

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Сибирский государственный университет науки и технологий
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

КУРСОВАЯ РАБОТА

Объектно-ориентированное программирование

Архиватор

Руководитель

подпись, дата

В.В. Вдовенко

инициалы, фамилия

Обучающийся БПИ 22-02, 221219040

номер группы, зачетной книжки

подпись, дата

К.В. Трифонов

инициалы, фамилия

Красноярск 2024 г.

Задание на курсовую работу

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

ЗАДАНИЕ

на курсовую работу по дисциплине Объектно-ориентированное программирование

Обучающемуся Трифонову Кириллу Вячеславовичу

Группа БПИ22-02 Форма обучения очная

Тема работы: Архиватор

Срок сдачи курсовой работы июнь 2024 г.

Перечень вопросов, подлежащих разработке при написании теоретической части:

1. Системный анализ предметной области.
2. Объектно-ориентированное моделирование программы:
 - 2.1. Разработка UML-диаграммы прецедентов
 - 2.2. Разработка UML-диаграммы активности
3. Разработка архитектуры программы

Перечень вопросов, подлежащих разработке при написании практической части:

1. Разработка кода программы
2. Тестирование программы
3. Разработка руководства программиста
4. Разработка руководства пользователя

Дата выдачи задания: 14 февраля 2024 г.

Руководитель Вдовенко В.В., доцент кафедры ИВТ

(подпись)

Задание принял к исполнению (дата) _____

(подпись обучающегося, И.О. Фамилия)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	5
1.1 Изучение особенностей архивации файлов	5
1.2 Обзор существующих программных продуктов для архивирования файлов.....	5
1.3 Разработка функциональных требований	9
1.4 Выводы по главе.....	9
2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА.....	11
2.1 Динамические аспекты поведения системы	11
2.2 Объектно-ориентированная структура программы.....	12
2.3 Руководство программиста.....	13
2.4 Тестирование программного продукта	14
2.5 Руководство пользователя	15
2.6 Выводы по главе.....	18
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	20
ПРИЛОЖЕНИЕ А	21

ВВЕДЕНИЕ

С развитием цифровых технологий и распространением интернета в современном мире растет потребность в эффективном управлении файловыми ресурсами. Важными инструментами становятся программы-архиваторы, предоставляющие пользователям возможность сжатия, распаковки и управления файлами в различных форматах архивов.

Цель данной курсовой работы заключается в разработке программы-архиватора, обеспечивающей удобный интерфейс для сжатия и распаковки файлов, выбора методов сжатия, а также эффективного управления архивами. Программа-архиватор должна обеспечивать надежное хранение данных, экономить место на диске и упрощать передачу файлов через сеть.

Для достижения поставленной цели необходимо решить следующие *задачи*:

- Изучить существующие аналоги программ-архиваторов и анализировать их функциональные возможности;
- Определить требования к функционалу программы-архиватора, учитывая разнообразие форматов архивации и методов сжатия;
- Осуществить проектирование и программную реализацию программы-архиватора с учетом современных технологий и стандартов безопасности;
- Провести тестирование разработанного программного продукта для обеспечения его надежной работы и соответствия требованиям пользователей.

Пояснительная записка к курсовой работе состоит из введения, двух глав, заключения и списка использованных источников из 5 наименований. Изложена на 21 странице и содержит 9 рисунков и 2 таблицы.

В первой главе курсовой работы рассматриваются особенности файловых операций, связанных с архивированием и сжатием данных, а также проводится анализ существующих программ-архиваторов.

Во второй главе подробно рассматриваются технические аспекты создания программы-архиватора, включая алгоритмы сжатия, структуру программного кода, тестирование и руководство пользователя.

В заключении подводятся итоги всей курсовой работы, а также предлагаются возможные улучшения и перспективы развития приложения.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Анализ специфики предметной области в рамках разработки приложения-архиватора предполагает исследование методов сжатия.

В рамках анализа предметной области также важно провести оценку требований пользователей к функционалу приложения и изучить существующие аналоги приложений-архиваторов.

1.1 Изучение особенностей архивации файлов

В этом подразделе рассматриваются основные операции с файлами в программе архиваторе.

Сжатие данных — это процесс уменьшения размера файла или потока данных путем удаления избыточной информации или использования более эффективного представления данных. Цель сжатия данных состоит в том, чтобы уменьшить объем хранимой или передаваемой информации, сэкономить пропускную способность сети или место на диске, а также улучшить скорость передачи данных.

Сжатие данных может быть осуществлено различными алгоритмами, включая алгоритмы Хаффмана, *LZ77* (алгоритм Лемпеля-Зива), алгоритм *DEFLATE* и многие другие. Каждый алгоритм имеет свои особенности и подходит для различных типов данных.

Распаковка данных — это процесс восстановления исходных данных из сжатого файла или потока данных. Когда данные сжимаются с использованием алгоритма сжатия, они становятся менее объемными, но при этом остаются в сжатом, неполном или кодированном формате. Распаковка выполняет обратную операцию сжатия, преобразуя сжатые данные обратно в их исходное состояние.

Важно отметить, что при распаковке данных происходит восстановление информации без потерь, если используется без потерь метод сжатия. Это означает, что распакованные данные полностью идентичны исходным данным перед сжатием.

Также следует учитывать возможность работы с различными форматами архивов, такими как *ZIP*, *RAR*, *7z* и другими, чтобы обеспечить совместимость с различными операционными системами и программами архивации.

Для архиватора также важно учитывать интерфейс пользователя, возможность выбора метода сжатия, уровня сжатия и других параметров. Это позволяет пользователю настраивать архиватор под свои нужды и условия.

1.2 Обзор существующих программных продуктов для архивирования файлов

Существует множество программных продуктов для архивирования файлов, каждый из которых имеет свои особенности и преимущества. Один из наиболее популярных архиваторов — *WinRAR*.

WinRAR

Программа предназначена для операционных систем Windows, в том числе для 32- и 64-разрядных версий, а также имеет версии для *Android*, *Linux*, *FreeBSD*, *macOS*, *MS-DOS* и *Windows Mobile*. Он позволяет создавать, изменять и распаковывать архивы форматов *RAR* и *ZIP*, а также работать с множеством других форматов архивов. В версии 5 и выше *WinRAR* добавил поддержку нового формата *RAR5*, который включает увеличение размера словаря до 64 Гб, использование алгоритма шифрования *AES-256* и алгоритма хеширования *BLAKE2* для контроля целостности архива. Кроме того, в *RAR5* улучшены возможности работы с путями файлов и добавлены средства для ускорения открытия больших архивов [1]. Окно программы представлено на рисунке 1.

