

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой  
О. В. Непомнящий  
подпись  
«\_\_\_\_\_» \_\_\_\_\_ 2020 г.

## БАКАЛАВРСКАЯ РАБОТА

09.03.01 – «Информатика и вычислительная техника»

Интерактивное электронное учебное пособие по алгоритмам сжатия данных

Руководитель	_____ подпись, дата	доцент, канд. техн. наук	А. И. Постников
Выпускник	_____ подпись, дата		Д. А. Дедик
Консультант	_____ подпись, дата	ст. преподаватель	К. В. Пушкарев
Нормоконтролер	_____ подпись, дата		А. И. Постников

Красноярск 2020

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
**«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_  
О. В. Непомнящий  
подпись  
« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ  
в форме бакалаврской работы**

Студенту Дедику Дмитрию Александровичу.

Группа: КИ16-08Б. Направление (специальность): 09.03.01  
«Информатика и вычислительная техника».

Тема выпускной квалификационной работы: «Интерактивное электронное учебное пособие по алгоритмам сжатия данных».

Утверждена приказом по университету № 6623/с от 26.05.2020.

Руководитель ВКР: А. И. Постников, канд. техн. наук, доцент, доцент каф. вычислительной техники ИКИТ СФУ.

Исходные данные для ВКР: нет.

Перечень разделов ВКР:

1. Анализ задания на выпускную квалификационную работу.
2. Проектирование и реализация приложения.
3. Инструкции.

Перечень графического материала: не требуется.

### **Задание**

Разработать приложение — интерактивное электронное учебное пособие по алгоритмам сжатия данных.

### **Основные требования**

Разработанное приложение должно иметь модульную архитектуру, быть кроссплатформенным для настольных систем (ОС Windows, Linux), иметь графический пользовательский интерфейс, открытый исходный код и обладать следующими возможностями:

- а) поддержка визуализации работы алгоритмов LZ77, LZ78, LZW;
- б) наглядная пошаговая демонстрация состояния компрессора и декомпрессора (по отдельности и вместе);
- в) возможность задавать входные данные и параметры алгоритмов;
- г) возможность модифицировать состояние алгоритмов вручную;
- д) автоматический и ручной режим воспроизведения;
- е) возможность возврата к предыдущим шагам;
- ж) определение коэффициента сжатия.

Руководитель ВКР

---

А. И. Постников

подпись

Задание принял к исполнению

---

Д. А. Дедик

подпись

30 декабря 2019 г.

## **РЕФЕРАТ**

Выпускная квалификационная работа по теме «Интерактивное электронное учебное пособие по алгоритмам сжатия данных» содержит 34 страницы текстового документа, 19 иллюстраций, 1 таблицу, 15 использованных источников.

**АЛГОРИТМ, СЖАТИЕ ДАННЫХ, ПРИЛОЖЕНИЕ, КЛАСС, ИНТЕРФЕЙС, УЧЕБНОЕ ПОСОБИЕ.**

Цель работы: создание кроссплатформенного приложения, позволяющего наглядно продемонстрировать работу словарных методов сжатия, а именно LZ77, LZ78 и LZW.

Задачи работы:

- анализ задания на выпускную квалификационную работу;
- подбор средств разработки приложения;
- проектирование приложения согласно заданию;
- реализация приложения с помощью выбранных средств.

В результате выполненной работы было спроектировано и реализовано интерактивное электронное учебное пособие по алгоритмам сжатия данных. Был разработан графический интерфейс программы, структура приложения. Адаптированы три алгоритма сжатия без потерь: LZ77, LZ78 и LZW. Была добавлена возможность настраивать алгоритмы.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Анализ задания на выпускную квалификационную работу .....	4
1.1 Обзор существующих решений .....	4
1.1.1 Gzip in Julia .....	5
1.1.2 Тренажер «RLE» .....	6
1.1.3 Тренажер «Huffman» .....	8
1.1.4 CodSH.....	9
1.1.5 LZ Dictionary Algorithms .....	11
1.1.6 Run Length Encoding .....	13
1.1.7 Dasher .....	15
1.1.8 Итоги обзора.....	18
1.2 Выбор средств разработки.....	20
1.3 Диаграмма прецедентов .....	20
2 Проектирование и реализация приложения .....	24
2.1 Интерфейс программы.....	24
2.2 Структура программы .....	25
3 Инструкции .....	27
3.1 Инструкция пользователя .....	27
3.2 Инструкция разработчика .....	30
ЗАКЛЮЧЕНИЕ .....	32
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	33

