МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования «Самарский национальный исследовательский   
университет имени академика С.П. Королева»

(Самарский университет)

|  |  |
| --- | --- |
| Институт | информатики, математики и электроники |
| Факультет | Информатики |
| Кафедра | геоинформатики и информационной безопасности |

КУРСОВОЙ ПРОЕКТ ПО ДИСЦИПЛИНЕ

«Технологии и методы программирования»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

«Разработка программных средств автоматического резервного копирования данных пользователя»

|  |  |
| --- | --- |
| Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | Ю.Д. Кропивенцев |
|  |  |
| Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | Е.В. Мясников |
|  |  |

САМАРА 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования «Самарский национальный исследовательский   
университет имени академика С.П. Королева»

(Самарский университет)

|  |  |
| --- | --- |
| Институт | информатики, математики и электроники |
| Факультет | информатики |
| Кафедра | геоинформатики и информационной безопасности |

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

Студенту ***Кропивенцеву Юрию Дмитриевичу*** группы 6312

Тема проекта: ***«Разработка программных средств для автоматического копирования данных пользователя»***

|  |  |  |
| --- | --- | --- |
| Планируемые результаты освоения образовательной программы (компетенции) | Планируемые результаты практики | Содержание задания |
| ОПК-3 ‑ способность применять языки, системы и инструментальные средства программирования в профессиональной деятельности | Знать: основные виды программных средств, технологию разработки алгоритмов и программ и методы их отладки, основы объектно-ориентированного подхода к программированию  Уметь: работать с современными системами программирования, самостоятельно осваивать новые программные средства  Владеть: языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ. | 1. Изучение существующих методов и видов копирования данных  2. Разработка алгоритма копирования  3. Изучение существующих программ по реализации резервного копирования  4. Изучение существующих программных средств для реализации алгоритма  5. Реализация разработанного алгоритма копирования  6. Разработка и реализация программного интерфейса пользователя, а также защиты от несанкционированного использования  7. Отладка и тестирование, и сборка программы |

Дата выдачи задания 5 сентября 2023 г.

Срок представления на кафедру отчета о практике 22 декабря 2023 г.

Руководитель курсового проекта

доцент каф. ГИиИБ, к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.В. Мясников

*(подпись)*

Задание принял к исполнению

студент группы № 6312-100503D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.Д. Кропивнецев

*(подпись)*

РЕФЕРАТ

**Пояснительная записка к курсовому проекту:** ?? c., ?? рисунков, ?? таблиц, ?? источников, ?? приложений.

РЕЗЕРВНОЕ КОПИРОВАНИЕ, АЛГОРИТМ КОПИРОВАНИЯ, АВТОМАТИЧЕСКОЕ РЕЗЕРВНОЕ КОПИРОВАНИЕ, РАБОТА С ФАЙЛАМИ, ФАЙЛОВАЯ СИСТЕМА, АРХИВАЦИЯ, РАБОТА С АРХИВАМИ, ЛИЦЕНЗИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ *(5-15 слов)*

Цель работы – разработка и реализация алгоритма резервного копирования данных, с последующей имплементацией к приложению с графическим интерфейсом.

В ходе проделанной работы был сделан обзор существующих методов и решений в области резервного копирования информации, что послужило основой для написания собственного алгоритма. В ходе разработки были изучены существующие программные реализации алгоритмов на языке программирования Python, с дальнейшим применением в основном программном коде. В процессе проделанной работы был создан специальный вспомогательный модуль для работы с файлами, разгрузивший основной файл приложения. Разработанный продукт обладает интуитивно понятным интерфейсом, способствующим лучшему пользовательскому опыту. Реализована работа программы «в трее», большое количество времени было уделено отладке и защите приложения от поведения, не способствующего корректной работе файловой системы. В последних версиях был добавлен вход по логину и паролю, что способствует выполнение требования «программной защиты от несанкционированного использования». В результате работы имеется приложение, способное выполнять резервную копию файлов как в ручном, так и в автоматических режимах.