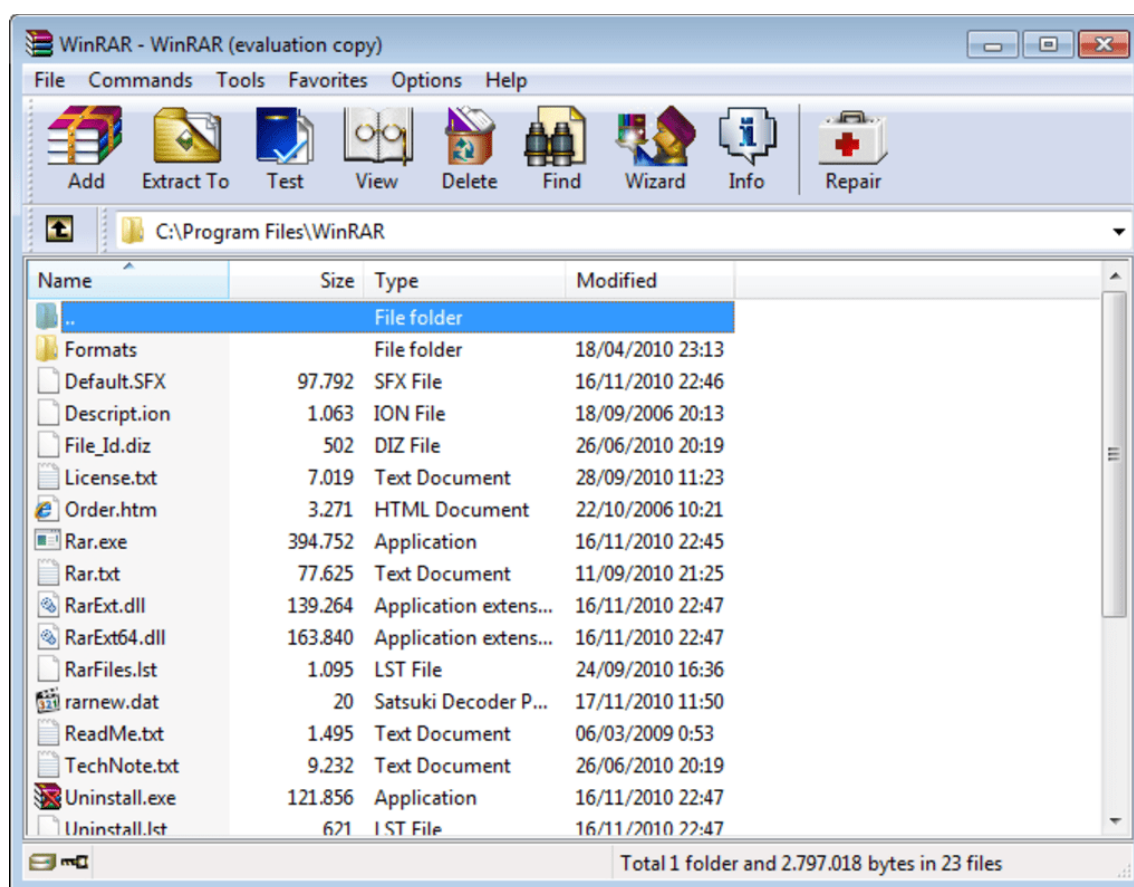


Рисунок 1 – Окно программы «WinRAR»

Достоинства программы:

- Высокая степень сжатия файлов;
- Широкая поддержка форматов архивов;
- Интуитивно понятный интерфейс;
- Возможность установки паролей для защиты архивов.

Недостатки:

- Коммерческое программное обеспечение, требует лицензионного ключа для полноценного использования;
- Устаревший дизайн.

7-Zip

Архиватор является свободным и имеет высокую степень сжатия данных. Он поддерживает различные алгоритмы сжатия и множество форматов данных, включая собственный формат 7z с эффективным алгоритмом сжатия *LZMA*. Разработка программы началась в 1999 году, она бесплатна и имеет открытый исходный код, за исключением некоторых частей, подпадающих под ограничения лицензии *GNU LGPL* [2]. Рабочее окно программы представлено на рисунке 2.