## **ВВЕДЕНИЕ**

В ходе изучения методов сжатия данных по дисциплине «Информатика» на первом курсе ИКИТ возникает потребность в наглядной демонстрации работы этих методов. Практически в каждом источнике информации о методах сжатия данных присутствует подробное текстовое описание алгоритма работы, но отсутствует визуальное представление этого алгоритма. Именно та информация, которую продемонстрировали наглядно, усваивается человеком лучше всего.

Целью данной выпускной квалификационной работы является создание кроссплатформенного приложения, позволяющего наглядно продемонстрировать работу словарных методов сжатия, а именно LZ77, LZ78 и LZW.

Для достижения цели были установлены следующие задачи:

- анализ задания на выпускную квалификационную работу;
- подбор средств разработки приложения;
- проектирование приложения согласно заданию;
- реализация приложения с помощью выбранных средств.

# **1 Анализ задания на выпускную квалификационную работу**

В данной выпускной квалификационной работе необходимо разработать кроссплатформенное приложение с открытым исходным кодом и модульной архитектурой, соответствующее следующим требованиям:

- поддержка ОС Windows и Linux;
- наличие графического интерфейса пользователя;
- поддержка русского и английского языков;
- поддержка визуализации алгоритмов LZ77, LZ78 и LZW;
- наглядная пошаговая демонстрация состояния компрессора и декомпрессора (по отдельности и вместе);
- возможность задавать входные данные и параметры алгоритмов;
- возможность модифицировать состояние алгоритмов вручную;
- наличие автоматического и ручного режимов воспроизведения;
- возможность вернуться к предыдущим шагам;
- возможность определения степени сжатия.

## **1.1 Обзор существующих решений**

Прежде чем приступать к проектированию и реализации приложения, нужно произвести поиск существующих на данный момент решений поставленной задачи.

На сегодняшний день существует огромное множество приложений, способных сжимать данные, используя различные алгоритмы. Однако количество решений, наглядно демонстрирующих подробности процесса сжатия, гораздо меньше.

При поиске аналогов основным критерием выступает наличие любой информации о процессе сжатия, которая обычно скрыта от пользователя при использовании привычных средств сжатия данных.

### **1.1.1 Gzip in Julia**

Данное решение с открытым исходным кодом реализовано на языке Julia, которое демонстрирует работу команды gunzip (распаковка сжатого файла в терминале ОС Linux) на примере текстового файла. Автором является Julia Evans [1].

Программа выполняет всего одно действие – показывает наглядно декомпрессию файла, сжатого с помощью команды gzip. В качестве входных данных используются строго сжатые командой gzip текстовые файлы. Графический интерфейс пользователя отсутствует. Процесс происходит в консоли в автоматическом режиме.