Содержание

[Содержание 4](#_Toc154092486)

[ВВЕДЕНИЕ 5](#_Toc154092487)

[1 МЕТОДЫ И АЛГОРИТМЫ РЕЗЕРВНОГО КОПИРОВАНИЯ 7](#_Toc154092488)

[1.1 Анализ методов резервного копирования 7](#_Toc154092489)

[1.2 Анализ существующих решений на рынке 9](#_Toc154092490)

[1.3 Изучение существующих программных средств для реализации алгоритма….. 12](#_Toc154092491)

[1.4 Разработка алгоритма… 12](#_Toc154092492)

[1.5 Выводы и результаты 12](#_Toc154092493)

[2 ОПИСАНИЕ РАЗРАБОТАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 13](#_Toc154092494)

[2.1 Выбор средств разработки и системных программных средств 13](#_Toc154092495)

[2.2 Требования к разрабатываемой программе 13](#_Toc154092496)

[2.3 Структура программы 13](#_Toc154092497)

[2.3.1 Основные модули 13](#_Toc154092498)

[2.3.2 Описание основных классов 13](#_Toc154092499)

[2.3.3 Описание основных структур данных 14](#_Toc154092500)

[2.3.4 Описание основных функций и процедур 14](#_Toc154092501)

[2.4 Описание интерфейса пользователя 14](#_Toc154092502)

[2.5 Контрольный пример и результаты тестирования 15](#_Toc154092503)

[2.6 Выводы и результаты 15](#_Toc154092504)

[3 ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ 16](#_Toc154092505)

[3.1 Данные для экспериментов 16](#_Toc154092506)

[3.2 Методика проведения эксперимента 16](#_Toc154092507)

[3.3 Результаты экспериментов 16](#_Toc154092508)

[3.4 Выводы и результаты 16](#_Toc154092509)

[ЗАКЛЮЧЕНИЕ 17](#_Toc154092510)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 18](#_Toc154092511)

[ПРИЛОЖЕНИЕ А 19](#_Toc154092512)

# ВВЕДЕНИЕ

В настоящее время мы почти каждый день имеем дело с далеко не самыми надежными носителями информации – крайне мало людей, не терявших важные данные из-за вышедшего из строя диска. Во избежание таких неприятных ситуаций были придуманы схемы копирования информации, примером может послужить легенда о правиле в компании Coca-Cola: оба создателя рецепта не должны летать в одном самолете. Так как сохранность информации актуальна и по сей день, резервное копирование информации обладает еще большим спросом, чем когда-либо.

На текущий момент времени на рынке представлено большое количество решений, предоставляющих большой спектр возможностей защитить свои данные. Копирование может быть представлено как загрузка на удаленный сервер хранения, что перекладывает ответственность на строну владельца сервиса, также архивация может производиться на внешний физический носитель, что позволяет вам непосредственно контролировать что происходит с вашей резервной копией. Большинство программ обладает сложным и перегруженным интерфейсом, что может вызвать затруднение у новых пользователей.

Целью данной работы является создать удобный и рабочий алгоритм резервного копирования, представленный в виде приложения для операционной системы Windows, обладающего простым и понятным интерфейсом, благодаря которому пользователь будет получать лучший опыт взаимодействия и решать задачи в области копирования информации. Для достижения поставленной цели были решены следующие задачи:

1. изучение видов резервного копирования и подходов к хранению копий, для последующей имплементации к проекту;
2. анализ существующих решений на рынке в целях получения общего представления об изучаемой сфере;
3. изучение и выбор программных средств и библиотек для написания авторской реализации полученных знаний;
4. разработка собственного алгоритма для последующей реализации;
5. реализация алгоритма копирования в виде программного кода на основе;
6. создание программы с графическим интерфейсом, обладающим простым и понятным дизайном;
7. отладка и тестирование созданного приложения.

# МЕТОДЫ И АЛГОРИТМЫ РЕЗЕРВНОГО КОПИРОВАНИЯ

## Анализ методов резервного копирования

Запрос на создание копии данных существовал задолго до появления компьютеров: второй экземпляр документов, летописи, те же конспекты – все это является дубликатом информации. С появлением цифровых устройств осуществлять подобные манипуляции вручную стало крайне долго и сложно в силу возросшего количества резервируемой информации. Решением стало «резервное копирование» – процесс, результатом которого является создание копии информации на носителе информации, например, на жестком диске или на удаленном сервере.