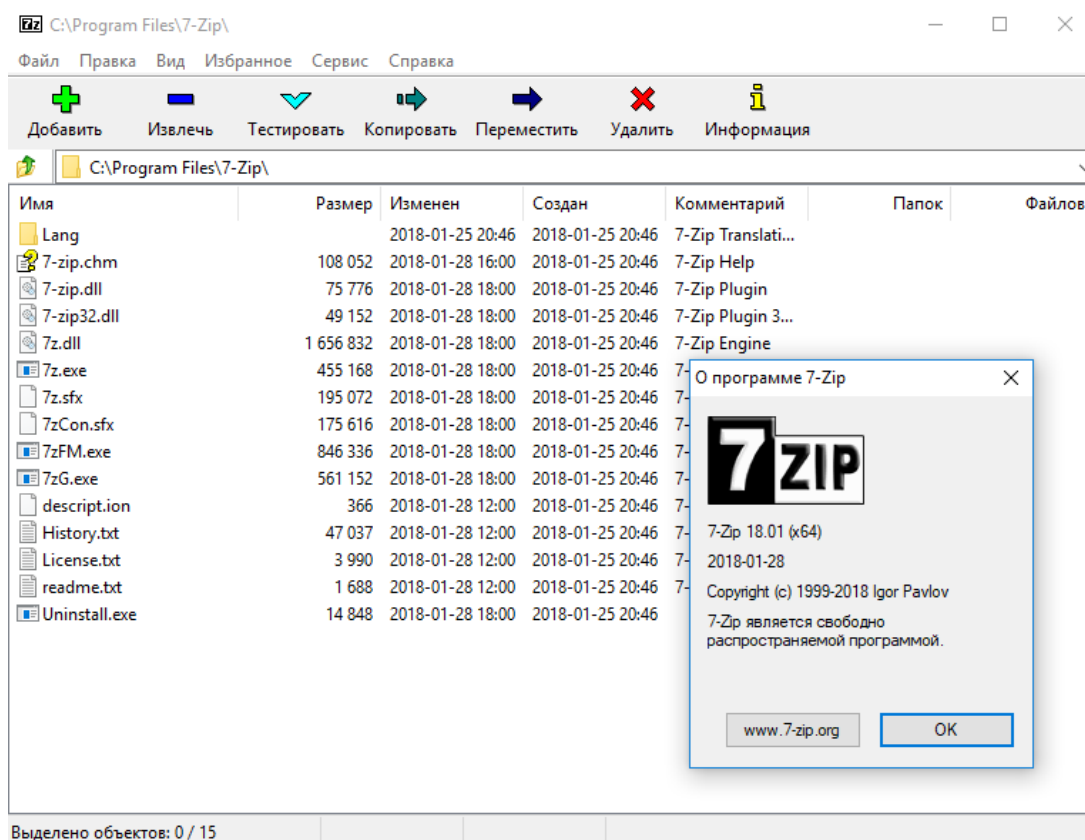


Рисунок 2 – Окно программы «7-zip»

Достоинства программы:

- Бесплатное и открытое программное обеспечение;
- Высокая степень сжатия;
- Поддержка большого количества форматов архивов;

Недостатки:

- Интерфейс может показаться менее привлекательным по сравнению с коммерческими аналогами;
- Некоторые пользователи отмечают, что скорость работы программы не всегда высока.

PeaZip

Представляет из себя свободное и бесплатное ПО, имеет кроссплатформенную поддержку и представляет собой графическую оболочку для других архиваторов. Он написан на *Free Pascal* и собран в *Lazarus*. *PeaZip* доступен для *Windows 9x*, *Windows NT* (включая операционную систему *ReactOS*) и *Linux* в виде установочных пакетов и портативных версий, которые не вносят изменений в операционную систему. Программа поддерживает собственный формат архивов *Pea*, а также другие форматы, используя внешние программы и библиотеки [3]. Интерфейс программы представлен на рисунке 3.

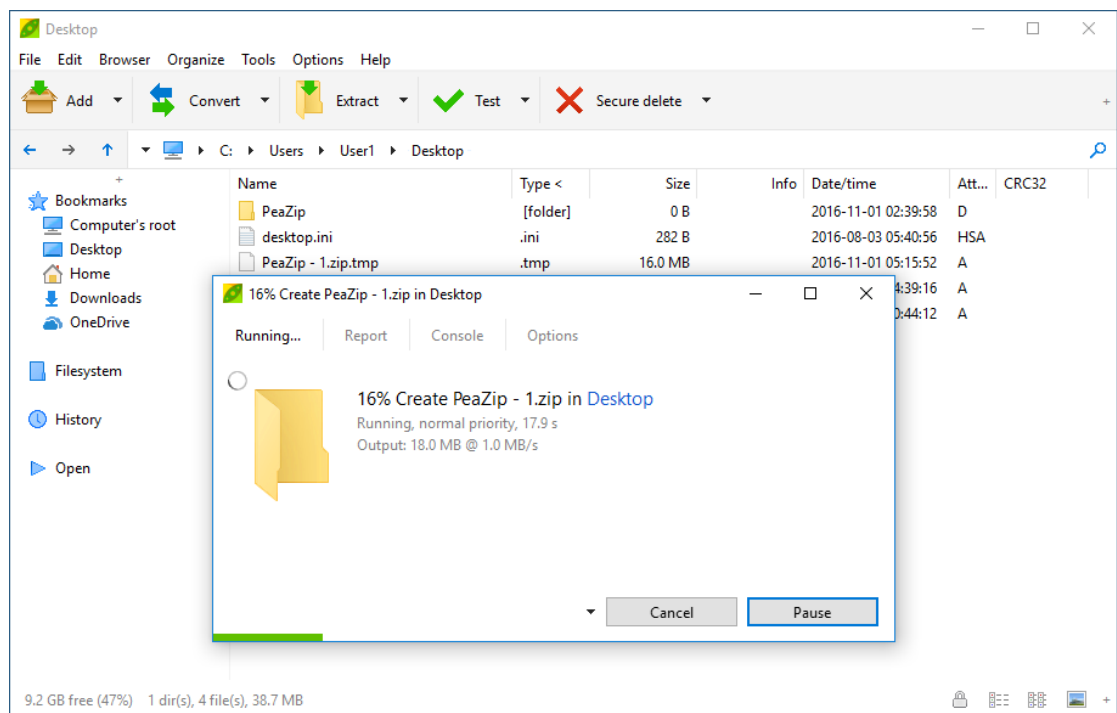


Рисунок 3 – Интерфейс программы «PeaZip»

Достоинства программы:

- Бесплатное и открытое программное обеспечение;
- Широкая поддержка форматов архивов;
- Интуитивно понятный интерфейс с современным дизайном (в стиле встроенного проводника *Windows*);
- Мощные возможности сжатия и шифрования.

Недостатки:

- Не так широко распространено и не получило такого широкого признания, как *WinRAR* или *7-Zip*.

Результаты сравнения рассмотренных программных продуктов представлены в таблице 1.

Таблица 1 – Сравнение программных продуктов

Характеристики	«WinRAR»	«7-Zip»	«PeaZip»
Поддержка форматов	RAR, ZIP, CAB, ARJ, и др.	7z, ZIP, TAR, и др.	7z, ZIP, TAR, и др.
Операционная система	Windows, MacOS, Linux	Windows, Linux	Windows, Linux, BSD
Интерфейс	Графический и командная строка	Командная строка	Графический и командная строка
Бесплатность	Только пробная версия	Полностью бесплатная	Полностью бесплатная
Ресурсоемкость	Выше среднего	Ниже среднего	Ниже среднего
Поддержка архивации в облаке	Да	Нет	Да
Поддержка архивации на FTP/SFTP	Да	Нет	Да
Поддержка альтернативных кодировок	Да	Нет	Да

1.3 Разработка функциональных требований

Функциональные требования к приложению описаны на диаграмме прецедентов на рисунке 4.

Пользователь должен иметь возможность:

1. Выбирать файлы для выполнения операций;
2. Перемещаться по файлам во встроенном проводнике
3. Сжимать файлы;
4. Распаковывать файлы.

