МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

ОТЧЕТ

О РЕЗУЛЬТАТАХ ИНДИВИДУАЛЬНОЙ

ПРЕДДИПЛОМНОЙ ПРАКТИКИ

ФГАОУ ВО «ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(наименование организации)

КАФЕДРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

(наименование структурного подразделения)

|  |  |  |
| --- | --- | --- |
| *ка была групповой, то указываются ФИО всех участвовавших обучающихся* | | |
| Выполнил  обучающийся 4 курса,  МОиАИС-184 группы |  | Поляков И. А. |
|  | (подпись) | (ФИО) |
| Руководитель практики  от института |  | Павлова Е.А. |
|  | (подпись) | (ФИО) |

Защищен \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022

Результаты

экзамена / зачета \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (подпись руководителя практики от института)

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc104992368)

[Раздел 1 (проект 2 курса) 5](#_Toc104992369)

[ГЛАВА 2. РАЗРАБОТКА ПРИЛОЖЕНИЯ 7](#_Toc104992370)

[2.1 Архитектура ПО 7](#_Toc104992371)

[Список литературы 12](#_Toc104992372)

[Приложение 12](#_Toc104992373)

# ВВЕДЕНИЕ

На сегодняшний день информация является одним из важнейших ресурсов и одной из движущих сил прогресса и цивилизации. Информационные технологии активно используются во многих сферах жизнедеятельности человека.

Объем информации, используемой человеком, настолько велик и разнообразен, что бумажные носители не обеспечивают ее эффективное накопление, хранение и использование. С момента появления первых вычислительных машин получило широкое распространение хранение информации в двоичном коде. Для ее хранения используются разнообразные запоминающие устройства.

Большой объем важной и конфиденциальной информации хранится в цифровом виде. Для ее защиты разработаны различные инструменты шифрования, ограничения прав доступа и резервного копирования.

Резервное копирование в последнее время стало синонимом защиты данных. Защита данных от потери, различных повреждений и других проблем является одной из приоритетных задач для IT компаний.

Разработка приложений для резервного копирования направлена на обеспечение безопасности данных при непредвиденных ситуациях, сбоях аппаратных средств, программного обеспечения.

**Актуальность** темы дипломной работы объясняется наличием большого объема информации в электронном виде, которую необходимо защищать от потерь с помощью программ резервного копирования и восстановления данных.

**Цели и задачи практики** – разработать эффективный механизм резервного копирования и восстановления данных с удаленным хранением копий.

Для достижения поставленной цели необходимо решить следующие задачи:

* + изучить основные принципы организации процесса резервного копирования и восстановления данных;
  + выделить оптимальные способы резервного копирования, а также выбрать место хранения резервных копий и способ их переноса;
  + разработать программное обеспечение, реализующее функции резервного копирования, синхронизации и восстановления данных;
  + централизованное автоматизированное управление процедурами резервного копирования, хранения и восстановления данных (далее - РК);
  + создание резервных копий данных на устройствах хранения данных;
  + управление расписанием создания резервных копий;
  + централизованно- управляемое восстановление данных;

# Раздел 1 (проект 2 курса)

**Название проекта**

**Проблема**

Предоставленный учебный план на сайте ТюмГУ сложен к зрительному восприятию студентам, так как избыточен и представлен в неудобном формате для чтения.

**Актуальность**

Учебный план в вузе — это документ, где указано, какие дисциплины, в каком порядке и каком объёме будет изучать студент конкретной специальности. Также из плана можно узнать, сколько практик будет у учащихся, по каким предметам и в какие семестры их ждёт защита курсовых. Удобный просмотр учебного плана позволяет не тратить много времени на поиск необходимой информации.

**для кого предназначен проект**

Студенты и преподаватели.

**насколько полезен**

Значительно помогает сократить время для поиска необходимой информации по учебному плану, благодаря поиску по фразам, словам, ключевым индексам, поддерживает работу с Excel.

**что нужно для реализации проекта**

Исходные данные с сайта ТюмГУ и хостинг сервера.

**описание данных и их источников**

Исходные данные с сайта ТюмГУ в форматах pdf/csv

**ваши предложения (с обоснованием) по использованию технологий и моделей обработки данных.**

Bootstrap-верстка адаптивного дизайна сайта, со множеством шаблонов оформления компонентов, что позволяет сэкономить время при разработке дизайна.

Git - удобный инструмент для взаимодействия нескольких разработчиком в одном проекте.

