Минобрнауки России
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Санкт-Петербургский государственный технологический институт
(технический университет)»

Направление подготовки	09.03.03 Прикладная информатика в химии
Направленность	Прикладная информатика в химии
Факультет	Информационных технологий и управления
Кафедра	Систем автоматизированного проектирования и управления
Учебная дисциплина	Программирование
Курс 1	**Группа** 485
Студент	Зобнин Илья Михайлович

Курсовой проект

Тема: Разработка программы, производящей поиск дублирующихся файлов на диске.

Студент	_____ (подпись)	_____ (дата)	И.М. Зобнин гр.№485
Руководители ст.преп.	_____ (подпись)	_____ (дата)	А.К. Федин
доц., к.т.н.	_____ (подпись)	_____ (дата)	И.Г. Корниенко

Санкт-Петербург
2019

Минобрнауки России
федеральное государственное бюджетное образовательное учреждение высшего образования
«Санкт–Петербургский государственный технологический институт (технический университет)»

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Направление подготовки		09.03.03 Прикладная информатика в химии	
Направленность		Прикладная информатика в химии	
Факультет		Информационных технологий и управления	
Кафедра		Систем автоматизированного проектирования и управления	
Учебная дисциплина		Программирование	
Курс	1	**Группа**	485
Студент		Зобнин Илья Михайлович	

Тема: Разработка программы, производящей поиск дублирующихся файлов на диске.

Цель работы: Разработка программы для поиска дублирующихся по имени, размеру, времени последнего изменения и по содержанию txt файлов на диске. Вывод информации о путях файлов, их размере, имени, времени последнего изменения.

Исходные данные по проекту:

1 Страуструп, Б. Programming: Principles and Practice Using C++ / Б. Страуструп. – М. : Вильямс, 2011. – 1248 с.
2 Норенков, И.П. Основы автоматизированного проектирования / И.П. Норенков. – М. : МГТУ им. Н. Э. Баумана, 2009. – 430 с.
3 Рождественский, Д. А. Автоматизация проектирования систем и средств управления: Т. 1: учеб. пособие / Д. А. Рождественский – Томск: Том. межвуз. Центр дистанц. Образования, 2004. – 167 с.
4 Вильямс, Я.Ш. C++: базовый курс / Я.Ш. Вильямс. – М. : Вильямс, 2014. – 624 с.
5 Липпман, С. Основы программирования на C++ / С. Липпман. – М. : Вильямс, 2014. – 1104 с.
6 Олифер, Н. А., Олифер, В. Г. Сетевые операционные системы / Н. А. Олифер В. Г. Олифер – СПб: Питер, 2009. – 669 с.

Перечень вопросов, подлежащих разработке:
1 Аналитический обзор
1.1 Обзор и анализ процесса поиска дублирующихся файлов на диске. Сравнительная характеристика существующих систем-аналогов.
1.2 Общая характеристика и особенности поиска дублирующихся файлов на диске.
1.3 Обзор и обоснование выбора инструментального программного обеспечения.
2 Цель и задачи курсового проекта.
3 Технологическая часть.

3.1 Формализованное описание процесса поиска дублирующихся файлов на диске как объекта обработки информации.
3.2 Постановка задач поиска дублирующихся файлов на диске.
3.3 Разработка функциональной структуры программного комплекса для поиска дублирующихся файлов на диске.
3.4 Создание алгоритма определения дубликата файла.
3.5 Разработка структуры интерфейса для пользователя.
3.6 Описание структур данных и алгоритмов (формат представления данных в памяти и на внешних носителях).
3.7 Описание структуры программы (модули, основные функции, классы и т.д.).
3.8 Тестирование программного комплекса (на заданном примере).
3.9 Оформление документации (пояснительной записки, презентации) по проекту.

Перечень графического материала:
1 Формализованное описание процесса поиска дублирующихся файлов на диске.
2 Функциональная структура программного комплекса по поиску дублирующихся файлов на диске.
3 UML-диаграмма использования программы пользователем.
4 Блок–схема алгоритма определения дубликата файла.
5 Тестовый пример работы программного комплекса по поиску дубликатов.
6 Характеристика аппаратного и программного обеспечения.

Требования к аппаратному и программному обеспечению:

Показатель	Значение
Тип ЭВМ	Персональный компьютер
Тактовая частота процессора, ГГц	3,2
Объем оперативной памяти, Гб	12
Объем внешней памяти, Гб	1024
Состав и характеристика периферийных устройств ЭВМ	Клавиатура, мышь, монитор с диагональю 24″ и разрешением 1920 точек
Операционная система	Windows 10
Прикладное программное обеспечение, необходимое для функционирования программного комплекса	Программы пакета Microsoft Office, Microsoft Visual Studio 17

Дата выдачи задания:

Дата предоставления курсового проекта к защите:

Заведующая кафедрой, проф.	Т. Б. Чистякова
Лектор, доц.	И. Г. Корниенко
Консультант, ст. преп.	А. К. Федин
Задание принял к выполнению	И. М. Зобнин

Оглавление

ВВЕДЕНИЕ

Темой курсового проекта стала разработка программы для поиска дублирующихся файлов на диске.

На компьютерах многих пользователей, как опытных, так и начинающих, зачастую скапливается множество дубликатов файлов, которые в большинстве случаев не несут в себе никакой практической пользы ни для системы, ни для пользователя, и лишь занимают лишнее место, которое могло быть использовано для хранения нужной пользователю информации. Составление программы для решения этой проблемы обусловили цель и задачи этого курсового проекта.

Поиск дублирующихся файлов на диске – это процесс, поиска и сопоставления файлов по выбранным пользователем критериям, будь то имя, размер или содержание.

Данный курсовой проект представляет собой портативную программу, ищущую дублирующиеся файлы на диске. Пользователь может выбрать любые из представленных критериев: имя, размер, последнее время изменения, содержание (только для текстовых файлов), а также расширение файла и папку, в которой будет производиться поиск.

1 АНАЛИТИЧЕСКИЙ ОБЗОР

1.1 Сравнительная характеристика существующих систем-аналогов

Систем-аналогов существует достаточно много. Большинство из них имеет схожий функционал, позволяющий искать файлы в заданной папке с выбранными критериями (имя, размер, дата изменения, содержание). Несколько таких программ будет описано ниже.

CCleaner

CCleaner - утилита для компьютеров под управлением Microsoft Windows, которая очищает «мусор», накапливающийся с течением времени: временные файлы, сломанные ярлыки и другие проблемы. Одним из разделов этой программы является поиск и удаление дубликатов файлов. Интерфейс прост, понятен и удобен в использовании (рисунок 1).

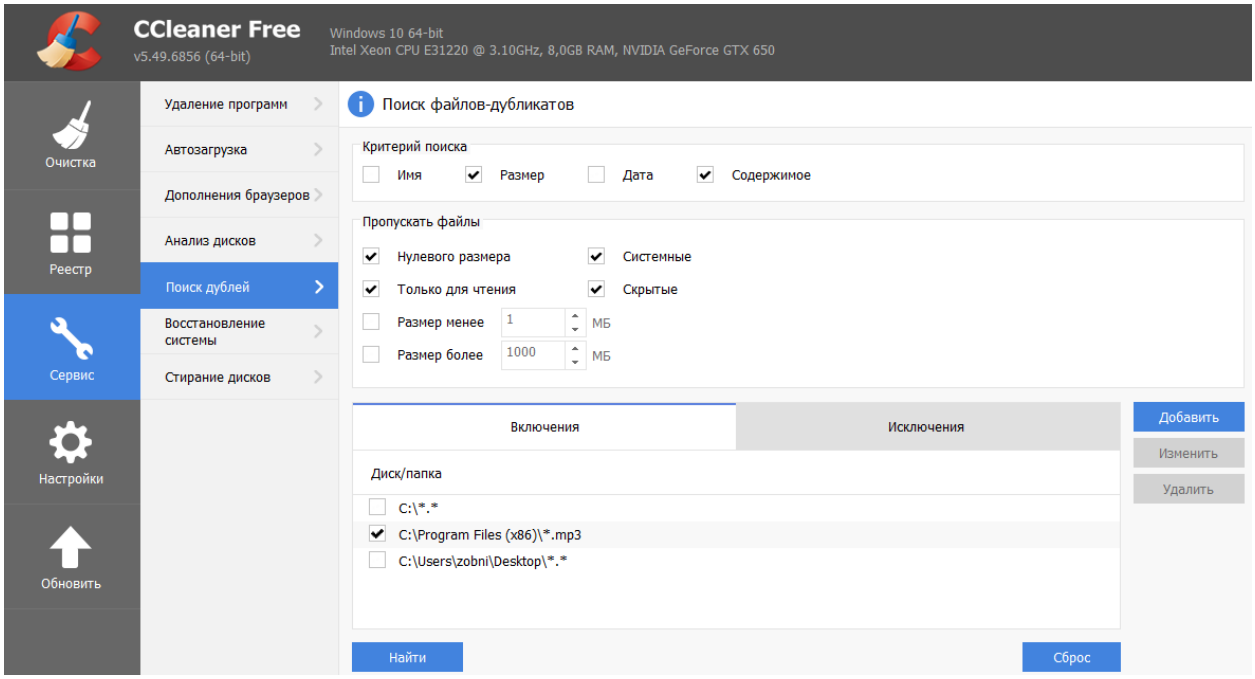


Рисунок 1 – Интерфейс раздела «Поиск дублей» утилиты CCleaner

На выбор пользователю предоставляются критерии поиска, путь, расширения файлов. Также пользователь может выбрать будет ли программа производить поиск среди файлов нулевого размера, системных файлов и т.д.