Резервное копирование преследует важную и простую цель: получение возможности быстро и недорого восстановить информацию в случае утери рабочей версии по какой-либо причине.

Система выполняющая роль резервного копирования должна отвечать нескольким требованиям:

* надежность хранения информации – обеспечение отказоустойчивости программных и технических средств;
* кроссплатформенность – возможность запускать систему резервного копирования на любом устройстве вне зависимости от операционной системы;
* простота эксплуатации – в первую очередь программой будут пользоваться обычные люди, что создает запрос на простой и понятный функционал;
* быстрое внедрение – развертывание системы должно происходить быстро, не требуя создания дополнительного окружения.

Разобравшись с требованиями и задачами, перейдем к описанию видов резервного копирования. Существует несколько видов резервного копирования:

* полное резервное копирование (full backup) – такое копирование, при котором выполняется полная копия всех данных, в основном затрагивается вся система и все файлы. Целесообразно производить раз в большое количество времени во избежание чрезмерной нагрузки на инфраструктуру и работу организации, а предыдущие копии сжимать и архивировать. Незаменим в случае необходимости восстановления системы.
* Дифференциальное резервное копирование (differential backup) – при таком типе копирования будет создана копия только изменившихся файлов с моментов последнего полного копирования. Положительно сказывается на процессе восстановления, особенно полезен при заражении файлов вирусами.
* Инкрементное копирование (incremental backup) – такой тип копирования отличается от двух вышеописанных подходов тем, что копирование происходит только тех файлов, которые были изменены с момента последней копии будь то полное копирование или дифференциальное. Последующее инкрементное копирование добавляет только изменившиеся файлы. Занимает меньше времени, однако требует большего времени восстановления из-за новых этапов: последовательное восстановление резервной копии, а после – инкрементных копий. Отличие инкрементного от дифференциального копирования в том, что поменявшиеся файлы добавляются на носитель независимо.
* Клонирование – копирование целого раздела со всем содержимым в другой раздел.
* Резервное копирование в виде образа – создание точной копии раздела или носителя, хранящейся в одном файле.
* Резервное копирование в режим реального времени – создание копии файлов, без остановки работы системы, процесс происходит «в фоне».
* Холодное копирование – копирование системы выполняется, когда пользователь не может вносить изменения в файлы, гарантируется согласованное состояние.
* Горячее копирование – такое копирование системы, при котором пользователи могут вносить изменения во время копирования, копия базы приводится в согласованное состояние путем использования журналов копирования по окончании основной операции резервного копирования.

Исходя из полученной информации было принято решение создать продукт, выполняющий задачи резервного копирования по типу полного резервного копирования и дифференциального резервного копирования в силу того, что предложенные подходы покрывают большинство случаев, а также отличаются простотой реализации. Также будет соблюдено требование кроссплатформенности и простоты эксплуатации.

## Анализ существующих решений на рынке

Убедившись в теоретическом аспекте вопроса, необходимо проверить, так ли нужен полноценный продукт покупателю, существует ли реальный запрос у рядового пользователя и у бизнеса, в частности.

Произведя анализ этой области рынка, было обнаружено множество решений, причем многие из них присутствуют на рынке уже далеко не первый год, а также, что радует, присутствует большое количество open-source проектов.

Проведем анализ наиболее известных и популярных решений:

* Acronis Cyber Protect Home Office (бывший Acronis True Image)

Программа для резервного копирования и восстановления данных, позволяющая создавать образы диска, резервные копии в ручном или автономном режиме, а онлайн хранилище позволяет дополнительно обезопасить данные. Поддерживается кроссплатформенность, запускать программное обеспечение (ПО) можно на операционных системах Microsoft Windows или MacOs или даже на мобильных устройствах с Android или Apple ios.

Отмечу интуитивный и приятный дизайн, возможность создания как онлайн, так и офлайн копии, а также возможность использования программы вне зависимости от операционной системы.