XAMPP - кроссплатформенная сборка веб-сервера, содержащая Apache, MySQL, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер.

# ГЛАВА 2. РАЗРАБОТКА ПРИЛОЖЕНИЯ

**Описывается программа и результаты её работы: инструментальные средства и технологии решения задачи в рамках ВКР**

## 2.1 Архитектура ПО

Приложение разрабатывается с помощью языка программирования C#. В качестве среды программирования используется Visual Studio 2022. Для взаимодействия между сервером и клиентом используется библиотека Network.

Приложение состоит из трех частей:

1. DesktopClient – приложение с графическим интерфейсом. С его помощью пользователь может добавлять, редактировать, а также мониторить список задач.
2. ClientService – работает в виде службы Windows. Получает список задач и по расписанию выполняет их (отправляет файлы на сервер).
3. Server – получает и сохраняет у себя файлы. Также хранит у себя список задач и передает его клиентам.

Взаимодействие между приложениями осуществляется по сети по протоколу TCP/IP. Для этих целей используется библиотека Network- [<https://github.com/Toemsel/Network>]. Библиотека позволяет посредством сети получать и отправлять пакеты с различными данными. Для передачи списка задач используется пакет ‘tasks’. Передача файлов осуществляется пакетом ‘backup’

Сервер сохраняет список задач в файле Tasks.json. Для сохранения списка в формате JSON применяется библиотека Newtonsoft.Json [https://habr.com/ru/post/481514/].

Для создания задачи необходимо указать файл, который будет сохраняться, выбрать периодичность выполнения задачи из трех вариантов: ежедневно, еженедельно или ежемесячно. Также нужно указать время выполнения.

Архитектура приложения представлена на рисунке 1.

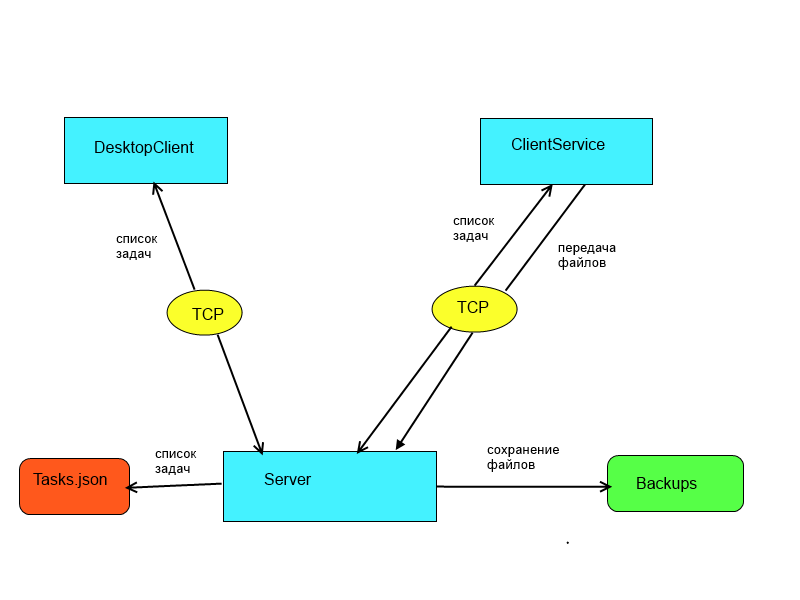


Рис 1. Архитектура приложения.

(описание и схемы, код методов, использование библиотек)

Сериализация/Десериализация  
newtonsoft.json -[https://habr.com/ru/post/481514/]

Взаимодействие между управляющей и подчиненной службой

Network-[<https://github.com/Toemsel/Network>]

Описание методов-[https://www.indie-dev.at/2016/10/03/server-setup/]

Картинки-[https://metanit.com/sharp/net/3.2.php]