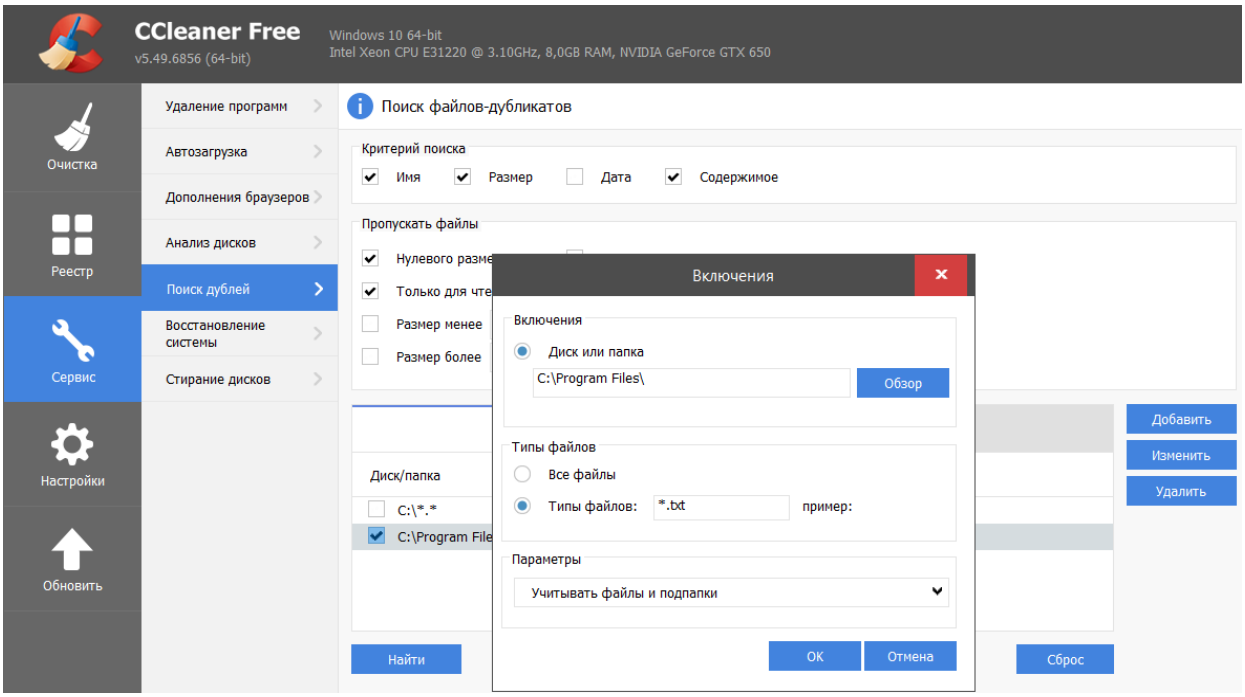


Рисунок 2 – Пример добавления типов файлов и пути для поиска дубликатов

CCleaner является примером простой программы с дружественным интерфейсом. Её функционала хватит большинству пользователей, и скорость поиска дубликатов высокая. Например, результаты, изображённые на рисунке 3 были выведены всего за 3 секунды.

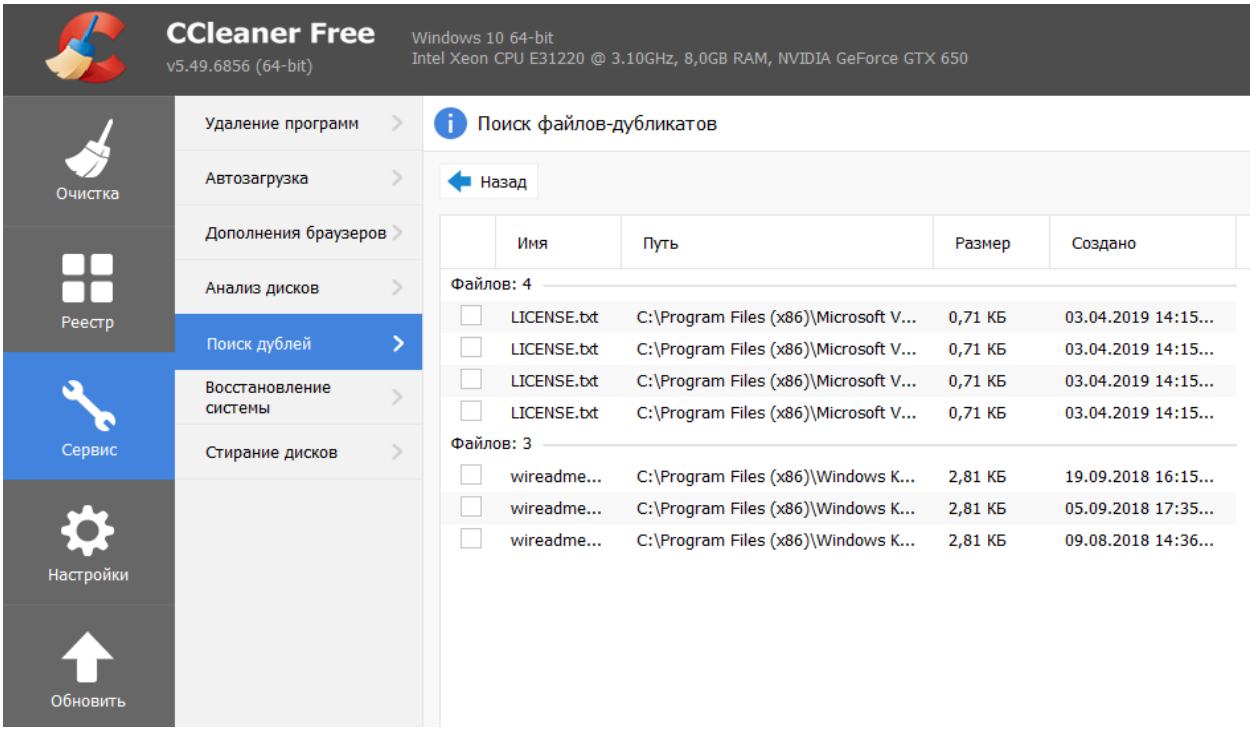
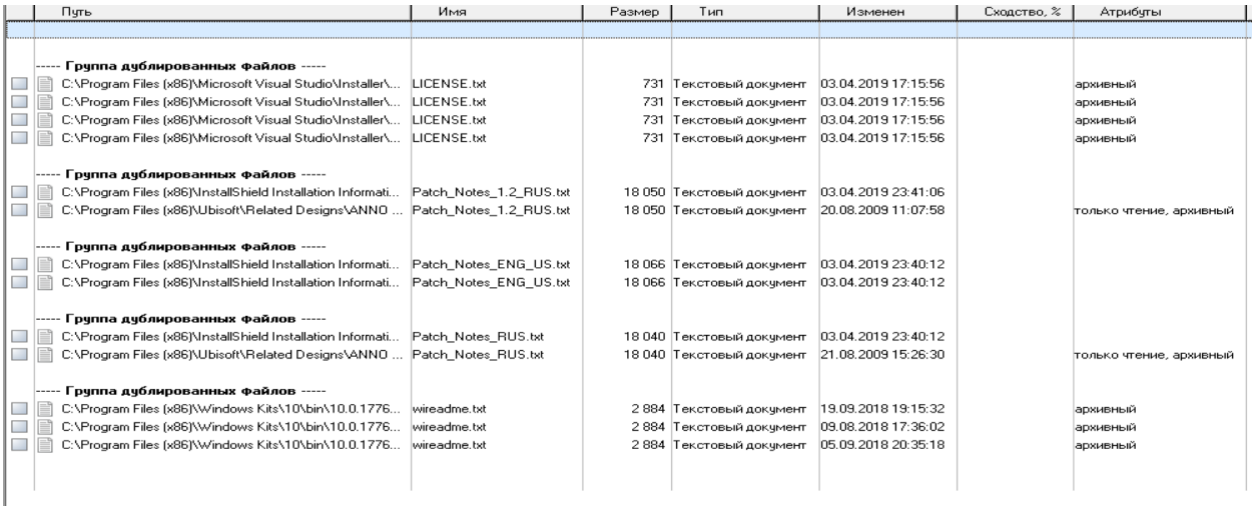


Рисунок 3 – Результаты поиска дубликатов текстовых файлов в папке “C:\Program Files (x86)\” по имени и размеру программой CCleaner

Однако за такое быстродействие и простоту приходится расплачиваться менее эффективными для пользователя результатами поиска. На рисунке 4 изображён результат работы программы DupKiller.



Путь	Имя	Размер	Тип	Изменен	Сходство, %	Атрибуты
----- Группа дублированных файлов -----						
C:\Program Files (x86)\Microsoft Visual Studio\Installer\...	LICENSE.txt	731	Текстовый документ	03.04.2019 17:15:56		архивный
C:\Program Files (x86)\Microsoft Visual Studio\Installer\...	LICENSE.txt	731	Текстовый документ	03.04.2019 17:15:56		архивный
C:\Program Files (x86)\Microsoft Visual Studio\Installer\...	LICENSE.txt	731	Текстовый документ	03.04.2019 17:15:56		архивный
C:\Program Files (x86)\Microsoft Visual Studio\Installer\...	LICENSE.txt	731	Текстовый документ	03.04.2019 17:15:56		архивный
----- Группа дублированных файлов -----						
C:\Program Files (x86)\InstallShield Installation Informati...	Patch_Notes_1.2_RUS.txt	18 050	Текстовый документ	03.04.2019 23:41:06		
C:\Program Files (x86)\Ubisoft\Related Designs\ANNO ...	Patch_Notes_1.2_RUS.txt	18 050	Текстовый документ	20.08.2009 11:07:58		только чтение, архивный
----- Группа дублированных файлов -----						
C:\Program Files (x86)\InstallShield Installation Informati...	Patch_Notes_ENG_US.txt	18 066	Текстовый документ	03.04.2019 23:40:12		
C:\Program Files (x86)\InstallShield Installation Informati...	Patch_Notes_ENG_US.txt	18 066	Текстовый документ	03.04.2019 23:40:12		
----- Группа дублированных файлов -----						
C:\Program Files (x86)\InstallShield Installation Informati...	Patch_Notes_RUS.txt	18 040	Текстовый документ	03.04.2019 23:40:12		
C:\Program Files (x86)\Ubisoft\Related Designs\ANNO ...	Patch_Notes_RUS.txt	18 040	Текстовый документ	21.08.2009 15:26:30		только чтение, архивный
----- Группа дублированных файлов -----						
C:\Program Files (x86)\Windows Kits\10\bin\10.0.1776...	wireadme.txt	2 884	Текстовый документ	19.09.2018 19:15:32		архивный
C:\Program Files (x86)\Windows Kits\10\bin\10.0.1776...	wireadme.txt	2 884	Текстовый документ	09.08.2018 17:36:02		архивный
C:\Program Files (x86)\Windows Kits\10\bin\10.0.1776...	wireadme.txt	2 884	Текстовый документ	05.09.2018 20:35:18		архивный

Рисунок 4 - Результаты поиска дубликатов текстовых файлов в папке “C:\Program Files (x86)\” по имени и размеру программой DupKiller

Как видно, утилита DupKiller смогла найти в 2 раза больше групп дублирующихся файлов. К сожалению, я не смог найти информации о том, с чем связана такая степень эффективности поиска CCleaner. Однако, возможно, это связано с некоторыми ограничениями по папкам и файлам, с которыми CCleaner в праве работать и получать информацию.