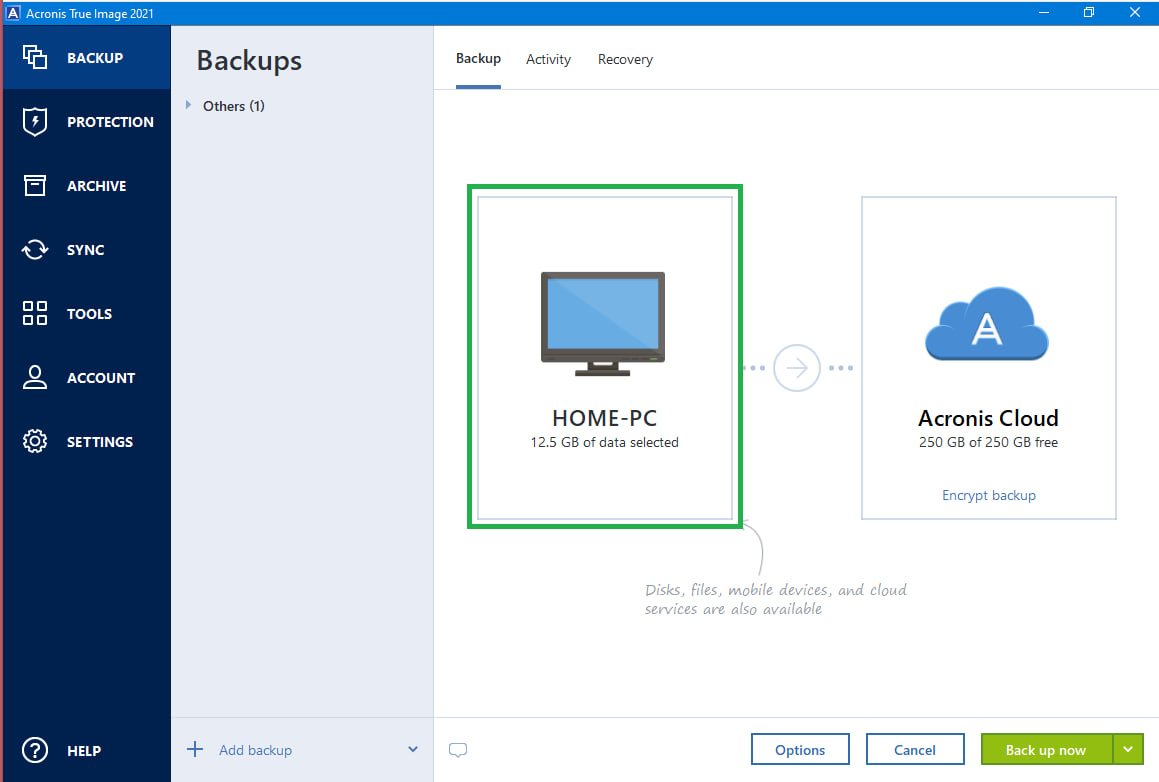


Рисунок 1 – Пример интерфейса программы Acronis True Image

* Time Machine (macOS) от Apple Inc

Программа, интегрированная в операционную систему Apple MacOs, позволяющая восстановить нужный вам файл или всю операционную систему в целом из нужного вам отрезка времени. Утилита работает как с облачным хранилищем, так и с физически подключенным к устройству. Обладая крайне простым и понятным функционалом, данная реализация имеет множество положительных отзывов от пользователей, потому что обладает крайне простым и понятным функционалом.