На рисунке 1 в фигурных скобках можно увидеть, какая часть текста была раскодирована в процессе декомпрессии [2].

```

bork@kiwi:~/work/hackerschool/gzip.jl$ julia gunzip.jl raven.txt.gz
Once upon a midnight dreary, while I {pon}dered weak an{d wea}{ry,}
Over many{ a }quaint{ and }curious volume of forgotten lore,
W{hile I }nodded, n{ear}ly napping, su{dde}n{ly }th{ere} ca{me }a t{apping,}
As{ of }so{me o}ne gent{ly }r{apping, }{rapping} at my chamb{er }door.
`'Tis{ some }visitor,'{ I }mu{tte}r{ed, }`t{apping at my chamber door} -
O{nly th}is,{ and }no{thi}{ng }m{ore}.'

Ah, distinc{tly }I reme{mber }it was in{ the} bl{eak }Dec{ember}{,
A}{nd }each separate dy{ing }{ember }wrou{ght }its ghost{ upon }{the }fl{oar.
}Eager{ly I }wish{ed }{the }{mor}row; - v{ain}{ly I }had{ so}{ught }to b{oorrow}
From{ my }book{s s}urcease{ of so}{rrow}{ - }{sorrow }{for}{ the }l{ost }Len{ore}{ -
}F{or the }ra{re }{and }radia{nt }mai{den}{ wh}{om }{the }angels{ na}{me }{Lenore -
}N{ame}less {here }{for }e{ver}{more.}

{
And }{the }silk{en }s{ad }u{nce}rt{ain} rustl{ing }{of }{each }purp{le }{cur}{tain}
Thrill{ed }{me }- f{illed me }with f{ant}a{sti}c {ter}ror{s n}{ever} felt be{for}e;
So{ th}{at }now,{ to }{sti}ll{ the b}eat{ing of }{my }h{ear}t,{ I }stoo{d r}ep{eating}{-
}`'Tis some visitor} {ent}r{eating }{entr}a{nce }{at my chamber door} -
}S{ome }l{ate }{visitor entreating entrance at my chamber door}{; -}{-
Th}{is }{it }{is, and nothing more},{'
}

}

```

Рисунок 1 – Демонстрация работы команды gzip

Так как данное решение имеет открытый исходный код, в процессе реализации собственного решения возможны некоторые заимствования, поскольку оно обладает рабочей реализацией алгоритма LZ77.

### 1.1.2 Тренажер «RLE»

Данное решение используется в учебных целях и демонстрирует работу алгоритма сжатия данных RLE. Автором является Константин Поляков [3]. Интерфейс пользователя представлен на рисунке 2.