1.4 Выводы по главе

В первой главе была изучена теоретическая база, представляющая основные понятия, принципы и особенности различных программ-архиваторов. Также были рассмотрены несколько существующих аналогов: WinRar, который является ограниченно бесплатным, 7-zip с устаревшим интерфейсом, а также PeaZip.

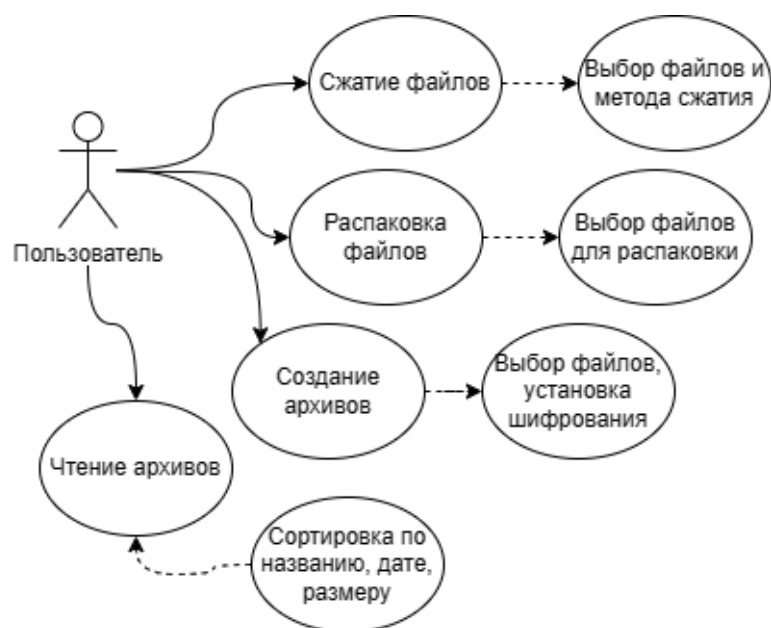


Рисунок 4 – Диаграмма прецедентов

В связи с этими недостатками рассмотренных программных продуктов была проведена оценка функциональных требований для полностью бесплатного Архиватора с интуитивно понятным графическим интерфейсом. Следующая глава посвящена непосредственно техническим аспектам разработки данного приложения.

2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

Для разработки приложения «Архиватор» был выбран язык программирования *Python*, как один из самых распространённых и универсальных языков. Также он имеет удобные встроенные библиотеки для создания графического интерфейса, такие как *PyQt5*, которая была использована для создания интерфейса в разработанном приложении. Также язык *Python* предоставляет множество модулей для работы с архивами, в частности *zipfile*.

Библиотека *zipfile* предоставляет множество алгоритмов сжатия, такие как *DEFLATE*, *BZip2*, *LZMA*. Выбор был остановлен на алгоритме *DEFLATE* поскольку он входит в число стандартных методов для работы с модулем *zipfile* [5].

В данной главе рассмотрены технические стороны создания данного приложения: динамические аспекты поведения системы, объектно-ориентированная структура программы, руководство программиста, тестирование программного продукта и руководство пользователя.

2.1 Динамические аспекты поведения системы

Установление динамических аспектов поведения системы позволяет определить, какие данные, объекты и методы будут существовать и как они будут взаимодействовать друг с другом и с другими структурами системы. В дальнейшем это позволяет определить и разработать объектно-ориентированную архитектуру системы. Аспекты поведения описывают диаграммы активности.

На рисунке 5 представлена диаграмма активности, отображающая процессы сжатия и распаковки данных.