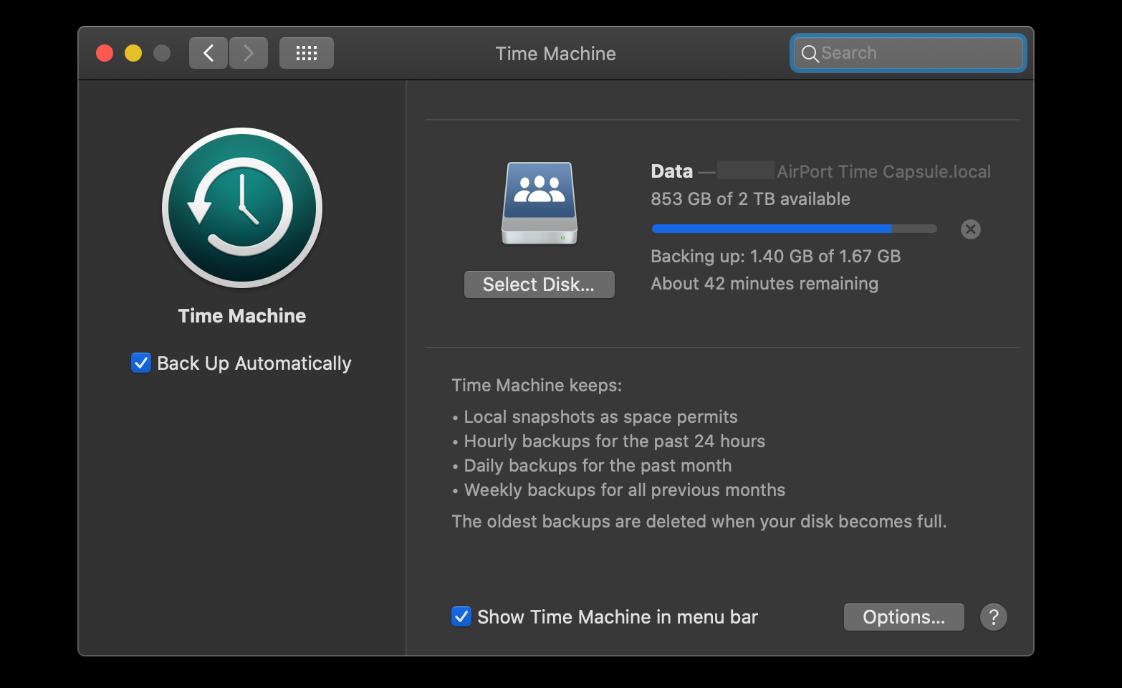


Рисунок 2 – Пример интерфейса Apple Time Machine

* Backup and Restore от Microsoft

Программа, являющаяся основной для создания резервной копии в операционной системе Windows 7. В ее функционал входит резервирование как файлов или папок, так и всей операционной системы в целом. Используется для восстановления системы. Копирование файлов и папок производится в двух режимах: полного и инкрементного копирований, а созданные копии хранятся в виде zip-архивов. А системы – в виде полноценного образа для последующего восстановления.



Рисунок 3 – Логотип Restore and Backup

Судя по рассмотренным вариантам, резервное копирование достаточно актуальная тема в силу того, что даже операционные системы предлагают собственные решения в этой области, а в финальную версию продукта войдут только лучшие решения.

## Изучение существующих программных средств для реализации алгоритма

Оценив количество работы, связанной с взаимодействием с файловой системой, необходимость создания современного и визуально приятного графического интерфейса и собственные требования к разработке, было принято решение в данном проекте использовать язык программирования Python. Такое решение обусловлено широким спектром возможностей при сравнительно меньших затратах усилий, причем качество разработки повышается за счет качественной документации и обилия материалов по языку.

Выбор библиотек оказался достаточно простым и быстрым, вместе с тем рассматриваемые экземпляры обладают отличной документаций и большим количеством примеров использования (os, shutl, flecmp, zipfile, pyqt5, haslib, logging):

* Библиотека os

Модуль предоставляет удобный вариант использования функций, зависящих от операционной системы, такие как работа с файловой системой, чтение и запись в файл, создание и удаление папок. Библиотека будет использоваться для проверки, существует ли путь, для создания папок. К достоинствам отнесем хорошую документацию и большое количество материалов.

* Библиотека shutil

Предоставляется высокоуровневое взаимодействие с файловой системой, отличительной чертой являются функции:

shutil.copytree(src, dst, symlinks=False, ignore=None, copy\_function=copy2, ignore\_dangling\_symlinks=False, dirs\_exist\_ok=False)

и

shutil.rmtree(path, ignore\_errors=False, onerror=None, \*, onexc=None, dir\_fd=None)

* Библиотека zipfile

При рассмотрении существующих решений выяснилось, что целесообразно хранить устаревшие копии в виде zip файлов, данная библиотека предоставляет такую возможность: чтение, запись, проверка и создание zip-архивов.

* Библиотека PyQt5

Qt – это набор кроссплатформенных библиотек C++, которые реализуют высокоуровневые API. PyQt – адаптированная под язык программирования Python описанной библиотеки, позволяющая использовать этот язык как альтернативу C++. С помощью этого фреймворка будет создана графическая составляющая проекта – окно для ввода и вывода.