Также CCleaner не позволяет тонко настраивать параметры поиска дублей и вывода результатов. Нижеприведённый пример служит противоположностью данной утилите в этих и некоторых других аспектах.

DupKiller

DupKiller – утилита, специализирующаяся на поиске и удалении дублирующихся файлов. Интерфейс этой программы значительно более комплексный и сложный. Он позволяет установить более тонкие параметры поиска. Например, можно регулировать степень схожести имён файлов (рисунок 5),

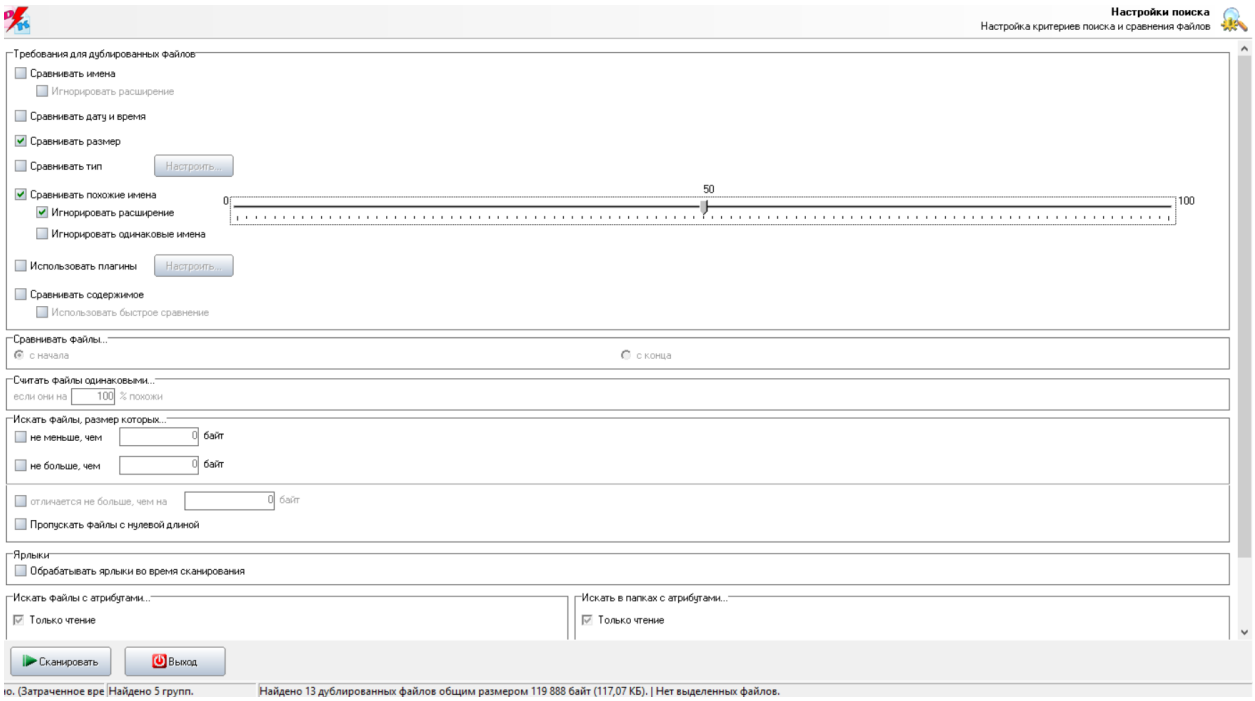


Рисунок 5 – Настройка параметров и критериев поиска

исключать некоторые папки и расширения файлов, регулировать цвета элементов списка дубликатов (рисунок 6) для более удобного просмотра, а также подключать плагины.

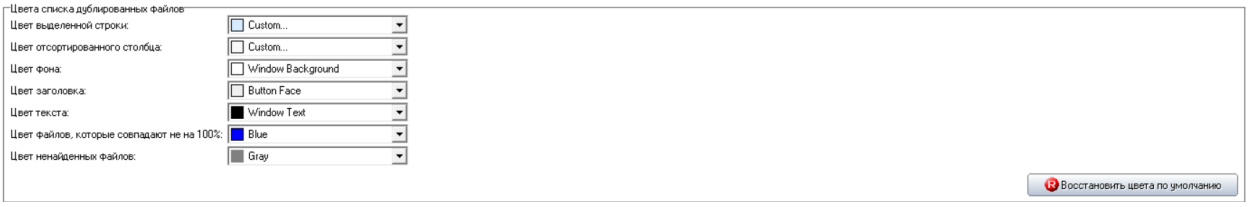


Рисунок 6 – Выбор цветов элементов списка дублитов

Время поиска относительно CCleaner при схожих параметрах в 2 раза больше – около 6 секунд. Однако результаты поиска, как было видно по рисунку 4, в два раза больше.

DupKiller поддерживает различные плагины, которые устанавливаются сразу же с программой. В настоящий момент разработчик предлагает использовать только три дополнения: ApproCom, Hearlt и Simple Image Comparer. Первый позволяет установить точный минимальный размер данных, второй позволяет проигрывать аудиофайлы по завершении поиска, а

при помощи третьего устанавливается минимальное разрешение изображений, которое будет учитываться при проверке.

DupKiller, без сомнений, превосходит по функционалу, глубине настройки и эффективности поиска предыдущий аналог. Это обусловлено тем, что данная утилита специализируется именно на поиске дубликатов, тогда как CCleaner охватывает более широкий спектр возможностей для очистки компьютера и предназначен для менее опытных пользователей, которые не хотят исследовать все тонкости настроек и интерфейса.

Таблица 1 — Сравнительная таблица систем-аналогов

Программный продукт	Простой интерфейс	Тонкая настройка	Возможность сохранения результатов	Сложность в освоении	Эффективность поиска	Время поиска
CCleaner	+	-	+	-	-	+
DupKiller	-	+	+	+	+	+

Исходя из плюсов и минусов представленных выше аналогов, было решено составить программу имеющую преимущества и избегающую недостатки обоих утилит.

1.2 Общая характеристика и особенности поиска дублирующихся файлов на диске

Поиск дубликатов требует от пользователя указания критериев поиска. Это те параметры, которые будут учитываться при сравнении файлов. Основные критерии поиска:

1. Наименование
2. Размер
3. Последняя дата изменения
4. Содержание

Также могут быть некоторые дополнительные критерии поиска, которые зависят от конкретной программы и функционала, который она предоставляет. Например, при поиске дубликатов музыкальных файлов некоторые программы дают возможность выбирать будут ли треки сравниваться по исполнителям, альбомам, жанрам, также можно указать степень схожести самих треков.

Кроме того, пользователь может выбрать типы файлов, среди которых программа будет искать дубли. Это позволяет дифференцировать нужные для пользователя файлы от тех, в поиске которых нет необходимости. Например, рядовой пользователь вряд ли захочет искать дубликаты, типы файлов которых нужны для функционирования системы. Чаще всего люди ищут дубликаты музыки, видео-файлов, картинок.

Более того, пользователь в праве указать папку, в которой будет производиться поиск дублирующихся файлов. Это сокращает время поиска и выводит только нужную пользователю информацию.

После указания всех этих параметров производится процесс поиска файлов, сохранение информации о них по группам. Параметры каждого найденного файла сравниваются с параметрами уже сохранённых, и в случае идентичности заданных пользователем критериев, они считаются дубликатами.

1.3 Обзор и обоснование выбора инструментального программного обеспечения

1.3.1 Обзор языков программирования

Для разработки программы, производящей поиск дублирующихся файлов на диске, был рассмотрен ряд языков программирования: C++, Java, Python.

Первым рассмотрим C++. Поддерживает разные парадигмы программирования, но, в сравнении с его предшественником — языком Си, — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

Таблица 2 – Плюсы и минусы C++

Плюсы	Минусы
Управление памятью; большое количество доступной литературы и документации.	Высокий порог вхождения; необходимость уделять внимание мелким деталям, из-за чего трудно реализовать некоторые простые задачи; относительно бедная стандартная библиотека.

Для Java характерен фокус на абстракциях; сильная статическая типизация; среда выполнения; ограничения на прямой доступ к памяти.

Таблица 3 — Плюсы и минусы Java

Плюсы	Минусы
Автоматическое управление памятью, многопоточность, стабильное сообщество, простой синтаксис.	Повышенное требование к оперативной памяти.

Среди характеристик Python можно отметить: сильное абстрагирование; динамическая, слабая типизация; полностью независимое управление памятью и наличие среды выполнения.

Таблица 4 — Плюсы и минусы Python

Плюсы	Минусы
Абстракции существенно облегчают выполнение даже сложных задач; большие стандартные библиотеки; простой синтаксис.	Динамическая типизация может стать источником проблем в больших проектах; низкая производительность (рисунок 8).

Название	Время выполнения (меньше — лучше)
Scala	3,1
Java	3,1
JavaScript	3,5
Go	3,7
C	3,7
JRuby	8,6
Ruby	16,5
PHP	16,5
Perl	18,0
Python	50,3

Рисунок 7 – Сводная таблица времени выполнения тестовых программ в секундах.

Тестовые программы состояли из 3 частей:

1. Загрузка данных из файла CSV.
2. Вычисления.
3. Вывод результатов.

Для данного проекта был выбрал язык C++, поскольку он обладает большим функционалом. Также я уже имею некоторый опыт разработки на этом языке с занятий в ВУЗе.

1.3.2 Обзор средств разработки

Для создания программного комплекса были рассмотрены следующие среды разработки: Visual Studio, Code::Blocks , Eclipse.

Visual Studio — среда разработки от Microsoft, известна для написания приложений, включающих в себя .NET. Это полный набор инструментов, позволяющий произвести точную отладку и настройку приложения.

Таблица 6 — Плюсы и минусы Visual Studio

Плюсы	Минусы
Поддержка других языков, таких как C#, Python, JavaScript; постоянно увеличивающаяся огромная библиотека расширений; список ошибок, упрощающих отладку.	Для работы требуются значительные ресурсы; высокий порог вхождения, без использования дополнительной литературы.

Code::Blocks — среда разработки с открытым исходным кодом. Code::Blocks также ориентирован на C и C++.