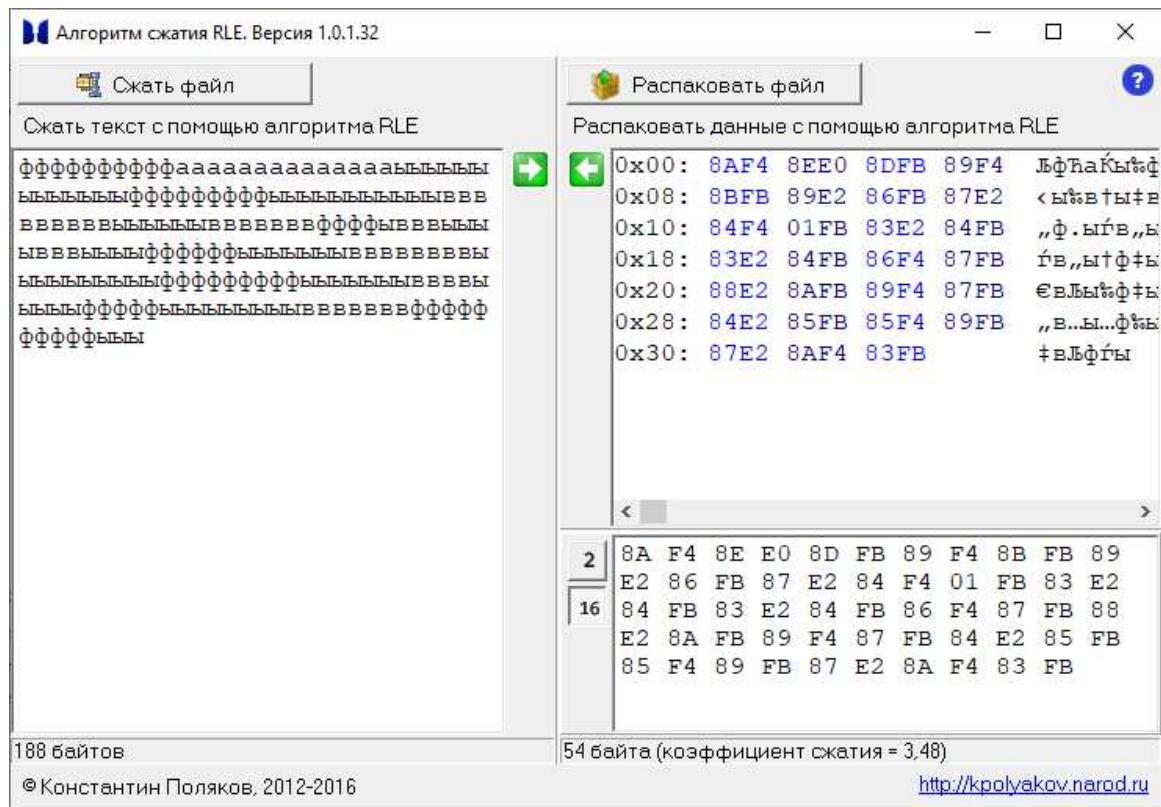


Рисунок 2 – Интерфейс пользователя тренажера «RLE»

Тренажер способен выполнять следующие действия:

- сжимать файлы любого типа;
  - распаковывать сжатые файлы;
  - сжимать текст в окне программы;
  - определять степени сжатия;
  - представлять сжатый файл в виде строк битов или байтов;
  - демонстрировать закодированные цепочки символов.

Автор данной программы не распространяет исходные файлы. Нет информации о том, на каком языке реализована. Графический интерфейс пользователя только на русском языке. Работает только под управлением ОС Windows. Внутри программы содержится инструкция к программе и краткая теоретическая справка по алгоритму «RLE».

Данное решение обладает понятным пользовательским интерфейсом – кнопки управления подписаны, под кнопками присутствует краткое описание операции, которые они выполняют, рядом с окном выходных данных есть кнопки «2» и «16», что означает представление в двоичной или шестнадцатеричной системе счисления соответственно. В ходе реализации собственного решения такой интерфейс может послужить в качестве примера.

### 1.1.3 Тренажер «Huffman»

Это решение, как и рассмотренное в п. 1.1.2, реализовано Константином Поляковым, и демонстрирует работу алгоритмов Хаффмана, Шеннона-Фано и LZW [3]. Исходный код программы недоступен, запуск только под управлением ОС Windows. Графический интерфейс пользователя только на русском языке. Внутри тренажера содержится инструкция к программе и краткая теоретическая справка об алгоритмах, реализованных в приложении.

Данная программа может выполнять следующие действия:

- сжимать выбранным методом файлы любого типа;
- распаковывать сжатые файлы;
- сжимать текст в окне программы выбранным методом;
- определять степени сжатия;
- представлять сжатый файл в виде строки битов;
- демонстрировать закодированные цепочки символов;
- демонстрировать сформированный словарь;
- анализировать файлы для определения предела сжатия.

Пользовательский интерфейс тренажера «Huffman» представлен на рисунке 3.