## Разработка алгоритма

В момент написания алгоритма стояла задач написать алгоритм копирования, способный выполнять несколько задач:

* Модульность с возможностью быстрой модернизации
* Простота реализации
* Комфорт в использовании

Разработанные алгоримы: алгоритм согдания копии, алгоримат сраврения

Разработанный алгоритм создания резервной копии представлен на рисунке 4.

В начале алгоритма у пользователя запрашиваются входные данные: пути для копирования, список файлов, которые нужно игнорировать, и временной интервал копирования, то есть через какое время будет произведено следующее копирование, причем по умолчанию выбран ноль, а также тип копирования, по умолчанию выбора вариант, подразумевающий полное копирование.

Далее происходит проверка введенных данных: все ли пути существуют и верно ли введено время, если данная проверка не будет пройдена, пользователь увидит сообщение о том, что в процессе настройки произошел сбой и что необходимо ввести валидные данные. В случае выполнения условий алгоритм выполняется в штатном режим и введенные данные заносятся в соответствующие переменные.

Следующие несколько этапов связаны с установкой параметров: устанавливается интервал копирования и выбирается тип копирования.

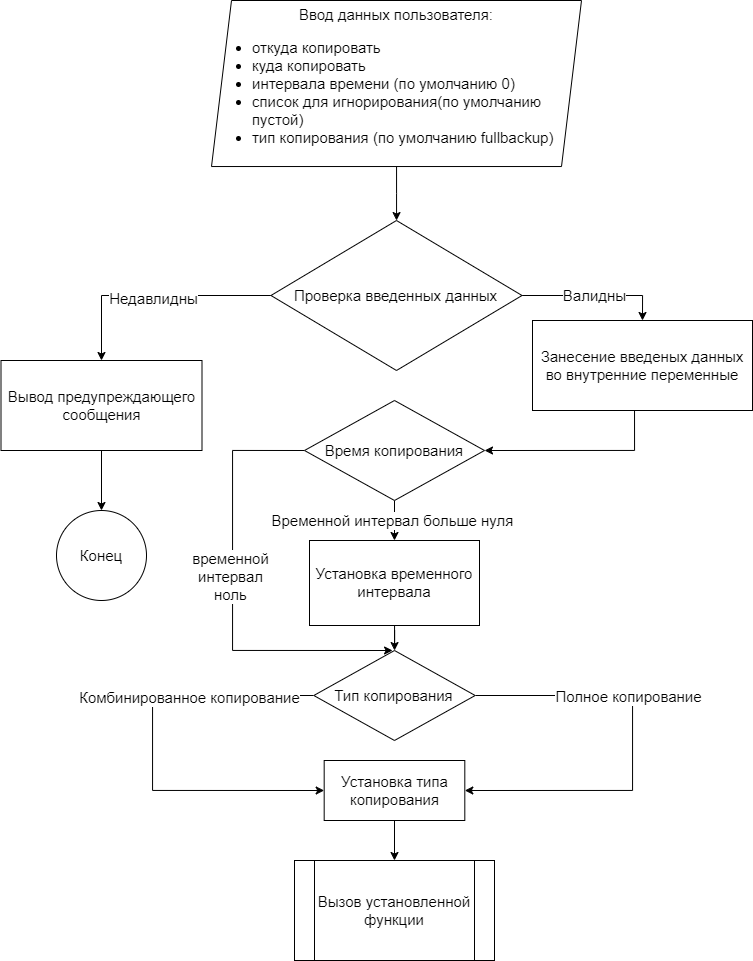


Рисунок 4 – Алгоритм копирования в виде блок схемы

Опишем алгоритмы копирования алгоритм полного копирования full\_backup представлен на рисунке 5

В начале алгоритма выполняется проверка на существование пути копируемой папки: если пути не существует – пользователь увидит сообщение об ошибке, а алгоритм завершится досрочно, в противном случае выполнение продолжится.