Таблица 7 — Плюсы и минусы Code::Blocks

Плюсы	Минусы
Поддержка большинства компиляторов, интегрированный список задач, отладчик, GUI и многое другое; простота интерфейса; кроссплатформенность	Редкость официальных релизов; нет подсветки ошибок в редакторе;

Eclipse – среда разработки, изначально ориентированная на работу с Java, прославилась большим количеством внешних модулей, существенно расширяющих её функциональность (в том числе, это касается количества поддерживаемых языков).

Таблица 8 — Плюсы и минусы Eclipse

Плюсы	Минусы
Множество пакетных решений, обеспечивающих многоязычную поддержку; Интерфейс, ориентированный на задачи, включая уведомления в системном трее; Автоматическое создание отчетов об ошибках.	Нехватка документации; Отсутствие единого сообщества разработчиков.

Подводя итог, далее будет представлена сравнительная таблица ключевых критериев средств разработки.

Таблица 9 — Сравнительная таблица сред разработки от 1 до 5

Критерий	Visual Studio	Eclipse	Code::Blocks
Функциональность	5	4	4
Надёжность	4	3	5
Удобство использования	4	4	4
Эффективность	5	4	4
Портативность	Средняя	Хорошая	Хорошая
Доступность	Есть бесплатная версия	Бесплатно	Бесплатно

Выбранной средой разработки стала Visual Studio, так как она многофункциональна, удобна в использовании, также у меня есть небольшой опыт работы в этой IDE.

2 ЦЕЛЬ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА

Целью данного курсового проекта является разработка программы для поиска дублирующихся по имени, размеру, времени последнего изменения или по содержанию файлов на диске. Вывод информации о путях файлов, их размере, имени, времени последнего изменения.

Для достижения поставленных целей были определены следующие задачи:

- Сравнение систем-аналогов.
- Разработка формализованного описания.
- Разработка функциональной структуры.
- Разработка блок-схем алгоритма поиска и определения дублирующихся файлов.
- Разработка UML диаграммы.
- Разработка интерфейса.
- Разработка структуры программного обеспечения.
- Определение требований к аппаратному и программному обеспечению.

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

3.1 Формализованное описание процесса поиска дублирующихся файлов на диске как объекта обработки информации

Ниже представлена диаграмма формализованного описания процесса поиска дублирующихся файлов на диске.

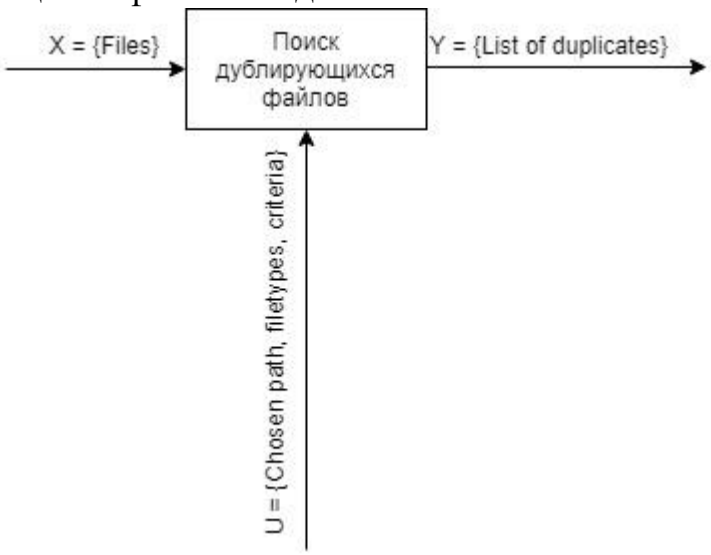


Рисунок 8 – Формализованное описание процесса поиска дублирующихся файлов на диске

Формализованное описание данного процесса состоит из одного основного блока: «Поиск дублирующихся файлов». X — вектор входных данных, {Files} — файлы, считанные с диска. Y — вектор выходных данных, {List of duplicates} — таблица дублирующихся файлов и информации о них. U — вектор управляющих воздействий, {Chosen path - заданный пользователем путь, filetypes – заданные пользователем типы файлов, criteria – заданные пользователем критерии соответствия}.

3.2 Постановка задач поиска дублирующихся файлов на диске

В процессе написания программы была поставлена задача поиска дублирующихся файлов на диске.

Необходимо в заданном пользователем каталоге и его подкаталогах произвести поиск дубликатов указанных пользователем типов файлов с заданными критериями соответствия. Для этого нужно разработать алгоритм поиска файлов в каталоге и его подкаталогах, алгоритм определения

соответствия типа файла тем, которые указал пользователь, также необходимо в процессе хэширования и сравнения информации о файлах отсеять неуказанные пользователем критерии соответствия, и не сравнивать файлы по ним.

Для сравнения файлов по указанным критериям необходимо разработать алгоритмы получения информации о размере файла, дате его последнего изменения. Также нужно получить и хэшировать наименование файла, а если файл является текстовым, то считать его содержимое, и также хэшировать его.

3.3 Функциональная структура программного комплекса для поиска дублирующихся файлов на диске

Функциональная структура программного комплекса представляет собой шесть рабочих модулей: модуль выборов параметров поиска, модуль поиска файлов, модуль хэширования заданных параметров, модуль сравнения и сохранения данных о файлах по группам, модуль формирования таблицы дубликатов, а также модуль отображения данных.

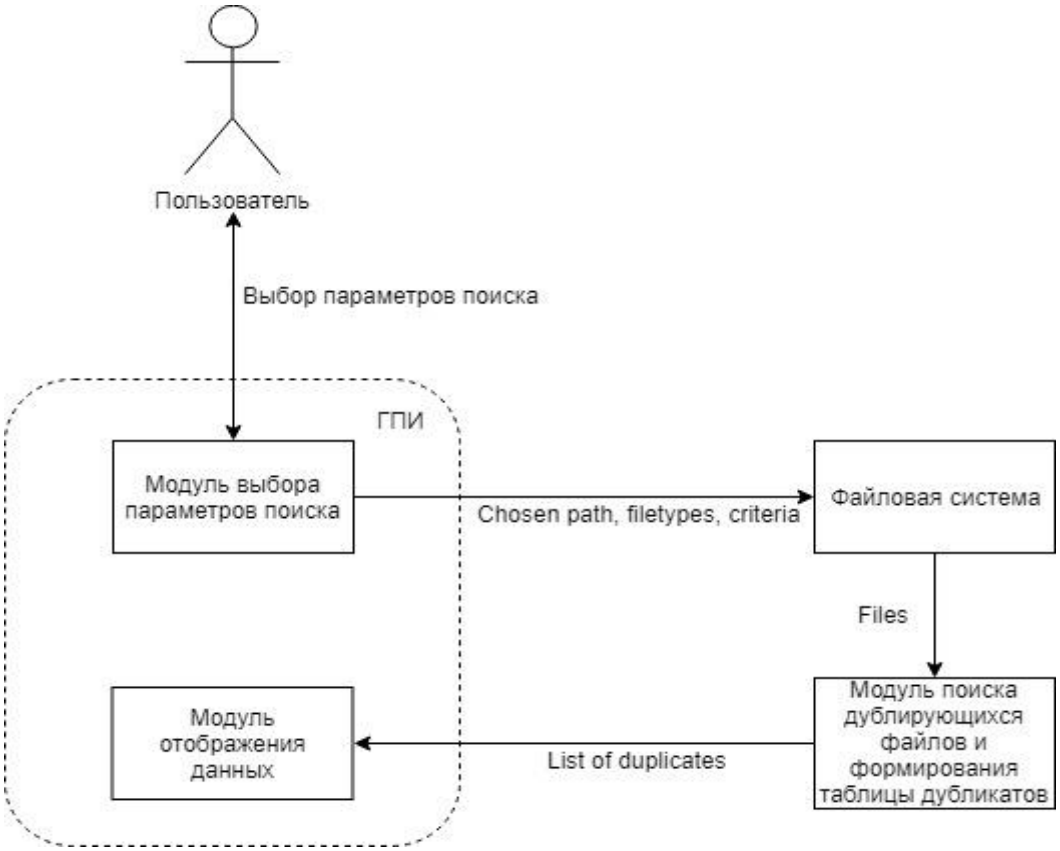


Рисунок 9 – Функциональная структура программного комплекса для поиска дублирующихся файлов на диске

Модуль выбора параметров поиска и модуль отображения данных находятся на графическом интерфейсе пользователя.

Модуль выбора параметров поиска получает выбранные пользователем путь, типы, критерии соответствия файлов и передаёт их в модуль файловой системы.

Модуль файловой системы ищет в указанном каталоге и его подкаталогах файлы и передаёт их по одному в модуль поиска дублирующихся файлов и формирования таблицы дубликатов.

Модуль поиска дублирующихся файлов и формирования таблицы дубликатов считывает данные найденного файла, хэширует и добавляет в специальный объект для сравнения нужные данные, затем сравнивает этот объект с уже сохранёнными и сохраняет в группу. По окончании сохранения данных о файле формируется таблица дубликатов.

