цветов.

## Требования к надёжности

Не имеется, так как обработка вводимой пользователем информации обеспечивается операционной системой Windows.

## Требования к программной документации

Программная документация должна быть выполнена в соответствии с ГОСТ ЕСПД. Программа должна содержать справочную систему, включающую всю необходимую информацию по работе с программой.

## Текст задания в соответствии с вариантом

Вариант 25.

Разработать двухпоточное приложение, выполняющие следующие операции:

Физическое копирование содержимого каталожной структуры с одного логического диска на другой.

Анализ расположения по отрезкам группы файлов на логическом диске.

**Исходные данные**

1. Тип файловой системы дисков - NTFS.
2. Имя диска для оригинала.
3. Имя диска для копии.
4. Полные имена файлов в группе.
5. Полное имя файла для вывода результатов.

# Описание программы.

## Общие сведения.

Разработанная прикладная программа имеет следующие атрибуты:

* Название исполняемого файла – NTFS2Threads.exe
* Размер исполняемого файла – 3773 кБ
* Версия файла – 1.0.0.0
* Название продукта NTFS2Threads
* Язык – Русский

## Функциональное назначение

### Классы решаемых задач.

Данное приложение позволяет решать задачи, связанные с взаимодействием с файловой системой пользователя данной системы

### Назначение программы.

Основное назначение приложения NTFS2Threads – предоставление некоторой служебной аналитической информации о файловой системе.

В приложении должны быть реализованы следующие функции:

* Возможность выбора пользователем каталога, содержащего некоторую структуру каталогов на диске
* Возможность выбора логического диска для вставки каталога, содержащего структуру
* Проверка выбранного диска на соответствие файловой системе New Technology File System
* Копирование выбранной структуры каталогов на другой логический диск
* Возможность выбора пользователем группы файлов на диске
* Формирование таблицы номеров кластеров, занимаемых файлами в заданной группе
* Вывод сформированной таблицы номеров кластеров, занимаемых файлами в заданной группе

## Описание логической структуры.

### Структура программы с описанием функций составных частей и связи между ними.

Информационная система состоит из набора классов, список которых приведен на рисунке 1. Главным классом является класс Program, описание которого содержится в файле Program.cs. В нём описан главный метод программы – Main(), который вызывается при запуске приложения.

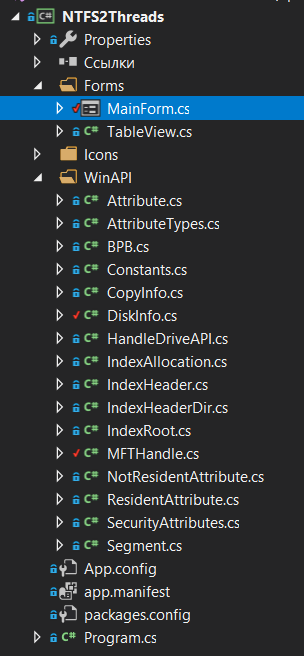


Рисунок . Список классов приложения NTFS2Threads

Обозначения, используемые на рисунке 1 и далее:

 – C# файл,

– файл формы,



– конфигурационный файл,



– файл манифеста приложения,



– свойства приложения



 – зависимости приложения

В каталоге WinAPI содержатся классы по работе с файловой системой NTFS. В классе HandleDriveAPI описаны обращения к системным функциям из библиотеки kernel32.dll. Класс Constants содержит необходимые для работы с файловой системой константы, такие, как количество байт в секторе, количество секторов в кластере и т.д. Класс DiskInfo содержит методы для работы с томом NTFS, такие, как чтение сектора по номеру, чтение кластера по номеру и т.д. В классе BPB определены методы по работе с блоком параметров BIOS, такие, как чтение и перевод в строчный формат содержащихся в BPB данных. Класс AttributeTypes содержит описание атрибутов записи MFT. В свою очередь класс MFTHandle предназначен для работы с записями MFT, загрузкой атрибутов записи, работой с индексными атрибутами, которые представлены в виде классов IndexRoot, IndexAllocation, IndexHeader, IndexHeaderDir. Класс AttributeTypes содержит описание типов атрибутов записей. Классы NotResidentAttribute и ResidentAttribute описывают соответственно нерезидентный и резидентный атрибуты. Класс Segment описывает один отрезок.

В каталоге Forms находится главная форма приложения, а также класс TableView, предназначенный для вывода в табличном виде информации о MFT записях файлов на экран.

Общий алгоритм работы приложения представлен на рисунке 2.



Рисунок – Общий алгоритм приложения

Продолжение см. на следующей странице



Рисунок 2 – Общий алгоритм приложения (продолжение)

Алгоритм, представленный на рисунке 2 реализуется следующим образом:

1. Пользователь посредством интерфейса выбирает задание, с которым намерен работать.
2. Если поток одного из заданий запущен, он приостанавливается и управление передается обработчику события нажатия кнопки пользователем. В этом обработчике запускается поток выбранного пользователем задания.
3. После того, как поток завершает работу, результирующие данные отображаются в программе.

Последовательность действий, выполняющаяся при выполнении обработки события выбора пользователем задания копирования структуры каталогов представлена на рисунке 3.



Рисунок – Алгоритм выполнения копирования структуры каталогов

Продолжение см. на следующей странице



Рисунок 3 – Алгоритм выполнения копирования структуры каталогов (продолжение)

Алгоритм, представленный на рисунке 3 реализуется следующим образом:

1. Пользователь выбирает каталог для копирования и диск, на который хочет скопировать каталог.
2. Запускается поток копирования, в который передаются путь до каталога и выбранный для копирования диск.
3. После чтения блока параметров BIOS производится проверка – является ли выбранный для копирования диск томом NTFS, если нет, выводится соответствующее сообщение. Если же диск является томом NTFS, то программа успешно копирует на него структуру каталогов и выводит соответствующее сообщение.

Последовательность действий, выполняющаяся при выполнении обработки события выбора пользователем задания анализа группы файлов по отрезкам представлена на рисунке 4.



Рисунок – Алгоритм анализа выбранной группы файлов по отрезкам

Алгоритм, представленный на рисунке 4 реализуется следующим образом:

1. Пользователь выбирает файлы для анализа.
2. Если пользователь не выбрал ни одного файла, выводится соответствующее сообщение.
3. Если пользователь выбрал файлы, то для каждого выбранного файла производится чтение пути файла и разбиение пути на строки, где каждая строка является именем каталога, который содержится в пути файла.
4. Для каждого каталога в пути к файлу считается номер записи в MFT. В конце концов мы доходим до номера записи в MFT о файле.
5. Производится вывод информации о данном файле.