Далее проверяется количество уже существующих копий, если они существуют, то будет проведено сравнение последней копии с уже существующей, вариант в котором обе папки идентичны завершается установкой в соответствующем поле статуса последнего успешного копирования, а алгоритм начинается заново, если есть таймер и просто завершается иначе. Если же копий нет, то производится процедура создания нового имени для папки, с последующим копированием всех файлов. Следующие шаги аналогичны уже рассмотренным.

Комбинированное копирование аналогично алгоритму full\_backup, их различия заключаются в том, что рассматриваемый тип использует сразу и полный и дифференциальный виды копирования. Вместо того, чтобы каждый раз создавать копию всего раздела, программа меняет только изменившиеся файлы.

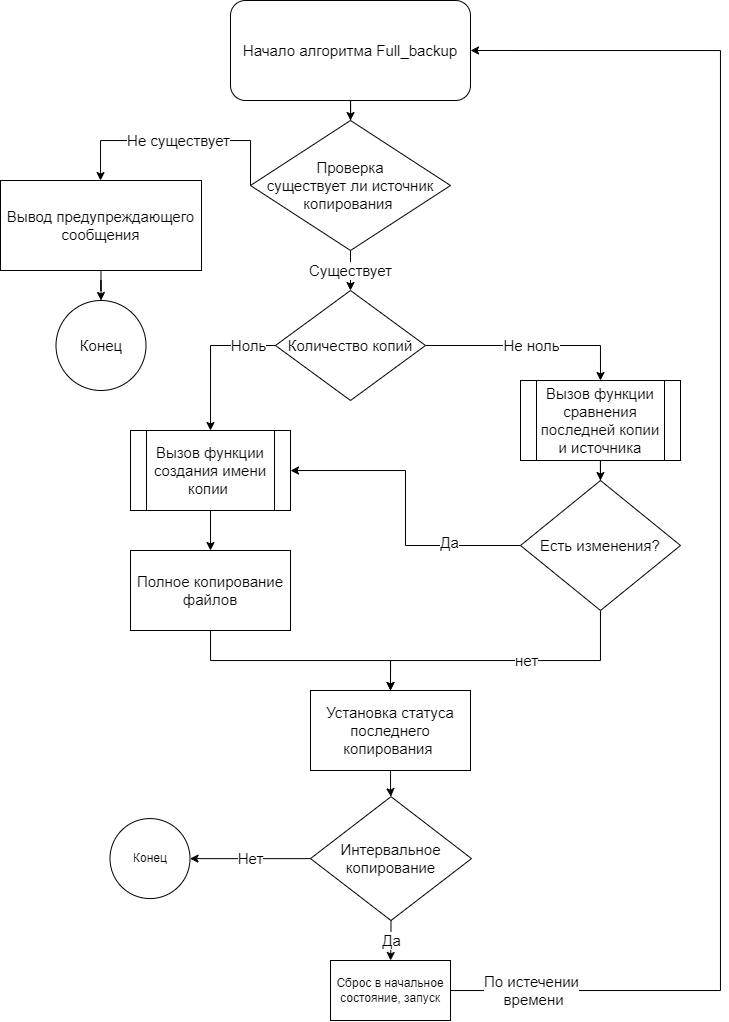


Рисунок 5 – Алгоритм полного копирования в виде блок схемы

## Выводы и результаты

В результате анализа видов резервного копирования, а также изучения существующих решений, было принято решение сосредоточиться на реализации полного и дифференциального видов резервного копирования. Основным видом послужит полное копирование, выполняющее роль основы всего проекта, а также алгоритм дифференциального копирования, чтобы разгрузить систему. Также будет реализован метод восстановления из резервной копии. Описанный функционал будет решать большинство задач, которые ставятся пользователем перед программой по автоматическому резервному копированию.

# ОПИСАНИЕ РАЗРАБОТАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В разделе описываются теоретические основы используемых в проекте алгоритмических и программных средств, изучение или разработка которых производилась студентом в соответствии с заданием.

Ориентировочный объем раздела 8-10 страниц для работ, имеющих исследовательскую часть, и 13-17 страниц для работ без исследовательской части. Примерная структура раздела приведена ниже.

## Выбор средств разработки и системных программных средств

В этом подразделе описывается выбранный язык и среда разработки с указанием причин такого выбора. Дается перечень необходимых для работы программы системных программных средств (операционная система, система управления базами данных и т.п.).

