**МИОНОБРНАУКИ РОССИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ»  
(СПбГЭУ)**

Факультет Информатики и Прикладной Математики  
Кафедра

**КУРСОВАЯ РАБОТА**  
по дисциплине:  
**«Информатика и программирование»**

Тема: Создание программного комплекса для компрессии и декомпрессии информации в электронном виде

Направление (специальность) *09.03.03 Прикладная информатика*

Направленность (специализация) *Управление бизнес-процессами и проектами*

Обучающийся *Стерлин Павел Михайлович*

Группа *ПИ-2201* Подпись:\_\_\_\_\_\_\_\_\_\_\_\_

Проверил: Сотавов А. К.

Должность: \_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_ Дата: \_\_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург  
2023

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 1](#__RefHeading___Toc799_3007368307)

[ВВЕДЕНИЕ 3](#__RefHeading___Toc801_3007368307)

[1. ИСТОРИЯ РАЗВИТИЯ АВТОМОБИЛЕЙ И ИХ ОБСЛУЖИВАНИЯ 4](#__RefHeading___Toc803_3007368307)

# ВВЕДЕНИЕ

**Актуальность темы.** Количество данных создаваемых человечеством поистине огромно, увеличиваясь по экспоненте каждый год. Часть этих данных попадет в долгосрочное хранилище — архив. Будь то обязанность компаний хранить данные о пользователях или необходимость такой услуги для функционирования организации, например соц сети, где пользователи ожидают сохранения своих данных на длительный срок, для всего этого необходимы мощность для хранения данных. Но нередко к таким данным не нужен быстрый доступ, как например к исполняемым файлам, поэтому для увеличения вместительности можно использовать компрессию данных. В таком случае мы обмениваем быстродействие на вместительность. Степень такого обмена может варьироваться, например мы можем использовать сложный алгоритм компрессии для хранения фотографий на сервисе, но при их отправке клиенту мы столкнемся с необходимостью их декомпрессировать, что вызовет повышенную нагрузку на сервер, что замедлит работу нашего клиента. Поэтому создание приложения с гибкой настройкой возможности компрессии является актуальной для сферы задачей.

**Цель исследования:** создание программного обеспечения, на основе дерева Хаффмана, позволяющего компрессировать и декомпрессировать данные.

**Для достижения цели исследования поставлены следующие задачи**:

* Разобраться в теории работы алгоритмов компрессии, математически доказать их оптимальность;
* Составить модель работы программы для работы с архивами;
* Реализовать на основе составленных в модели алгоритмов приложение.

**Объект исследования**: серверы и персональные компьютеры.

**Предмет исследования**: архивация данных, представленных в электронном виде.

# ИСТОРИЯ РАЗВИТИЯ И СПОСОБЫ КОМПРЕССИИ ИНФОРМАЦИИ

Одним из ранних и общеизвестных примеров примеров компрессии информации является код Морзе. Разработанный в 1838 году Самюэлем Морзе, он предназначался для использования в телеграфии, в нем для обозначения более часто используемых символов применяются коды меньшей длины, жертвуя длиной менее часто используемых символов, тем самым уменьшая общую длину сообщения. Например, в русском варианте, самый популярный символ - «Е», поэтому он кодируется самым коротким кодом - «.». Самые длинные коды принадлежат знакам препинания, например восклицательный знак имеет код - «\_ \_ . . \_\_».

Но современные понятия о компрессии данных начались с создания теории информации. Отцом информационной теории считается Клауд Шеннон(1916 — 2001). Статьи Шеннона «Математическая теория связи» и «Теория связи в секретных системах» считаются основополагающими для теории информации и криптографии. Его помощником и второй по величине фигурой был Роберт Фано(1917 — 2016). Вместе они разработали код Шеннона-Фано, позволяющего определить коды для блоков текста на основе частоты их появления текста, базовый принцип работы приведен ниже:

1. Подсчет вероятностей. Для каждого отдельного символа необходимо подсчитать вероятность его появления в сообщении. Для этого подсчитаем сколько раз он встречается в сообщении и поделим на количество символом во всем тексте.

2. Сортировка по вероятности. Символы сортируются по вероятности в убывающем порядке, это гарантирует присвоение сиволам с большей вероятностью меньших по длине кодовых слов.

3. Разделение и присвоение кодов. На этом символы распределяются в 2 множества таким образом, что сумма вероятностей в каждом множестве приближается к 0,5. Одному из множеств присвоим код «0», второму «1».

4. Рекурсивное кодирование. Шаг 3 повторим для каждой группы дописывая в конец кодового слова «1» или «0», в зависимости от группы в которую они попадают. Необходимо повторить эту процедуру пока каждому индивидуальному символу не будет присвоен уникальный код.