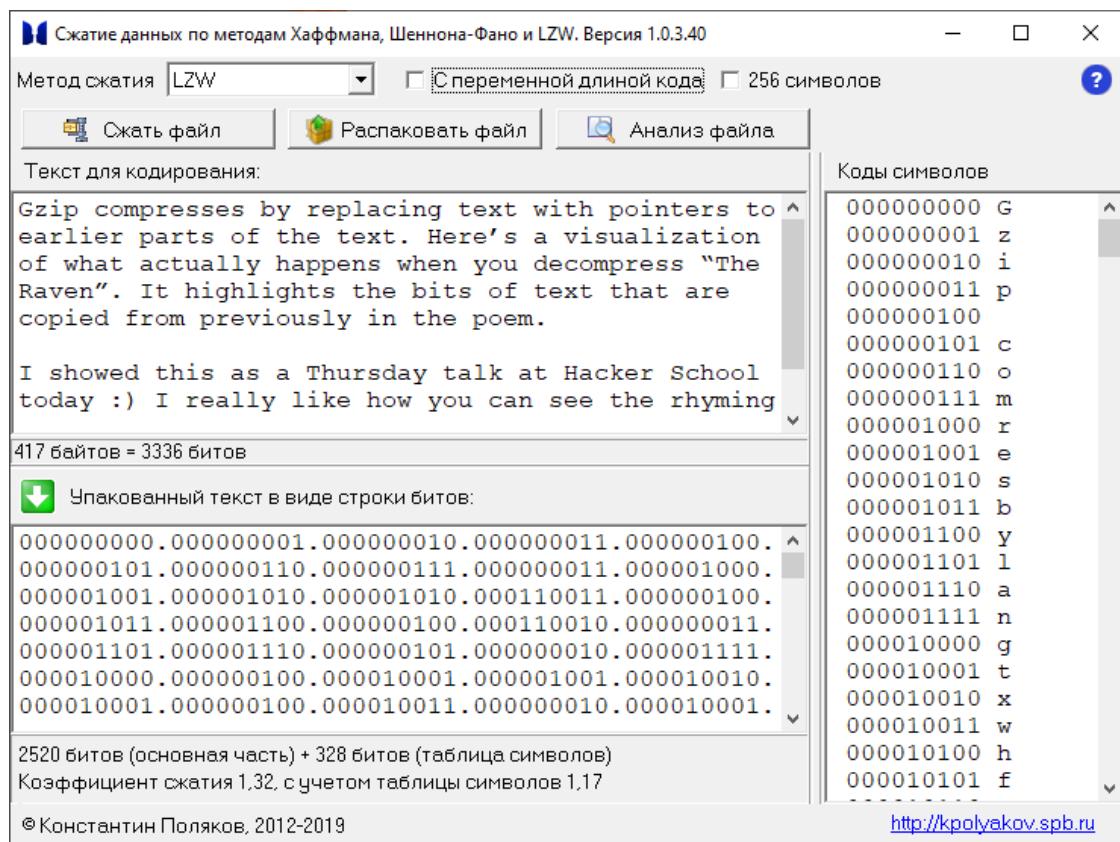


Рисунок 3 – Интерфейс пользователя тренажера «Huffman»

В данном решении реализована возможность изменения параметров алгоритма LZW. Аналогичные функции следует добавить при реализации собственного решения.

#### 1.1.4 CodSH

Электронное обучающее пособие CodSH позволяет освоить алгоритмы оптимального побуквенного кодирования методами Хаффмана и Шеннона-Фано, не прибегая к построению деревьев. Программа позволяет загрузить алфавит из библиотеки или ввести вручную. Визуальная среда дает возможность наблюдать пошаговое выполнение алгоритмов построения кода, а также возможность ввести код самостоятельно [4]. Авторами данной

программы являются Родион Кирюхин и Сергей Татаринцев. На рисунках 4 и 5 представлены главное окно программы и окно кодирования методом Шеннона-Фано соответственно.