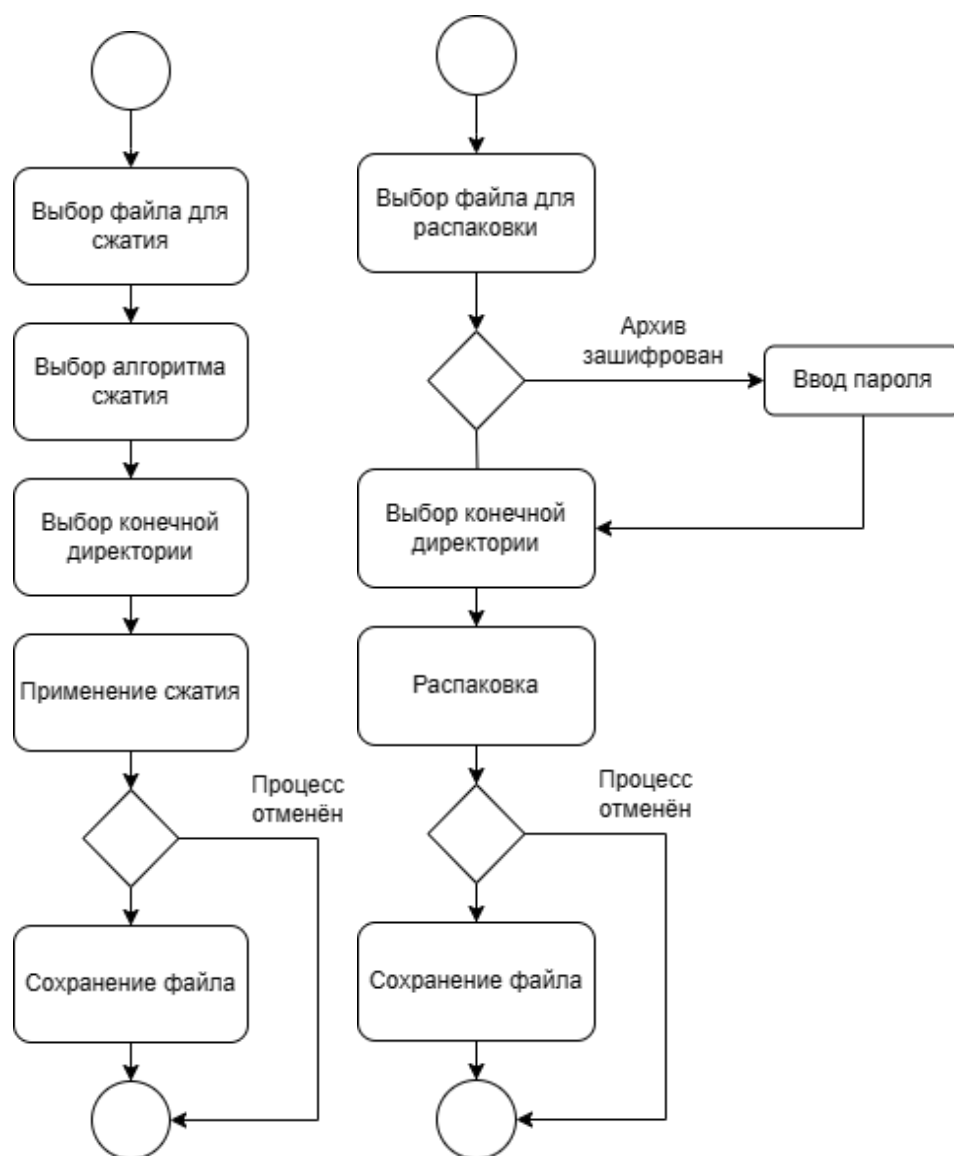


Рисунок 5 – Диаграммы активности «Сжатие и распаковка файла»

После того, как определены процессы поведения, определяется объектно-ориентированная структура программы.

2.2 Объектно-ориентированная структура программы

Программа состоит из 3 основных модулей:

1. Главный модуль (*source.py*) – является точкой входа в программу, запускает основное окно проводника с выбором файлов;
2. Модуль проводника (*FileExplorer.py*) – представляет из себя графический интерфейс программы, отвечает за работу окон, кнопок и полей ввода, а также управляет основными потоками сжатия и распаковки;
3. Модуль архиватора (*ArchiveTreads.py*) – описывает работу потоков сжатия и распаковки архивов, благодаря чему пользователь может приостановить процесс без закрытия программы.

Процессы сжатия и распаковки описываются следующими классами:

Класс *CompressionThread* (поток сжатия) – содержит поля пути к исходным файлам и целевому пути сохранения:

- *file paths* – пути к файлам;
- *save path* – путь сохранения архива.

Класс содержит следующие методы:

- *run* – метод для запуска потока сжатия.

Класс *UnzipThread* (поток распаковки) – содержит пути к архивам и путь для распаковки файлов:

- *archieve_path* – путь к архивам
- *extract_to* – путь для распаковки

Класс содержит следующие методы:

- *run* – метод запуска потока распаковки

Интерфейс приложения реализован с помощью библиотеки *PyQt5*.
Окно программы (конкретно файлового проводника) реализовано отдельным классом *FileExplorer*.

Класс *FileExplorer*.

Данный класс описывает главное окно программы, расположение графических элементов и связывает их с потоками сжатия/распаковки.

Класс содержит следующие методы:

- *go_home* – метод для кнопки возврата на домашнюю страницу проводника;
- *pc* – метод для кнопки открытия страницы «Этот компьютер»
- *go_up* – метод для кнопки поднятия вверх по пути «Вверх».
- *open_file* – метод для двойного нажатия по папке или файлу.
- *update_path_label* – метод для обновления строки текущего пути, когда там ничего не введено
- *change_directory* – метод для кнопки перехода по введённому пути
- *create_zip* – метод для кнопки «Добавить» связывающий её с потоком сжатия.
- *extract_zip* – метод для кнопки «Извлечь» связывающий её с потоком распаковки.

2.3 Руководство программиста

1. Средства, используемые для разработки приложения:

- язык программирования: *Python*;
- интегрированная среда разработки: *VSCode*;
- библиотеки: *PyQt5*, *zipfile*, *os*, *sys*.

2. Средства, необходимые для компиляции: *IDE VSCode* (или др.);

3. Характеристики программы:

- количество файлов с кодом: 3;
- общее количество строк кода: 245;
- объём файлов программы: 210 Кб;
- системные требования: минимальные;
- объём потребляемой памяти: до 4 Гб;
- версия операционной системы: *Windows 10*.

4. Входные данные:

- Наименование и путь к файлам сжатия/распаковки: *string*.

5. Выходные данные:

- Сжатые файлы;
- Распакованные файлы.

2.4 Тестирование программного продукта

Тестирование программного продукта проводилось методом «чёрного ящика». В таблице 2 приведена информация о тестировании приложения.

Таблица 2 – Результаты тестирования

№	Объект тестирования	Описание	Результат
1	Сжатие данных	Правильный выбор файлов для сжатия	Корректный ввод и добавление в таблицу
		Нажатие на кнопку «добавить», не выбрав ни один файл	Создание пустого архива
		Сравнение веса исходного и сжатого файлов	Сжатие в среднем до 60%
2	Распаковка данных	Правильный выбор файлов для распаковки	Корректная распаковка файлов по указанному пути
		Нажатие кнопки «Извлечь», не выбрав ни один файл	Ничего не произошло
		Распаковка файла, не являющегося архивом	Ничего не произошло
		Проверка правильности распаковки видео, программ, текста	Корректная распаковка с неизменной работоспособностью
3	Работа с файлами в проводнике	Двойное нажатие по файлу любого типа	Открытие папки, или запуск файла
		Указан некорректный путь/путь к несуществующему файлу	Откроется путь «Этот компьютер»
		Указан пустой путь к файлу	Ничего не произошло

Важно отметить, что тестирование процента сжатия на методе *DEFLATE* проводилось на бинарных и текстовых файлах, поскольку использование сжатия на файлах, к которым уже было применено сжатие, таким как *PNG*, *JPG* или *MP3*, не эффективно [4].

Таким образом, в ходе тестирования была проверена работоспособность программного продукта, возможность выполнения всех требуемых функций, а также поведение при некорректных действиях пользователя. Результаты проверки полностью соответствуют ожиданиям, следовательно, можно сделать вывод, что тестирование пройдено успешно.

2.5 Руководство пользователя

Приложение «Архиватор» создано для удобства пользователя работы с различными архивами, добавлением файлов в архив и их распаковки.

Графический интерфейс программы представляет из себя окно выбора файлов, в котором следует выбрать файлы для архивации или архивы для распаковки (рисунок 6).

На верхней панели расположены кнопки возврата в директорию пользователя (иконка дома), директорию всех накопителей (иконка компьютера), выхода на путь выше директории (иконка стрелки вверх).

Также там находится текстовое окно текущего пути, в которое можно вписать собственную директорию, к которой необходимо перейти. Для перехода к вписанной директории необходимо нажать на кнопку (иконка двойной стрелки вправо).