3.4 Алгоритм определения дубликата файла

3.4.1 Алгоритм поиска файлов

Для определения дублирующихся файлов нужно сначала найти их в каталоге и подкаталогах. Для этого был создан следующий алгоритм:

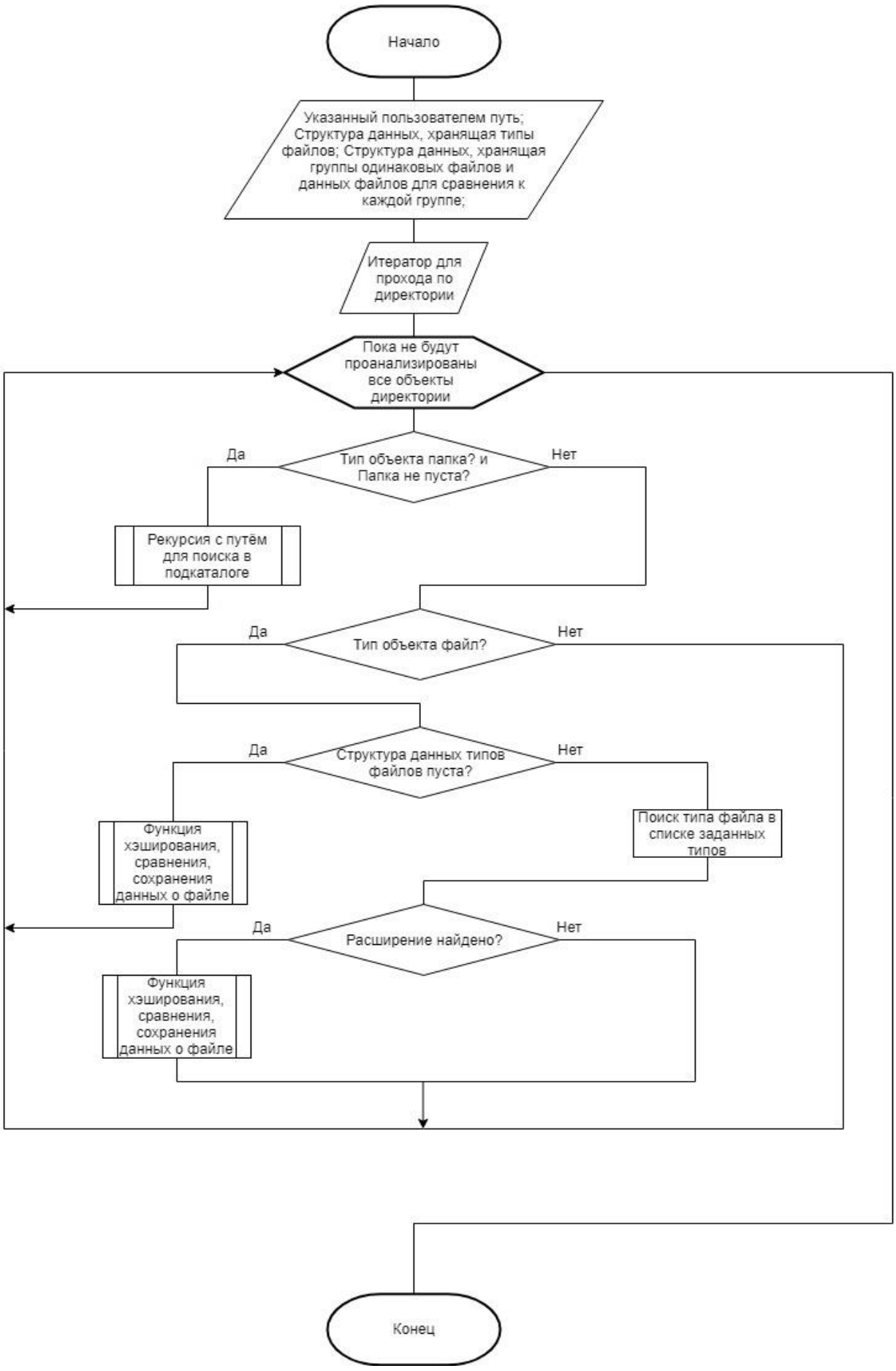


Рисунок 10 – Блок-схема алгоритма поиска файлов на диске

На этой блок-схеме (рисунок 10) представлен алгоритм поиска файлов в каталоге и подкаталогах. Сначала заводится итератор, проходящий по всем объектам заданного каталога. В случае если итератор наткнулся на папку, и она не пуста, происходит рекурсия с путём подкаталога. Если же тип объекта – файл, функция проверяет, является ли пустой структура данных, хранящая типы файлов. Если да, то производится хэширование, сравнение и сохранение данных о файле. Если нет, то проверяется, соответствует ли найденный файл хоть одному из расширений, указанным в структуре данных. Если да, то производится хэширование, сравнение и сохранение данных о файле. Иначе итератор переходит к следующему объекту каталога.

3.4.2 Алгоритм соответствия критериям

На представленной ниже блок-схеме представлен алгоритм соответствия критериям.

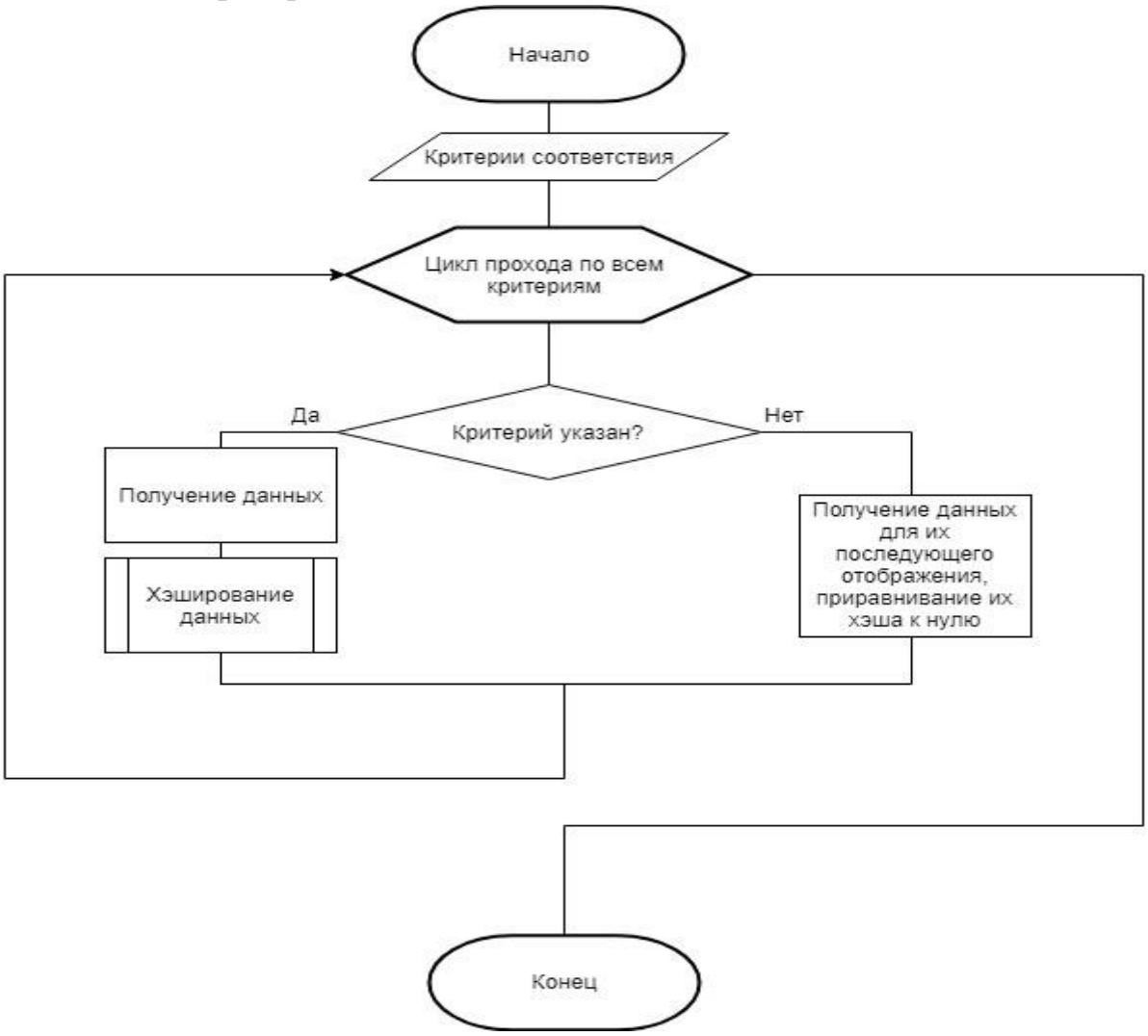


Рисунок 11 – Блок-схема алгоритма соответствия критериям

Данный алгоритм позволяет отсеивать те данные, которые не нужно хэшировать, оптимизируя тем самым работу программы. Если пользователь указал критерий соответствия, то данные, связанные с ним, будут считаны и захэшированы. В противном случае, если эти данные не будут нужны для дальнейшего отображения, то есть содержимое файла, они не будут считаны, и их хэш будет приравнен к нулю. Если это данные нужные для отображения и не нуждающиеся в хэшировании, например, размер файла, или время изменения, они будут сначала занесены в отдельную структуру данных, затем приравнены к нулю. Приравнивание этих данных, а также строк и хэшей имени и содержания файлов к нулю необходимо для того, чтобы они в дальнейшем не влияли на сравнение файлов.

3.4.3 Алгоритм хэширования

Представленная ниже блок-схема отражает алгоритм хэширования имён и содержимого файлов.

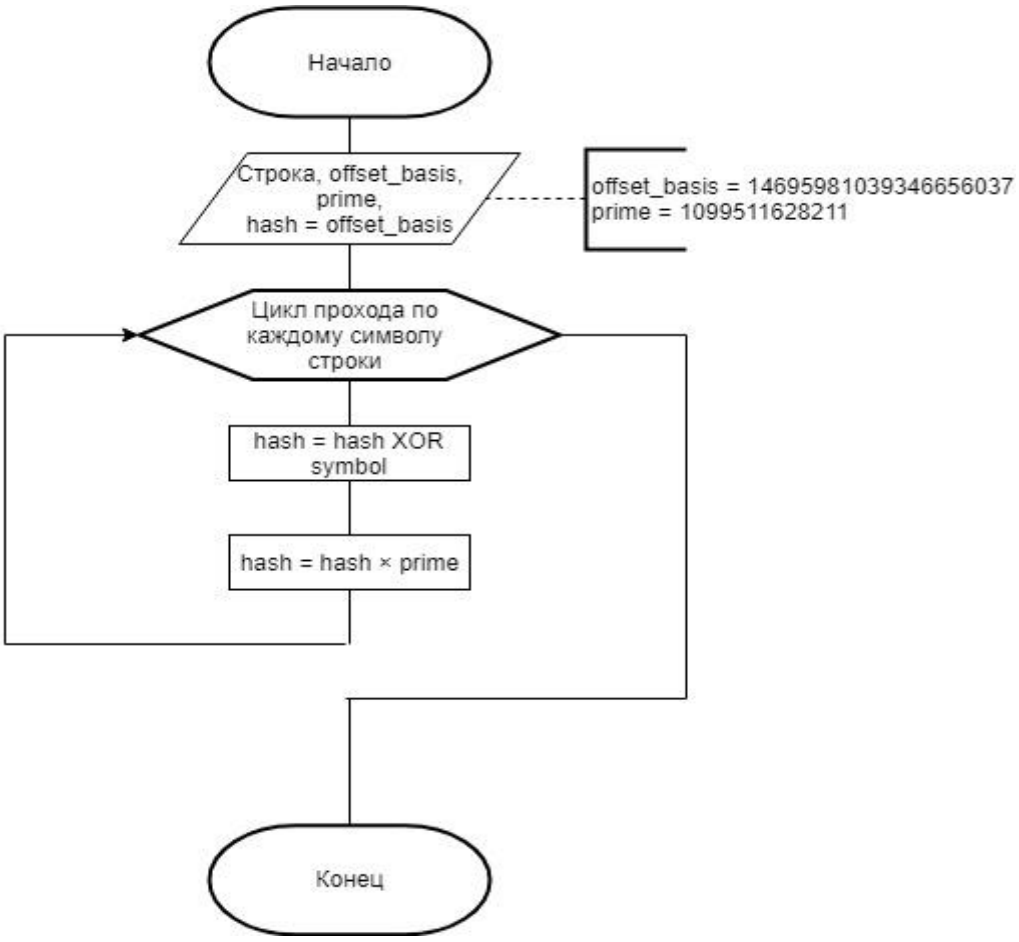


Рисунок 12 - Блок-схема алгоритма хэширования Fowler–Noll–Vo (FNV-1a)

Выбор пал именно на FNV-1a из-за простоты в понимании его работы, неплохом «лавинном эффекте» (один из критериев уникальности хэша) и из-за отсутствия необходимости писать алгоритм вручную, так как он был представлен в стандартных функциях C++.