## Требования к разрабатываемой программе

В этом подразделе формулируются функциональные и нефункциональные требования к разрабатываемой программе.

## Структура программы

Здесь описывается структура программы. Указывается, из каких программных компонент (исполняемых файлов) она состоит.

### Основные модули

Для каждой компоненты указывается, из каких файлов с исходными кодами осуществляется сборка компоненты.

### Описание основных классов

Здесь дается описание основных классов, разработанных в рамках проекта. Для каждого класса приводится описание наиболее значимых полей и методов.

### Описание основных структур данных

Здесь дается описание основных структур данных (за исключением описанных в предыдущем разделе), разработанных в рамках проекта, если таковые имеются.

### Описание основных функций и процедур

В том случае, если часть логики программы располагается в подпрограммах (за исключением методов классов), здесь дается описание основных функций и процедур.

Фрагменты кода подпрограмм, используемые в разделе, рекомендуется оформлять следующим образом:

int Sum(int a, int b);

Рисунок 2 – Заголовок функции суммирования

Функция *Sum* возвращает сумму двух целочисленных значений, задаваемых входными параметрами *a* и *b*.



Рисунок 3 – Пример фрагмента кода

## Описание интерфейса пользователя

В этом разделе описывается интерфейс пользователя с указанием порядка работы с программой и назначением всех элементов пользовательского интерфейса.

## Контрольный пример и результаты тестирования

Необходимо разработать контрольные примеры и показать, что на них разработанная программа работает нормально, что должно подтверждаться приводимыми на рисунках копиями экрана.

Для исследовательских работ, содержащих раздел 3 «ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ», допускается не выполнять п. 2.5.

## Выводы и результаты

В этом подразделе приводятся основные выводы и результаты по разделу.

# ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

Ориентировочный объем раздела 5-7 страниц.

## Данные для экспериментов

Здесь описывается то, какие данные были использованы при проведении экспериментов.

## Методика проведения эксперимента

Здесь описывается методика проведения эксперимента. Каковы были критерии сравнения методов или алгоритмов, каким образом измерялись или оценивались величины. Сколько раз выполнялся эксперимент, что варьировалось и т.д.

## Результаты экспериментов

Результаты экспериментов приводятся в виде таблиц, графиков и диаграмм с описанием полученных результатов.

## Выводы и результаты

В этом подразделе приводятся основные выводы и результаты по разделу. Какие из сравниваемых методов и алгоритмов лучше использовать, в каких случаях и почему.

ЗАКЛЮЧЕНИЕ

Краткие выводы и оценка полученных в курсовом проекте результатов (не более 1 страницы).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Лебеденко, Ю.И. Биометрические системы безопасности [Текст]: учеб. пособие для вузов / Ю.И. Лебеденко. – М.: Директ-Медиа, 2012. – 160 с.  
- пример оформления учебного пособия

2 Форматы звуковых файлов: [Электронный ресурс] // Ингибитор. Второй сборник. URL: http://ingibit.rigalink.lv/info/c2/formats.html. (Дата обращения:  18.02.2018).  
- пример оформления электронного ресурса

3 Сорокин, В.Н. Распознавание личности по голосу: аналитический обзор [Текст] / В.Н. Сорокин, В.В. Вьюгин, А.А. Тананыкин // Информационные процессы. – 2012. – Т. 12, №1. – С. 1-30.  
- пример оформления книга 3 автора

4 Первушин, Е.А. Обзор основных методов распознавания дикторов [Текст] / Е.А. Первушин // Математические структуры и моделирование. –2011. – № 24. – С. 41-54.  
- пример оформления статья в журнале

5 Рабинер, Л.Р. Цифровая обработка речевых сигналов [Текст]: пер. с англ. / Л.Р. Рабинер, Р.В. Шафер. – М.: Радио и связь, 1981. – 496 с.

ПРИЛОЖЕНИЕ А

Код программы

В приложения выносится вспомогательный материал:

* исходные коды программ;
* результаты измерений;
* описания используемых наборов данных и т.д.

Допускается разделять раздел на подразделы, например, «А.1», «А.2». Такие подразделы могут соответствовать, например, разным программным модулям.