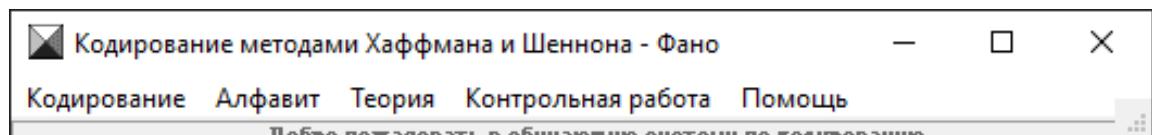


Рисунок 4 – Главное окно программы CodSH

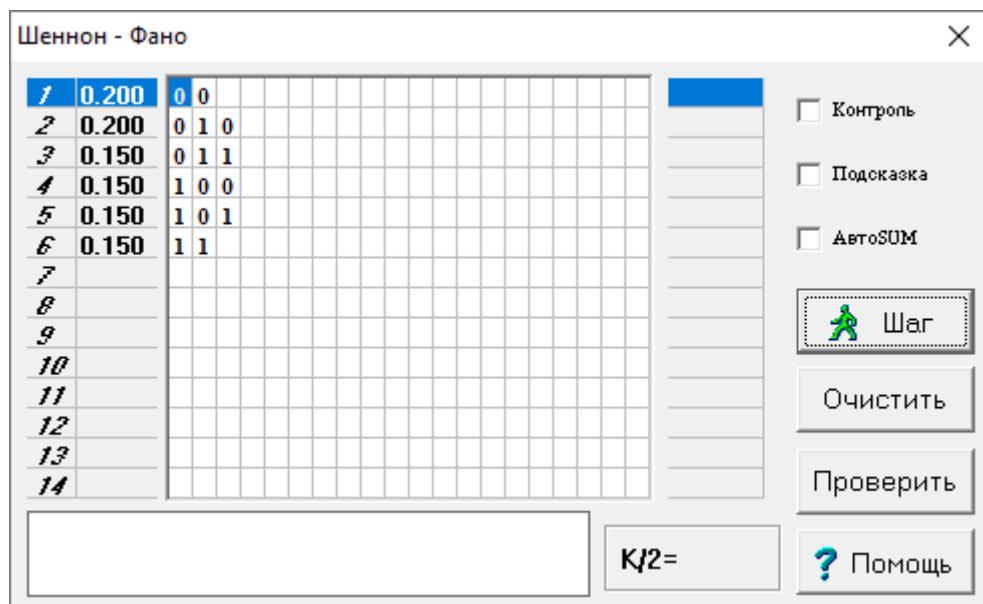


Рисунок 5 – Окно кодирования методом Шеннона-Фано

Главное окно программы выглядит и работает не совсем корректно – текстовое поле видно только на половину строки, ввиду чего невозможно разобрать текст, а также наблюдается коллизия кнопок меню и текстового поля. При этом изменить размер окна невозможно. Язык интерфейса только русский. Нет информации о программной составляющей программы – язык, код, архитектура.

Приложение имеет внутри инструкцию по использованию, а также предоставляет необходимую теоретическую справку для выполнения задания студентом. В целом, данная программа полностью выполняет свои учебные функции по получению навыков кодирования.

### 1.1.5 LZ Dictionary Algorithms

Данное приложение, реализованное на языке Java версии 1.2, наглядно демонстрирует работу словарных алгоритмов сжатия, а именно LZ77, LZ78 и LZW. Авторами являются Sami Khuri и Hsiu-Chin Hsu [5]. На рисунках 6 и 7 представлены главное окно программы и интерфейс пользователя на примере окна с алгоритмом LZ78 соответственно.