5. Построение двоичного дерева. На основе нашего алгоритма построим двоичное дерево, для последующего декодирования.(приложение 1)

Такой метод использует особенности языка, где одни символы встречаются чаще других. Позднее оптимальный метод построения такого дерева был предложен Дэвидом Хаффманом, этот принцип мы рассмотрим во 2 разделе так как он является основой практической части работы.

Ранние варианты компрессии использовали предзаписанные в «железе» кодовые слова. Это делалось для упрощения вычислений и коррекции ошибок. Позднее с увеличением производительности компьютеров появилась возможность динамически, то есть в процессе работы, менять кодовые слова. Такая методика получила название Адаптивное дерево Хаффмана. Это улучшало компрессию, ведь учитывались особенности данных, с которыми осуществлялась работа, но требовало дополнительных вычислительных затрат.

До этого рассмотренные нами алгоритмы работают с любыми данными в двоичном представлении, они могут отражать все что угодно от звуков, то текста и 3д моделей. Все они используют компрессию без потерь, то есть мы можем полностью восстановить данные без потерь информации. Но не все типы данных требуют абсолютной точности, конечно текст нельзя сжимать с потерей информации, он быстро станет труднопонимаемым, а если он будет представлять машинные команды, то абсолютно бесполезным. Для таких данных характерно сжатие в 3 раза, но для изображений сжатие может происходить и в 20 раз. Такое сжатие без потерь невозможно, но потери в таком случае не являются критичными для человека. Алгоритмы, которые мы рассмотрим далее, уже используют особенности восприятия человека, чтобы иметь возможность убрать ненужную или не заметную для человека информацию.

Первые алгоритмы для обработки звука использовались для обработки человеческой речи, для кодирования, например музыки, они не использовались, ведь они опирались на особенности работы человеческого голоса. Так называемая Линейно предсказательная модель (*Linear predictive coding*) делит звуковую дорожку на части по несколько десятых миллисекунд, назовем их сэмплами. За это время частота изменяется линейно и чтобы получить достаточный уровень детализации достаточно примерно 8 сэмплов, которые были записаны до этого, новый сэмпл будет линейной комбинацией этих 8 сэмплов, это позволяло достигать сжатия в 8 раз. Такой метод не мог быть использован для музыки, ведь он не рассчитан для неё, в отличии от формата MP3.

MP3 это формат и аудикодек, созданный в 1990ых годах Институтом интегральных схем общества Фраунгофера (*Fraunhofer-Institut für Integrierte Schaltungen*). Он быстро приобрел популярность благодаря своей способности сильно уменьшать размер фалов, почти незаметно ухудшая их качество.

Этот алгоритм сжатия данных с потерями, поэтому результат после декомпрессии отличается от оригинального файла. Сжатие может достигать 11 раз. Такое высокое сжатие и малая потеря качества обусловлена использованием достижений психоакустики — науки о восприятии музыки человеком. Благодаря особенностям слуха человека, будь то динамическое восприятие громкости или чувствительность к определенным частотам позволяет эффективно находить избыточную для слушателя информацию.

При работе звуковая дорожка делится на «кадры» - маленькие части файла, которые обрабатываются отдельно. Алгоритм затем анализирует каждый из этих кадров для выявления избыточной информации. Например многие устройства неспособны воспроизводить очень низкие и очень высокие частоты, поэтому они удаляются.

После обработки кадров они кодируются с изменяемым битрейтом, насколько он большой определяется сложностью музыки, но не только. Например после громкого кадра, следующий тихий за ним тихий, может быть более низкого качества, ведь наш слух не успел перестроится. В общем случае более монотонные и громкие кадры используют низкий битрейт.

MP3 стал сейчас почти стандартом для потребляемой музыки, он поддерживается всеми современными устройствами и приложениями, его используют не только для музыки, но и для аудиокниг, подкастов и аудио эффектов.

Для компрессии изображения также используются особенности восприятия человека. Например формат JPEG использует тот факт, что человек лучше воспринимает яркость чем цвет. Поэтому блоки одинакового размера, содержащие информацию о цвете имеют разрешение 8 на 8, а содержащие информацию о яркости — 16 на 16. Также применяются математические преобразования вроде дискретного косинусного преобразования и другие сложные операций про которые мы не будем говорить. Можно заметить тренд на увеличение сложности алгоритмов с усложнением самой информации.

Компрессия видео сейчас является актуальной проблемой с распространением потокового видео, применяемого на прямых трансляциях, видеохостингах и онлайн-конференциях. Из-за большого объёма данных в первый видеокодек — h.261, созданный в 1988, была встроена компрессия. На данный момент самая популярная версия кодека — h.264. Который использует множество техник для компрессии видео изображения. Например предикативное кодирование, которое использует информацию из прошлого кадра, чтобы предсказать следующий и хранит только разность между ними. Также исполь