Далее расположены кнопки «Добавить» для добавления выделенных файлов в архив и «Извлечь» для извлечения выбранных архивов.

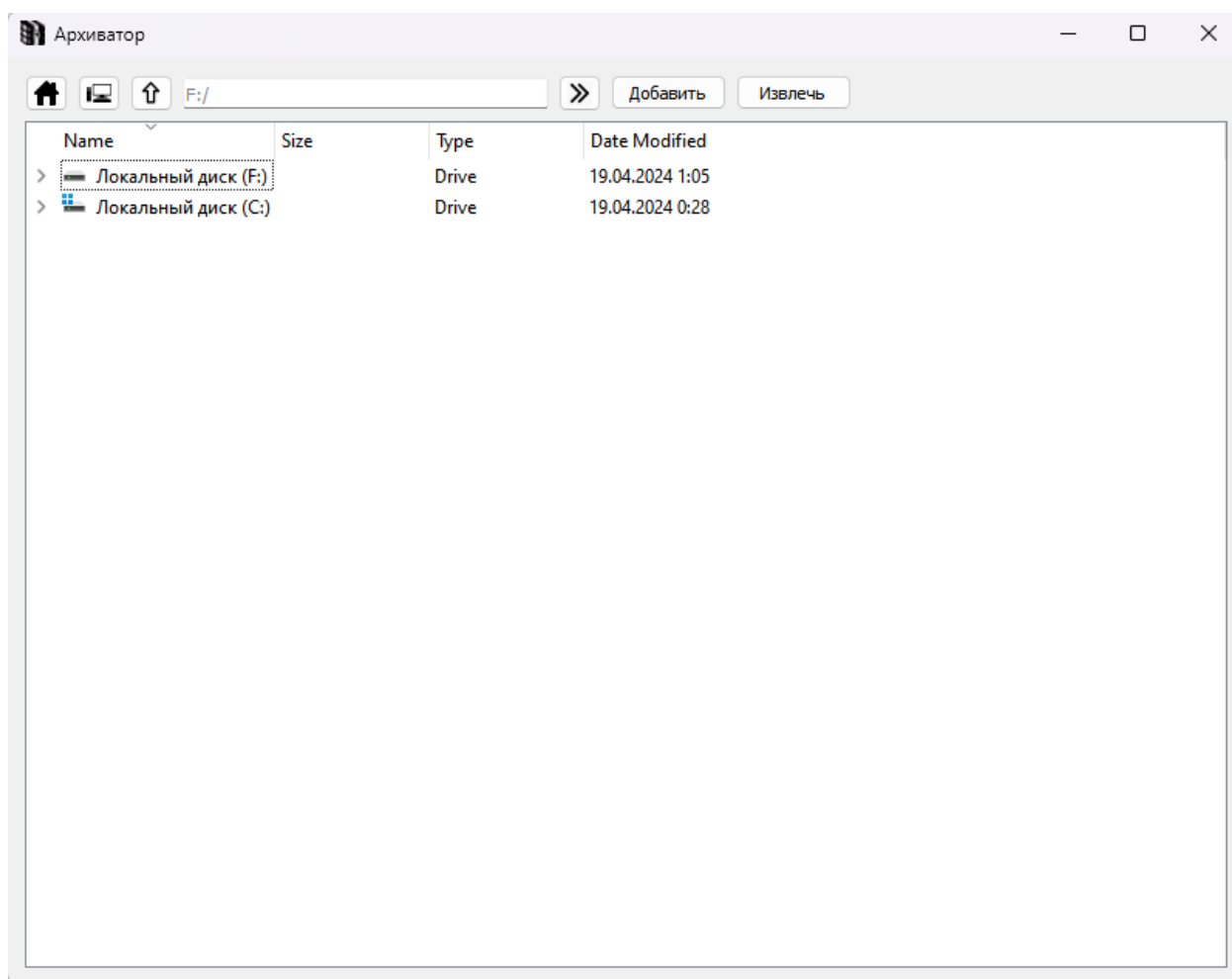


Рисунок 6 – стартовое окно программы

При добавлении файла в архив откроется окно выбора расположения и ввода названия архива (Рисунок 7).

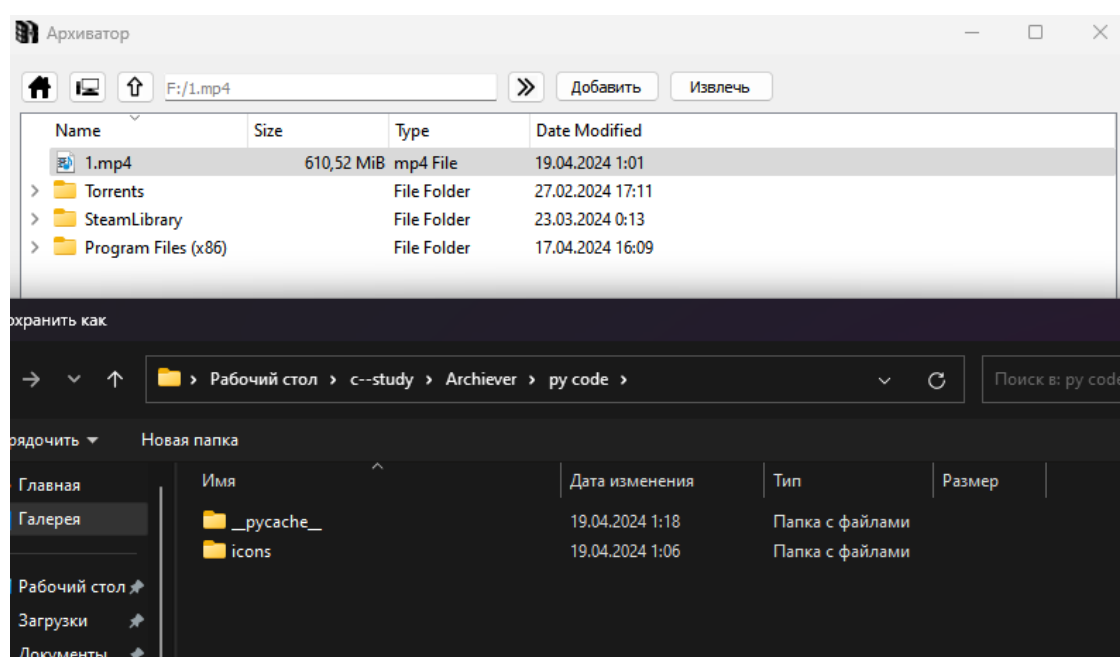


Рисунок 7 – окно выбора расположения и ввода названия архива

После выбора расположения и ввода имени архива запустится окно прогресса создания архива (Рисунок 8).