Также представлены формулы и условия для расчёта значения «prime»:

$prime = 256^t + 2^8 + b$, где

$prime \, mod \, (2^{40} - 2^{24} - 1) > (2^{24} + 2^8 + 2^7)$;

$0 < b < 2^8$;

$t = int\left(\frac{5 + 2^S}{12}\right)$, где

$4 < S < 11$

3.4.4 Алгоритм определения дубликата файла

Ниже представлена блок-схема алгоритма определения дублирующегося файла

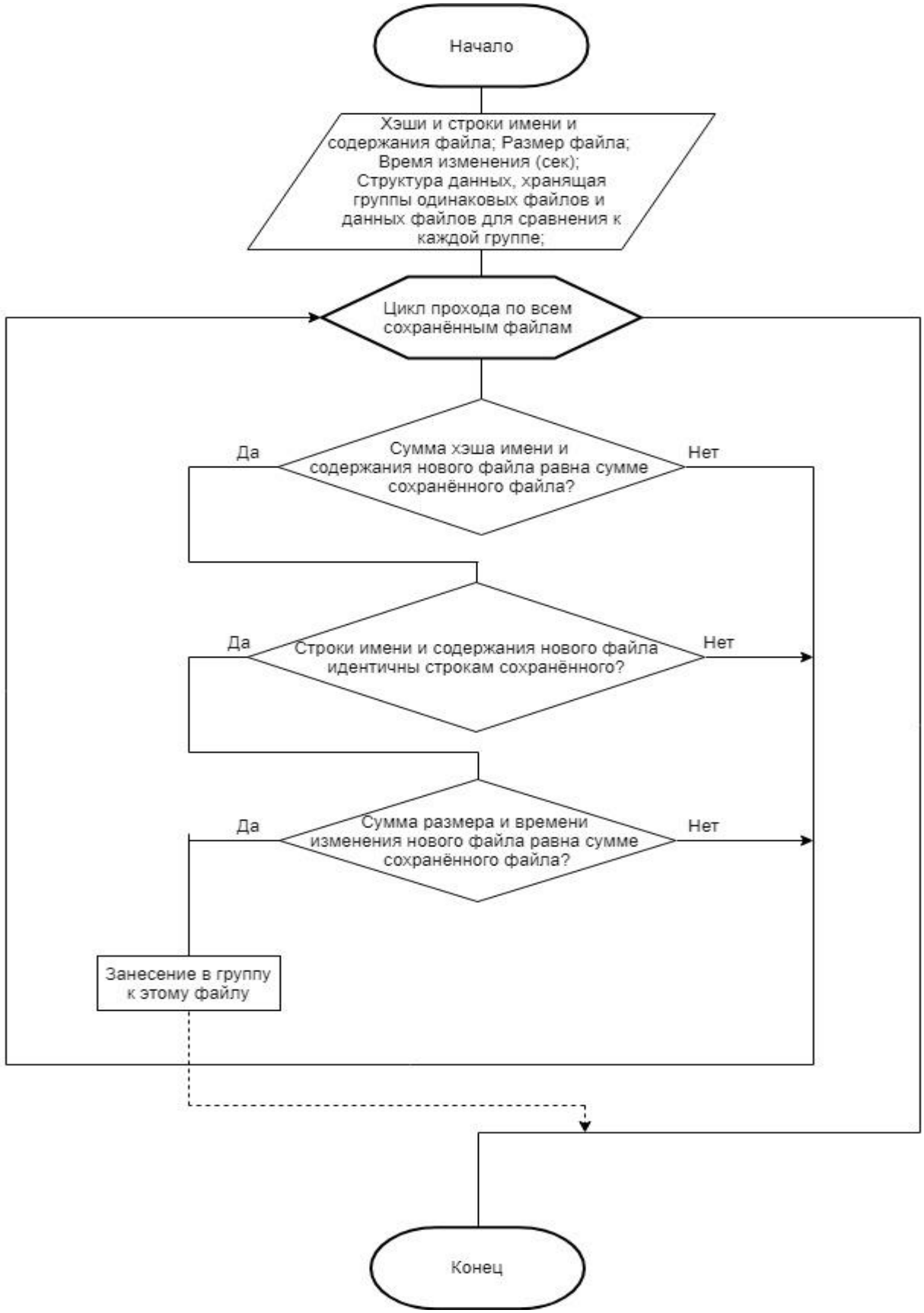


Рисунок 13 – Блок-схема алгоритма определения дубликата файла

После того, как стали известны значения размера файла и его времени изменения, а также строки и хэши имени и содержания, сравнением и сохранением данных о файле занимается вышеупомянутый алгоритм.

3.5 Разработка структуры интерфейса для пользователя

Одной из задач этого курсового проекта стала разработка интуитивно понятного и простого интерфейса, в котором пользователь без затруднений сможет найти нужные функции.

Сначала были добавлены «чекбоксы» для выбора критериев поиска (рисунок 14). Пользователь должен выбрать хотя бы один критерий, иначе будет невозможно запустить поиск.

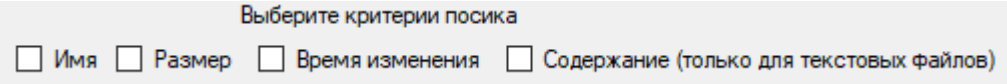


Рисунок 14 – Критерии поиска программы Duplicate Finder

Далее было решено добавить кнопки для выбора пути поиска, а также поиска и сохранения результатов в текстовый файл (рисунок 15).

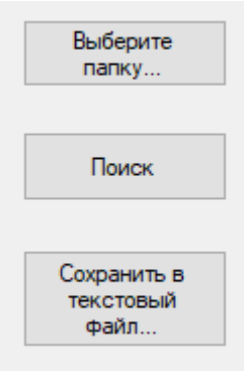


Рисунок 15 – Кнопки выбора папки для поиска дубликатов, поиска и сохранения результатов в текстовый файл

Также было решено добавить окно для введения и добавления типов файлов пользователем (рисунок 16)

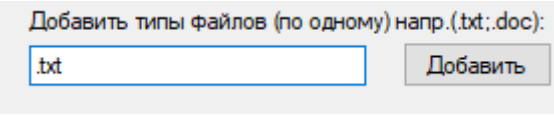


Рисунок 16 – Окно для добавления пользователем расширений файлов

и список уже добавленных, с возможностью удаления (рисунок 17). Галочка в этом списке означает, что в случае поиска дубликатов, он будет производиться только по выбранным расширениям. Также при нажатии кнопки «Удалить»

будут удалены именно выбранные расширения. В случае, если список пуст или не отмечен ни один тип файлов, поиск будет произведён по всем типам файлов.

Рисунок 17 – Список добавленных расширений

Последним важным шагом стало добавление таблицы с дубликатами, содержащую информацию о файле, то есть имя, размер, путь, время последнего изменения (рисунок 18).

Дубликаты:

№	Имя	Путь	Размер	Дата изменения
8	Microsoft.Data.E...	C:\Program Files (x86)\Microsoft Visual...	13344 Б	Tue May 7 13:00:37 2019
Группа 4				
1	s_sortedby_18.svg	C:\Program Files (x86)\Adobe\Acrobat...	783 Б	Wed May 15 22:01:22 2019
2	s_sortedby_18.svg	C:\Program Files (x86)\Adobe\Acrobat...	783 Б	Wed May 15 22:01:22 2019
Группа 5				
1	Microsoft.Data.E...	C:\Program Files (x86)\Microsoft Visual...	21536 Б	Tue May 7 13:00:37 2019
2	Microsoft.Data.E...	C:\Program Files (x86)\Microsoft Visual...	21536 Б	Tue May 7 13:00:37 2019
3	Microsoft.Data.E...	C:\Program Files (x86)\Microsoft Visual...	21536 Б	Tue May 7 13:00:37 2019
4	Microsoft.Data.E...	C:\Program Files (x86)\Microsoft Visual...	21536 Б	Tue May 7 13:00:37 2019
5	Microsoft.Data.E...	C:\Program Files (x86)\Microsoft Visual...	21536 Б	Tue May 7 13:00:37 2019
Группа 6				
1	StreamJsonRpc.dll	C:\Program Files (x86)\Microsoft Visual...	71152 Б	Thu Mar 14 22:18:44 2019
2	StreamJsonRpc.dll	C:\Program Files (x86)\Microsoft Visual...	71152 Б	Thu Mar 14 22:18:44 2019
3	StreamJsonRpc.dll	C:\Program Files (x86)\Microsoft Visual...	71152 Б	Thu Mar 14 22:18:44 2019
4	StreamJsonRpc.dll	C:\Program Files (x86)\Microsoft Visual...	71152 Б	Thu Mar 14 22:18:44 2019
Группа 7				
1	Wex.Common.dll	C:\Program Files (x86)\Windows Kits\...	693248 Б	Mon Oct 22 20:12:04 2018
2	Wex.Common.dll	C:\Program Files (x86)\Windows Kits\...	693248 Б	Mon Oct 22 20:12:04 2018
Группа 8				
1	Wex.Common.dll	C:\Program Files (x86)\Windows Kits\...	694784 Б	Mon Oct 22 20:12:04 2018
2	Wex.Common.dll	C:\Program Files (x86)\Windows Kits\...	694784 Б	Mon Oct 22 20:12:04 2018
Группа 9				