**Используемые технологии**

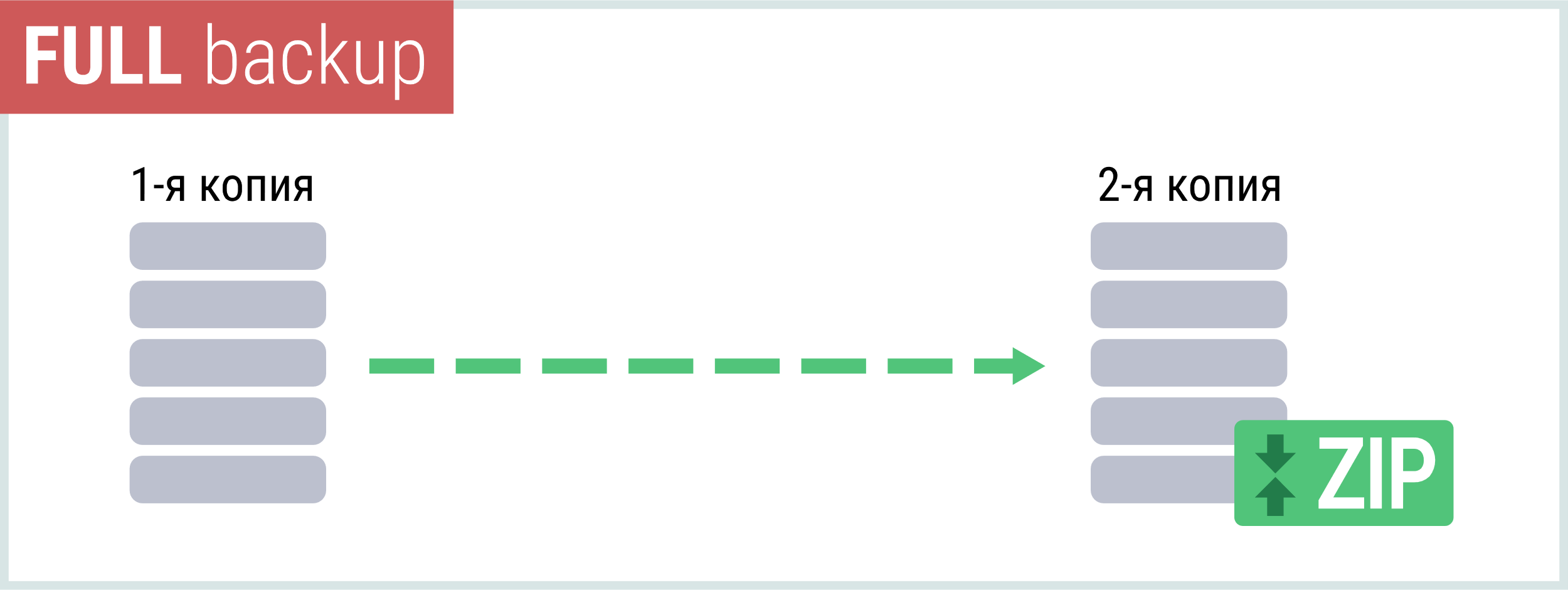
* C# — [объектно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) для платформы [Microsoft .NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework) и [.NET Core](https://ru.wikipedia.org/wiki/.NET_Core).
* Windows Forms — интерфейс программирования приложений, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework.

**Алгоритмы**

(виды резервирования, описать подробно)

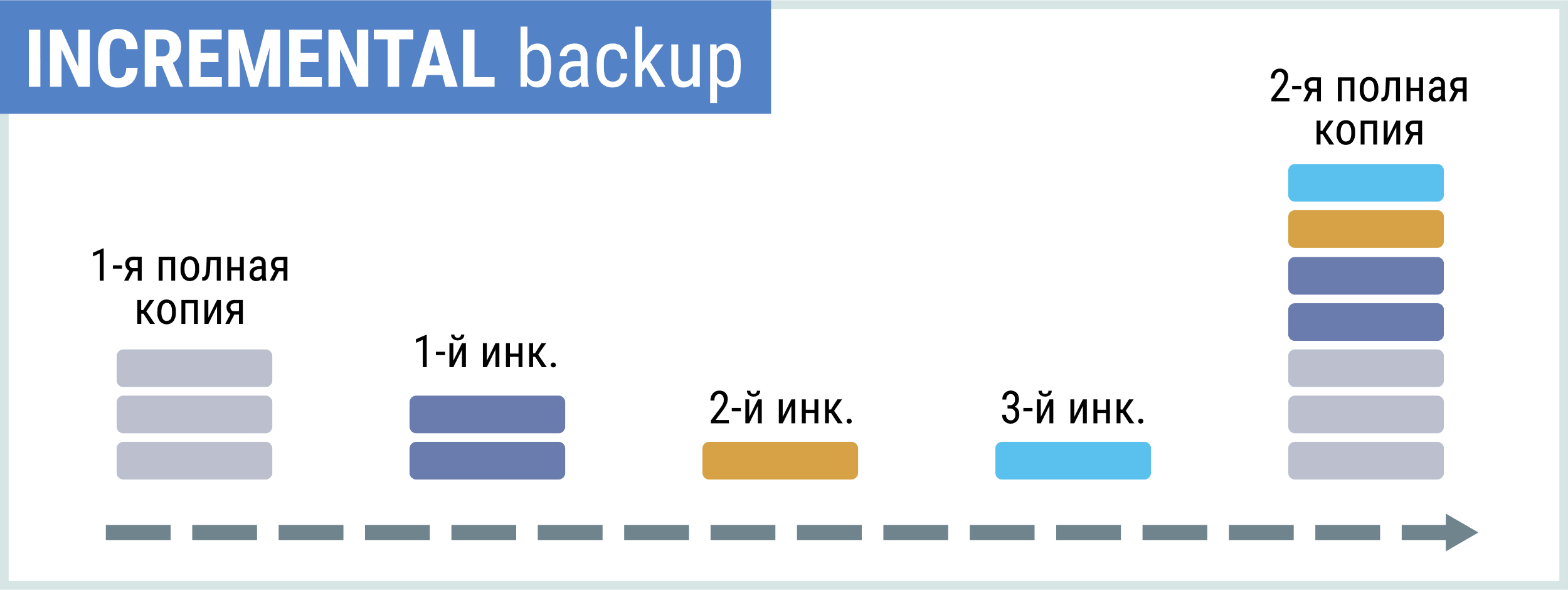
Приложение поддерживает следующие схемы резервирования данных:

* + полное резервное копирование;



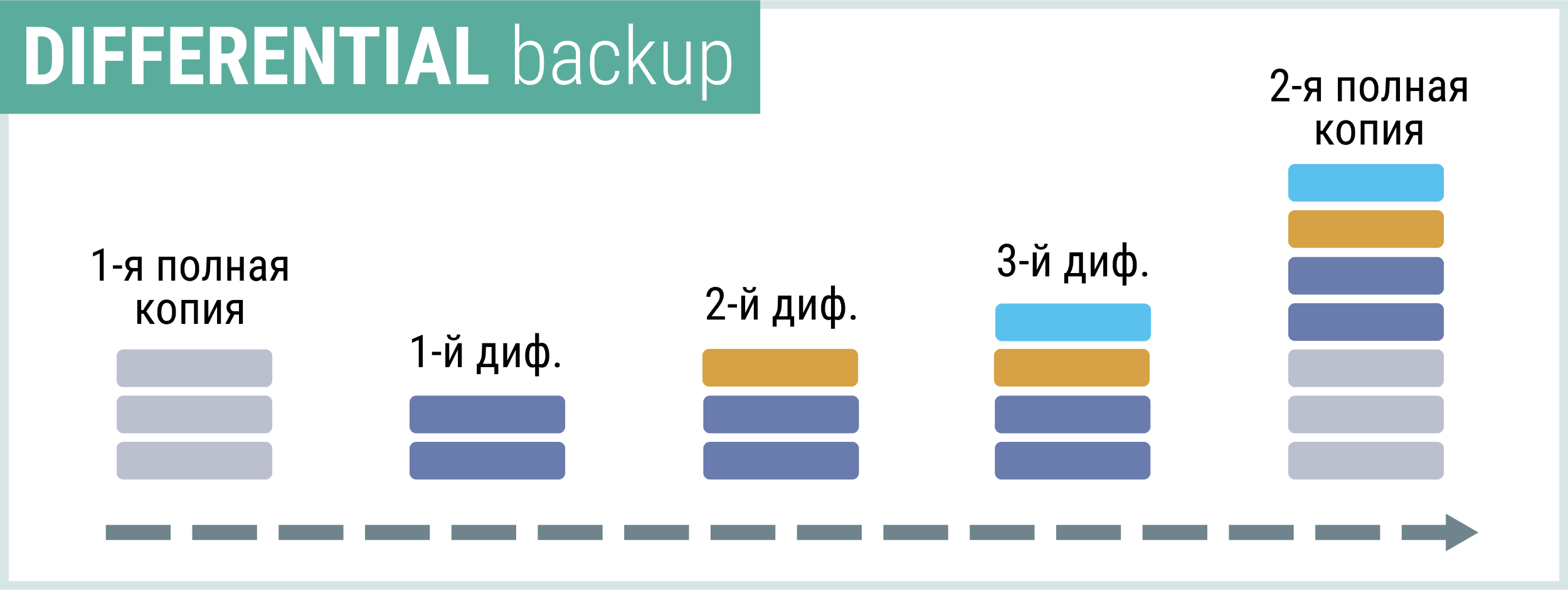
При полном бэкапе каждый раз создается полная копия **всей системы**, точнее, всех тех данных, которые вы определили для резервного копирования при постановке задачи. Для уменьшения итогового объема резервной копии все данные сжимаются в архив. Таким образом, в вашем хранилище при полном резервном копировании с заданной периодичностью появляются архивы, где данные в основной своей массе дублируются (поскольку на протяжении долгого времени не изменяются). Это серьезный недостаток, ведь расходуется огромный объем ресурсов: место в хранилище, время создания и процессорное время, вычислительные мощности, наконец, ресурсы трафика при транспортировке архивов в удаленную СХД.