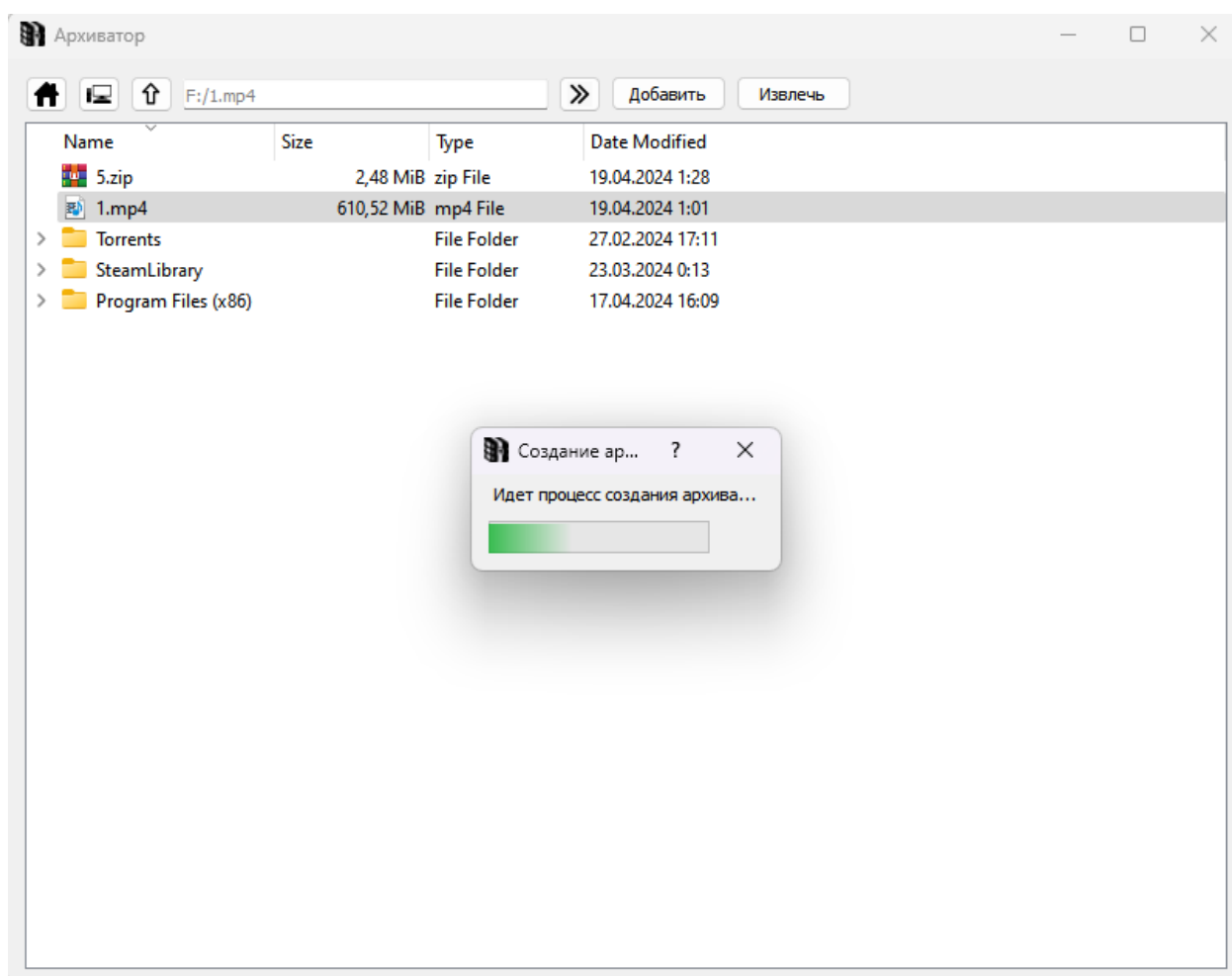


Рисунок 8 – окно прогресса создания архива

Процесс можно отменить, закрыв окно прогресса, программа будет способна к дальнейшей работе.

При распаковке выбранного архива пользователю также будет предложено выбрать путь распаковки (Рисунок 9).