Рисунок 18 – Таблица дубликатов

В итоге получилась следующая диаграмма прецедентов использования:

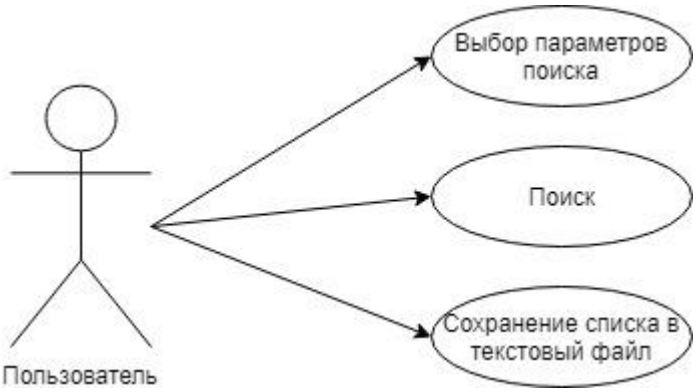


Рисунок 19 – UML-диаграмма прецедентов использования

3.6 Описание структур данных

Структур данных в этой программе я использовал две. Первая – list. Она содержит в себе тип данных string. Выполняет функцию хранения типов файлов. Вторая структура данных – map. В неё я положил две структуры как типы данных и ещё одну как «кастомный компаратор» для собственных типов

данных. Первая называется “WhatToCompare” и содержит в себе имя файла, его содержимое, размер, время изменения (в секундах), хэши содержания и имени. По этой структуре будут в дальнейшем сравниваться файлы с помощью написанного отдельно «кастомного компаратора». Вторая структура называется “File” и содержит в себе имя файла, его размер, путь и время последнего изменения (в формате день недели месяц число час:минуты:секунды год). Она служит для вывода информации о файлах.

3.7 Описание структуры программы

FieldsAndCalc – основной модуль программы, в котором содержатся функции поиска файлов, хэширования данных, сравнения информации о файлах и её сохранение.

Comparator operator() – «кастомный компаратор», сравнивающий данные нового файла с уже сохранёнными.

Hash – функция, хэширующая любую строку, будь то имя или содержание.

FileContent – функция, считывающая содержимое текстового файла.

TimeCalc – функция, получающая информацию о времени последнего изменения файла, как в секундах (для сравнений), так и в нужном для пользователя формате.

CompareAndInsert – функция, фильтрующая нужные данные о файле, вызывающая вспомогательные функции для получения этих данных.

Search – функция, ищущая файлы в каталоге и подкаталогах.

Также этот модуль содержит в себе три структуры. Первая называется “WhatToCompare” и содержит в себе имя файла, его содержимое, размер, время изменения (в секундах), хэши содержания и имени. По этой структуре будут в дальнейшем сравниваться файлы с помощью написанного отдельно «кастомного компаратора». Вторая структура называется “File” и содержит в себе имя файла, его размер, путь и время последнего изменения (в формате день недели месяц число час:минуты:секунды год). Она служит для вывода

информации о файлах. Третья структура – “Comparator” содержит лишь перегруженный оператор “()”, который служит «кастомным компаратором» для структуры данных map.

3.8 Тестирование программного комплекса

Для тестирования программного комплекса на диске C была создана папка 1 и файл 1.txt с содержанием “b” и в ней подпапка 2 с файлами 2.txt с содержанием “b” и пустым файлом 1.txt.

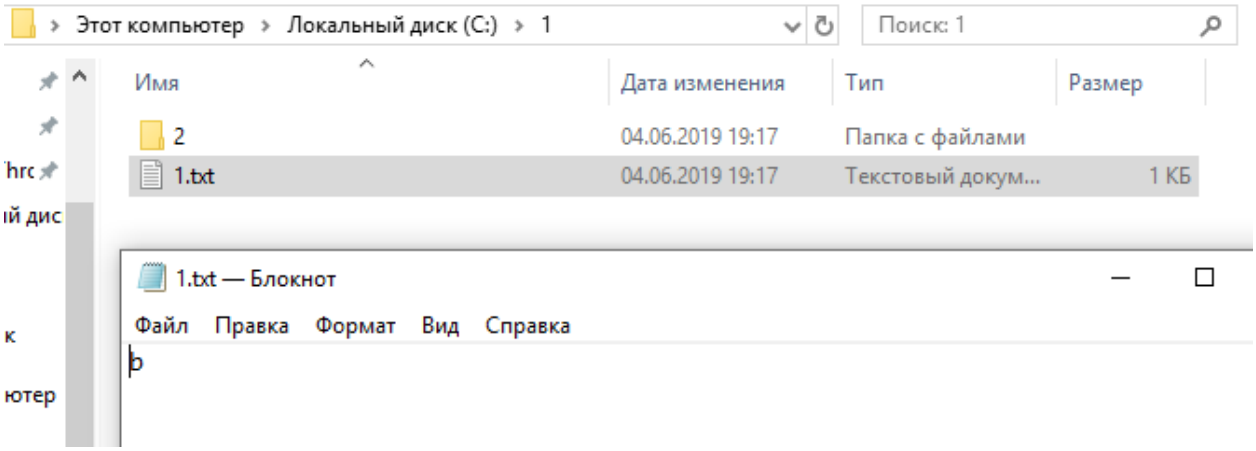


Рисунок 20 – Файл 1.txt в папке 1

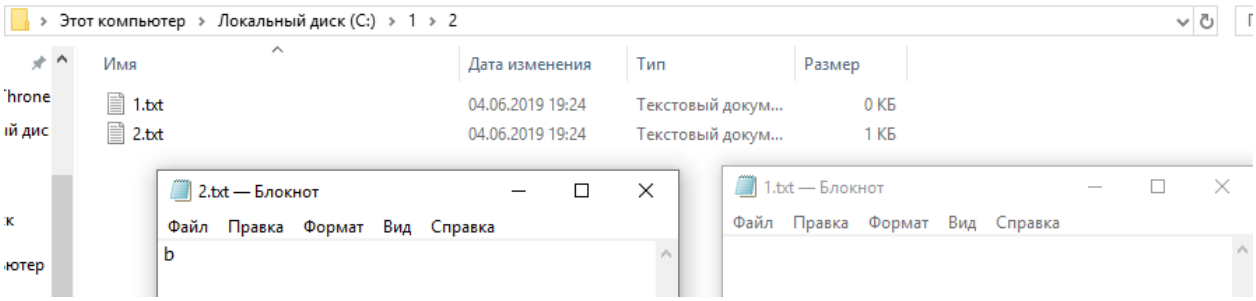


Рисунок 21 – Файлы 1.txt и 2.txt в папке 2

Далее был произведён поиск дубликатов в папке 1 и её подпапке программой Duplicate Finder с критерием поиска – имя (рисунок 22).

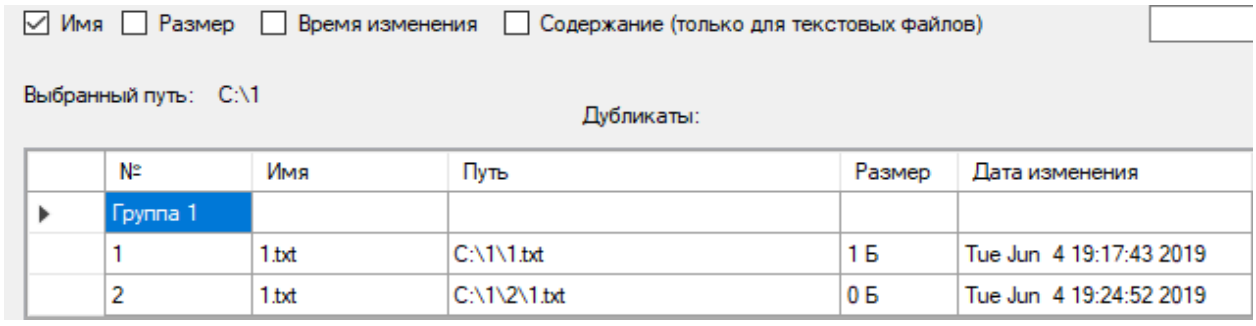


Рисунок 22 – Результаты поиска по имени

После был произведён поиск по содержанию (рисунок 23).

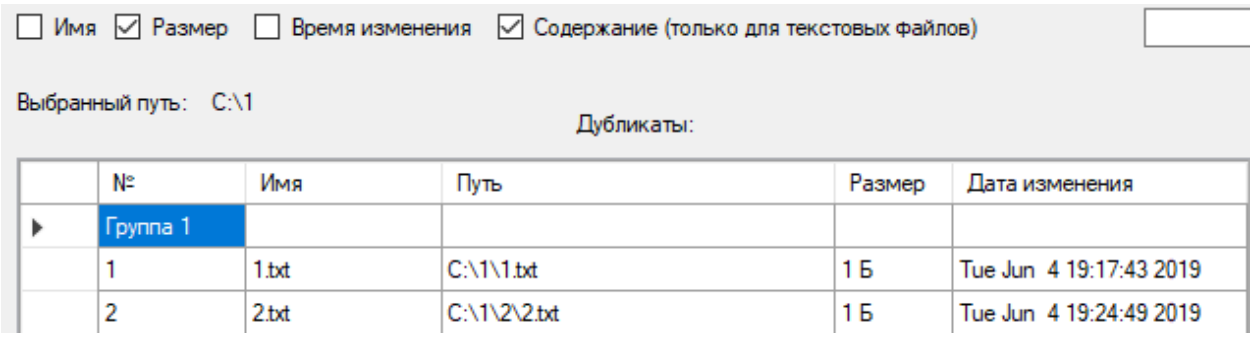


Рисунок 23 – Результаты поиска по содержанию

После этого содержание файла 1.txt в папке 2 было изменено на “b” (рисунок 24) и был произведён поиск по имени и содержанию (рисунок25).