* Метод полного копирования ранее был очень распространенным из-за высокой надежности, однако в чистом виде на сегодняшний день он признан малоэффективным. Например, для резервного копирования невысокой глубиной (менее двух недель) или с высокой частотой (раз в сутки, раз в несколько часов) полный бэкап чрезмерно расходует ресурсы.
* Немного спасет ситуацию механизм **дедупликации** – выявление и удаление дублирующихся данных в полных копиях. Он также задается специальными программными средствами как на уровне СХД или сервера, так и на клиенте непосредственно. Статистика в некоторых источниках приводит впечатляющие результаты степени дедупликации – от 90% до 98%.
* Преимуществом полного бэкапа можно назвать разве что скорость восстановления: когда данные поднимаются из одного архива, это происходит быстрее, чем при инкрементальном или дифференцированном бэкапе. На сегодняшний день метод полного резервного копирования, как правило, используется исключительно как базовый в сочетании с другими методами, менее ресурсоемкими. Иногда такой подход называют еще **смешанным** или **синтетическим** бэкапом.



Инкрементальный вид резервного копирования гораздо экономичнее и быстрее, чем полный бэкап, поскольку в этом процессе копируются только те файлы, которые изменились со времени **предыдущего** резервного копирования. Исходные данные, записанные изначально, не перезаписываются. Механизм инкрементального копирования прост: в качестве начальной точки бэкапа Х0 выбирается время (например, полночь с воскресенья на понедельник), в которое делается полный бэкап; в точке Х1 (полночь с понедельника на вторник) делается копирование файлов, измененных и/или появившихся с момента Х1; в точке Х2 (полночь со вторника на среду) копируются файлы, измененные/появившиеся с момента выполнения Х1; … в точке Хn происходит завершение цикла и делается следующий полный бэкап.

Этот метод гораздо более **экономично** расходует ресурсы и места в хранилище, и времени, и трафика передачи данных, по сравнению с другими. Однако при восстановлении данных в случае необходимости из резервной копии происходит поэтапное восстановление из точек Хn-1…Х2, Х1, Х0 – до последнего полного бэкапа включительно, и этот процесс может занять много времени в зависимости от объема данных.



Дифференциальный бэкап выигрывает перед инкрементальным в случае **скорости** восстановления данных – время на эту операцию у него меньше, поскольку сравниваются полные копии Х0 и Хn и не требуется поэтапного восстановления. Однако в части объема пространства для размещения в СХД дифференциальное резервное копирование сопоставимо с полным, поэтому экономии места в хранилище и трафика практически не достигается.

При дифференциальном бэкапе происходит копирование «нарастающим итогом»: каждый измененный файл в каждой последующей точке бэкапа копируется заново. То есть выглядит это как: Х0, Х1, Х1+Х2, Х1+Х2+Х3, … +Хn, Х0+Х(1+…n)

Словом, очень громоздко и сложно при расчете места в СХД.

Понять разницу между инкрементальным и дифференциальным бэкапом достаточно просто. Фактически – она в одном слове. Просто сравните:

* инкрементальный обрабатывает файлы, измененные или созданные с момента выполнения предыдущего бэкапа;
* дифференциальный обрабатывает файлы, измененные или созданные с момента выполнения предыдущего полного бэкапа.

Виды резервирования- [https://www.sim-networks.com/ru/blog/backup-full-increment-differential]

**структура базы данных с описанием таблиц**

(еще нет, возможно sqlite)

**интерфейс (скриншоты)**

**интерпретация результатов работы программы (выводы, иллюстрации, таблицы)**

**инструкция пользователя**.

# Список литературы

1. Работа с библиотекой Newtonsoft.Json на реальном примере // Ресурс для IT-специалистов Хабр [Электронный ресурс]. URL: <https://habr.com/ru/post/481514/> (дата обращения: 14.05.2022).

# Приложение