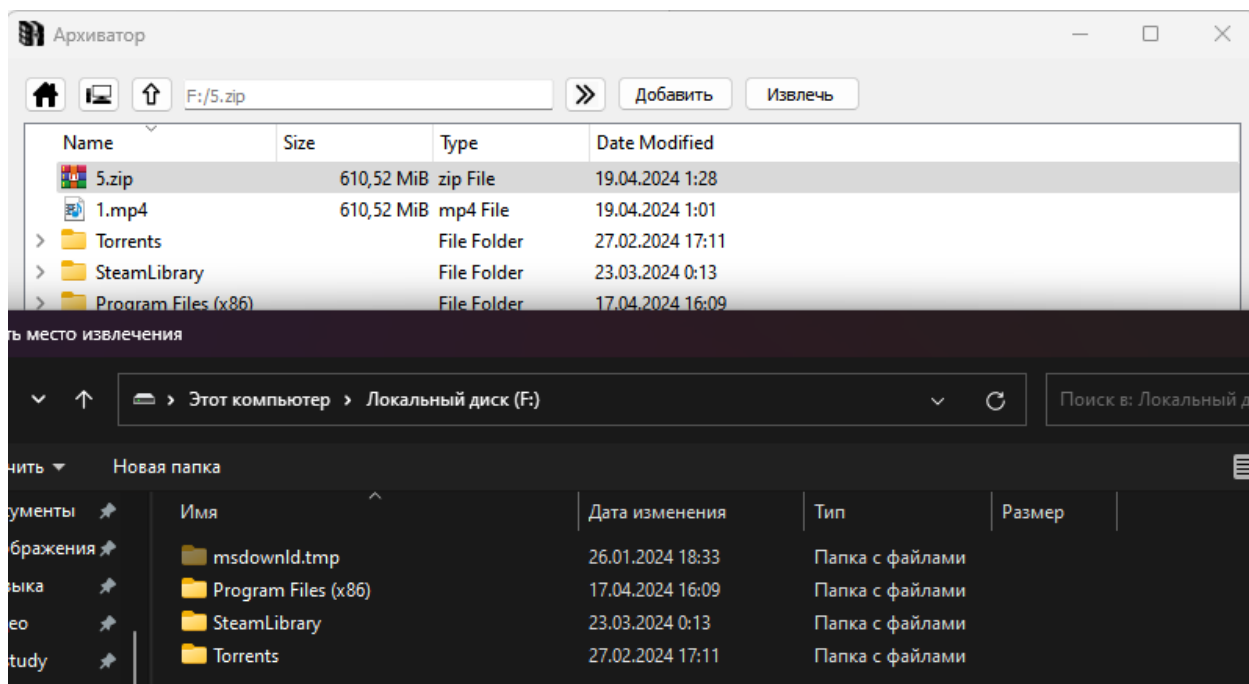


Рисунок 9 – окно выбора пути распаковки

Также любые файлы можно запускать из проводника или же заходить в папку дважды нажав по ним.

2.6 Выводы по главе

В данной главе мы подробно рассмотрели технические аспекты создания программы-архиватора. Были представлены динамические аспекты поведения системы, включая процессы сжатия и распаковки файлов, а также выбор методов сжатия. Кроме того, рассмотрена объектно-ориентированная структура программы, описаны классы и методы, используемые для работы с архивами.

Также было представлено руководство программиста, которое содержит подробное описание алгоритмов сжатия и распаковки файлов, а также инструкции по использованию различных функций программы-архиватора. Это позволяет разработчикам быстро ориентироваться в коде и вносить необходимые изменения.

В конце главы представлено руководство пользователя, которое содержит информацию о том, как пользоваться программой-архиватором. Пользователям предоставлены инструкции по сжатию и распаковке файлов, выбору методов сжатия, а также рекомендации по обеспечению безопасности данных.

В результате проведенного тестирования программный продукт успешно прошел все тесты, что подтверждает его стабильную работу и соответствие требованиям. Таким образом, можно сделать вывод, что разработанный программный архиватор является полностью рабочим и готовым к использованию.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы было разработано приложение "Программа-архиватор". Это приложение предоставляет пользователям удобный интерфейс для сжатия, распаковки и управления файлами.

Для достижения этой цели были выполнены все поставленные задачи: изучены существующие аналоги программ-архиваторов, определены требования пользователей к функционалу приложения, осуществлены проектирование и программная реализация проекта, а также проведено тестирование разработанного приложения.

Дальнейшее развитие и усовершенствование программы-архиватора может быть направлено на расширение функционала: добавление возможности работать одновременно с несколькими архивами, реализация функции сохранения и загрузки информации из приложения в текстовые документы или базу данных. Также усовершенствования могут быть направлены на улучшение интерфейса и добавление новых функций для удобства пользователей.

Разработанное приложение имеет огромную значимость для практического применения, так как является неотъемлемым инструментом для сжатия и управления файлами в современном мире цифровых технологий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ru.wikipedia.org – WinRAR: сайт – URL: <https://ru.wikipedia.org/wiki/WinRAR> (дата обращения: 10.03.2024) – Текст и изображения: электронные.
2. ru.wikipedia.org – 7-Zip: сайт – URL: <https://ru.wikipedia.org/wiki/7-Zip> (дата обращения: 10.03.2024) – Текст и изображения: электронные.
3. ru.wikipedia.org – PeaZip: сайт – URL: <https://ru.wikipedia.org/wiki/PeaZip> (дата обращения: 10.03.2024) – Текст и изображения: электронные.
4. pylot.me – Работа с ZIP архивами с помощью Python: сайт – URL: <https://pylot.me/article/35-rabota-s-zip-arhivami-s-pomoshyu-python/?ysclid=lw7ptn3qqu893031849#> (дата обращения: 14.05.2024) – Текст и изображения: электронные.
5. docs.python.org - zipfile — Work with ZIP archives: сайт – URL: <https://docs.python.org/3/library/zipfile.html> (дата обращения 14.05.2024) - Текст и изображения: электронные.

ПРОВЕРКА НА ОРИГИНАЛЬНОСТЬ

Проверка на оригинальность проводилась при помощи сайта *users.antiplagiat.ru*

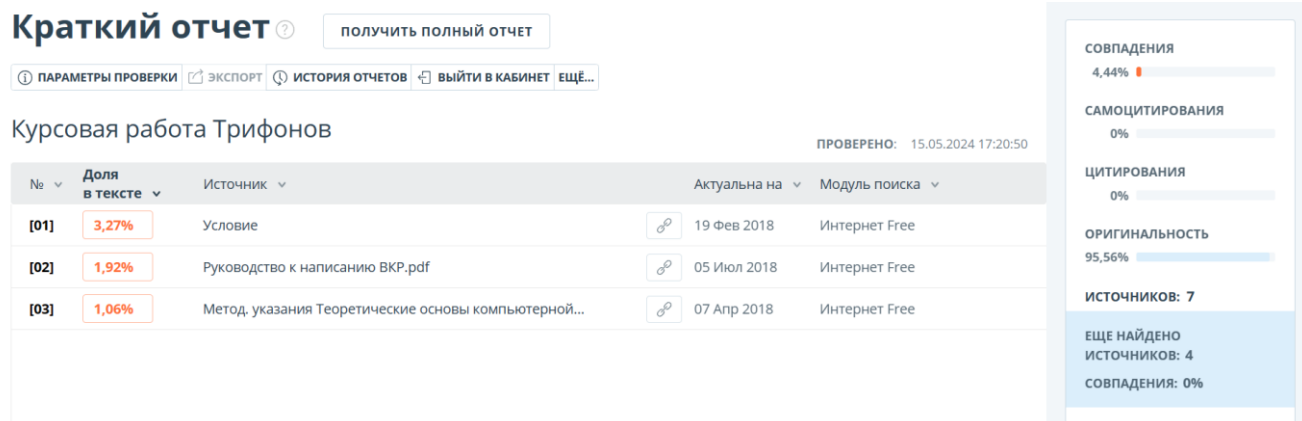


Рисунок 1 - Результат проверки на оригинальность