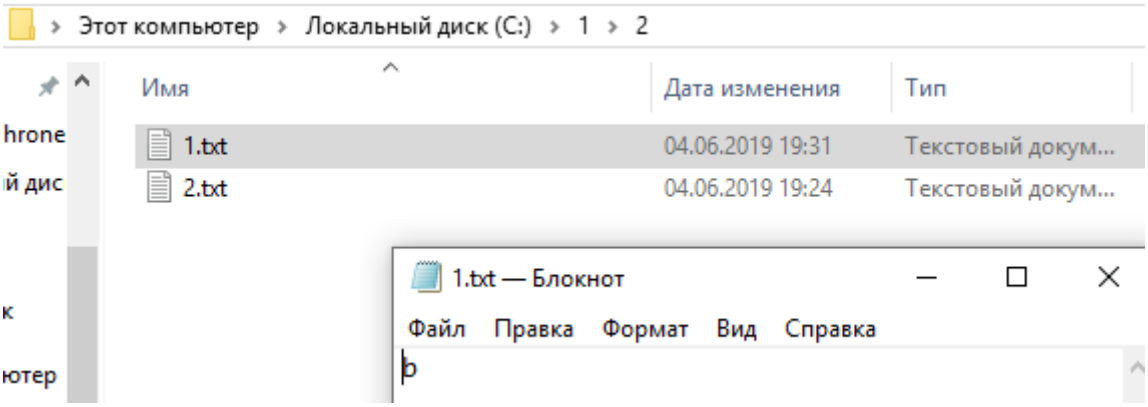


Рисунок 24 – Изменение содержания файла 1.txt

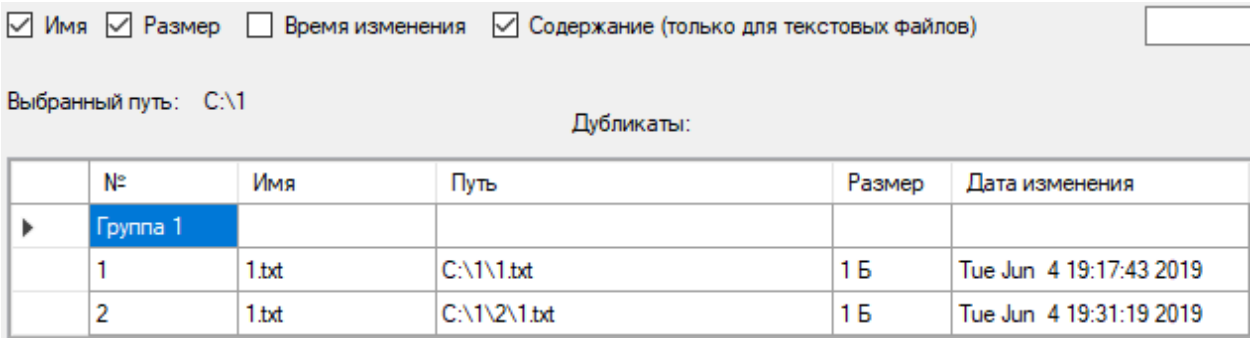


Рисунок 25 – Поиск по имени и содержанию

Также результаты последнего тестирования программы были выведены в текстовый файл:

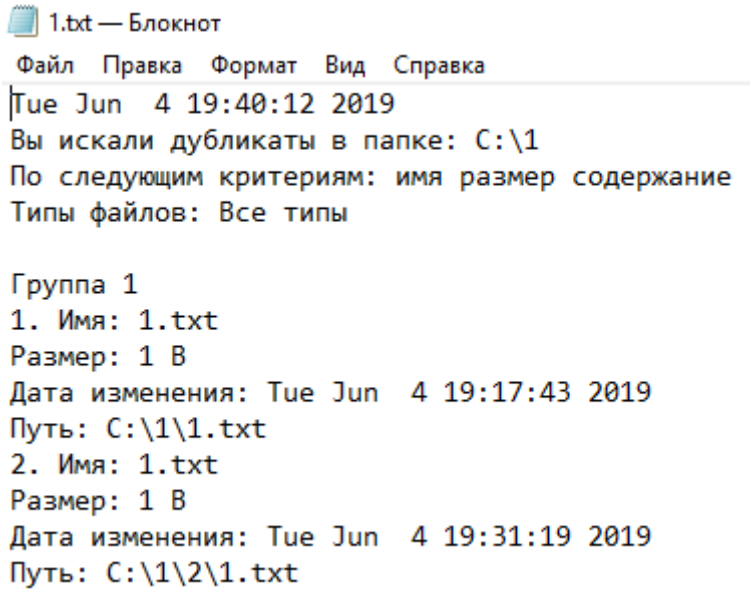


Рисунок 26 – Результаты, сохранённые в текстовый файл

3.9 Характеристика программного и аппаратного обеспечения

Программный комплекс разработан под управлением ОС Windows 10 на платформе .Net Framework 4.6.1. Средой разработки является Microsoft Visual Studio 2017.

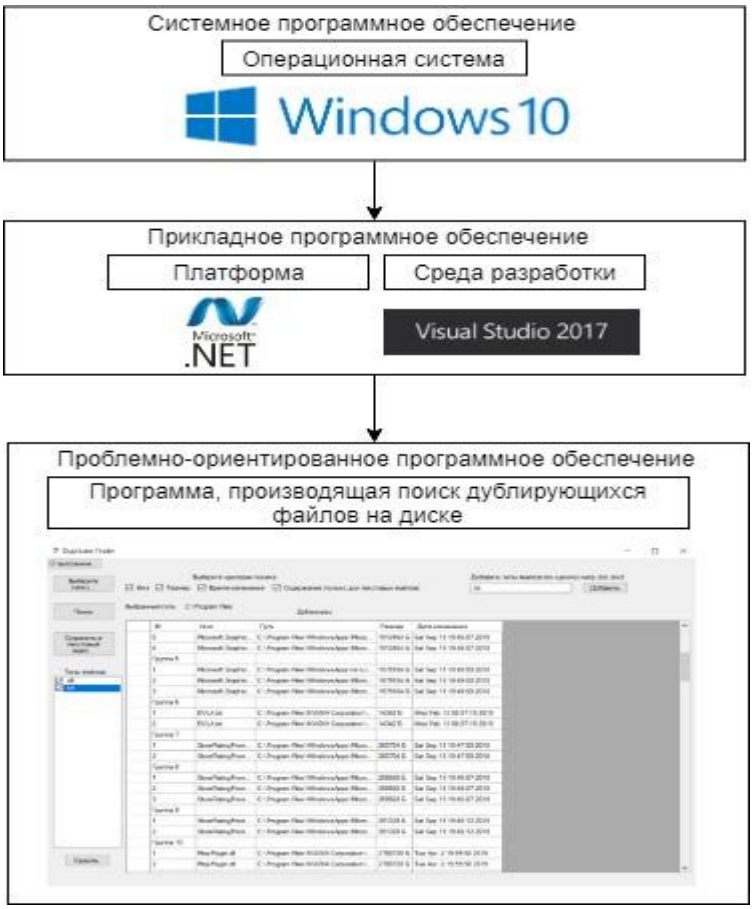


Рисунок 27 – Структура программного обеспечения

Таблица 10 – Минимальные системные требования

Показатель	Значение
Тип ЭВМ	Персональный компьютер
Тактовая частота процессора, ГГц	2,5
Объём оперативной памяти, Гб	8
Объём внешней памяти, Гб	128
Состав и характеристика периферийных устройств ЭВМ	Клавиатура, мышь, монитор с диагональю 24″ и разрешением 1920×1080 точек
Операционная система	Windows 10
Прикладное программное обеспечение	.NET Framework 4.6.1

Таблица 11 – Характеристика программного обеспечения

Показатель	Значение
Среда разработки	Microsoft Visual Studio 2017
Технология программирования	ООП
Язык программирования	C++
Количество входных переменных	7
Количество выходных переменных	В зависимости от результатов поиска
Количество классов	3
Количество функций	26
Размер исполняемого файла, Кб	1229
Время обработки данных и визуализации результатов	В зависимости от количества файлов

ВЫВОДЫ

Результатом проделанной работы стал программный комплекс, производящий поиск дублирующихся файлов по указанному пути, с указанными критериями поиска и типами файлов с возможностью просмотра и сохранения результатов в текстовый файл.

В ходе разработки данного программного комплекса были выполнены следующие задачи:

- Был произведен обзор языков программирования, в результате которого был выбран такой язык, как C++.
- Был произведен обзор сред разработки, подходящих для данного языка, в результате которого в качестве среды разработки была выбрана Visual Studio.
- Было разработано формализованное описание данного процесса, состоящее из одного основного блока: «Поиск дублирующихся файлов». Входными данными являются заданные пользователем путь, типы файлов, критерии соответствия. Выходные данные - таблица дублирующихся файлов и информации о них.
- Была разработана функциональная структура программного комплекса, представляющая собой шесть рабочих модулей: модуль выборов параметров поиска, модуль поиска файлов, модуль хэширования заданных параметров, модуль сравнения и сохранения данных о файлах по группам, модуль формирования таблицы дубликатов, а также модуль отображения данных.
- Была разработана UML-диаграмма прецедентов использования.
- Были построены блок-схемы алгоритма поиска и определения дублирующихся файлов.
- Было проведено тестирование программного комплекса, в ходе которого была продемонстрирована работа программы с критериями поиска.

В будущем хотелось бы улучшить разработанную программу, а именно:

- Ускорить работу программы.
- Добавить возможность удаления дубликатов напрямую из программы.
- Добавить возможность предпросмотра дубликатов файлов.
- Сделать интерфейс более эстетически приятным.
- Добавить индикаторы прогресса процесса поиска дубликатов файлов.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Страуструп, Б. Programming: Principles and Practice Using C++ / Б. Страуструп. – М. : Вильямс, 2011. – 1248 с.

2. Норенков, И.П. Основы автоматизированного проектирования / И.П. Норенков. – М. : МГТУ им. Н. Э. Баумана, 2009. – 430 с.

3. Рождественский, Д. А. Автоматизация проектирования систем и средств управления: Т. 1: учеб. пособие / Д. А. Рождественский – Томск: Том. межвуз. Центр дистанц. Образования, 2004. – 167 с.

4. Вильямс, Я.Ш. C++: базовый курс / Я.Ш. Вильямс. – М. : Вильямс, 2014. – 624 с.

5. Липпман, С. Основы программирования на C++ / С. Липпман. – М. : Вильямс, 2014. – 1104 с.

6. Олифер, Н. А. Сетевые операционные системы / Н. А. Олифер В. Г. Олифер – СПб: Питер, 2009. – 669 с.

7. Чистякова, Т. Б. Программирование на языках высокого уровня. Базовый курс: учеб. пособие / Т. Б. Чистякова, Р. В. Антипин, И.В. Новожилова. – СПб. : СПбГТИ(ТУ), 2008. – 101 с.