Рисунок 6 – Главное окно LZ Dictionary Algorithms

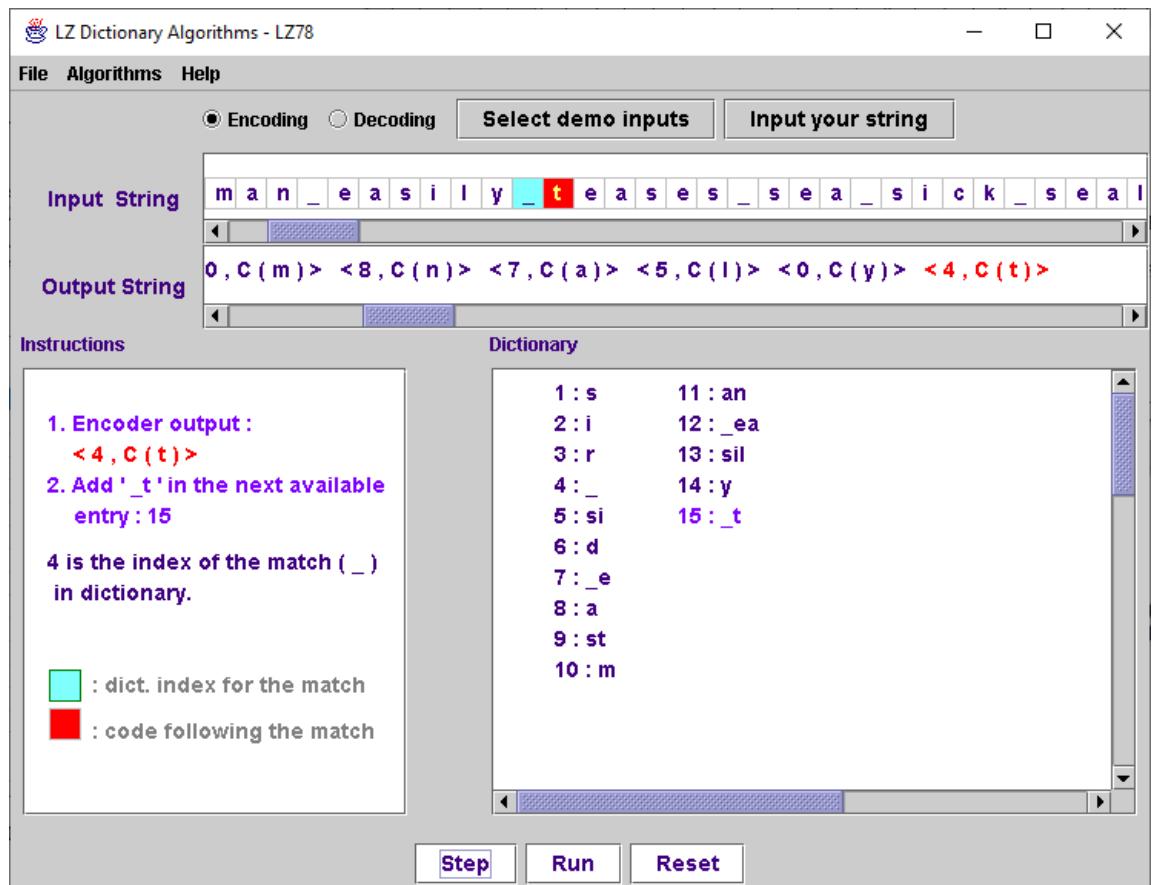


Рисунок 7 – Интерфейс пользователя LZ Dictionary Algorithms

Данная программа может выполнять следующие действия:

- кодировать и декодировать текст;
- выбор алгоритма (LZ77, LZ78 или LZW);
- выполнять алгоритм пошагово;
- демонстрировать сформированный словарь на каждом шаге алгоритма;
- использовать в качестве входных данных введенную пользователем строку;
- выбирать размер буфера для алгоритма LZ77.

Приложение содержит справочную информацию по словарным методам сжатия, а также инструкцию по использованию и описание работы алгоритма на каждом шагу при ручном воспроизведении.

Программа распространяется в виде проекта, который можно запустить как на Windows, так и на Linux, а также имеет модульную архитектуру – проект состоит из множества классов, каждый из которых отвечает за конкретное действие. Исходный код недоступен.

Помимо прочего, в данном решении не предусмотрены определение степени сжатия и возврат к предыдущим шагам. Интерфейс пользователя только на английском языке.

### **1.1.6 Run Length Encoding**

Данное решение, как и рассмотренное в п. 1.1.5, написано на языке Java версии 1.2 теми же авторами, а именно Sami Khuri и Hsiu-Chin Hsu [5]. Программа демонстрирует работу алгоритма RLE с различными способами «обхода» исходного файла. Главное окно программы представлено на рисунке 8.

Данное приложение может выполнять следующие действия:

- наглядно демонстрировать «проход» по файлу с разной скоростью;
- пошагово исполнять алгоритм;
- работать с заранее заготовленными файлами;
- определять степень сжатия.

«Обход» файла осуществляется с помощью одного из пяти алгоритмов на выбор:

- ряд за рядом;
- столбец за столбцом;
- зигзаг;
- кривая Гильберта;
- кривая Серпинского.

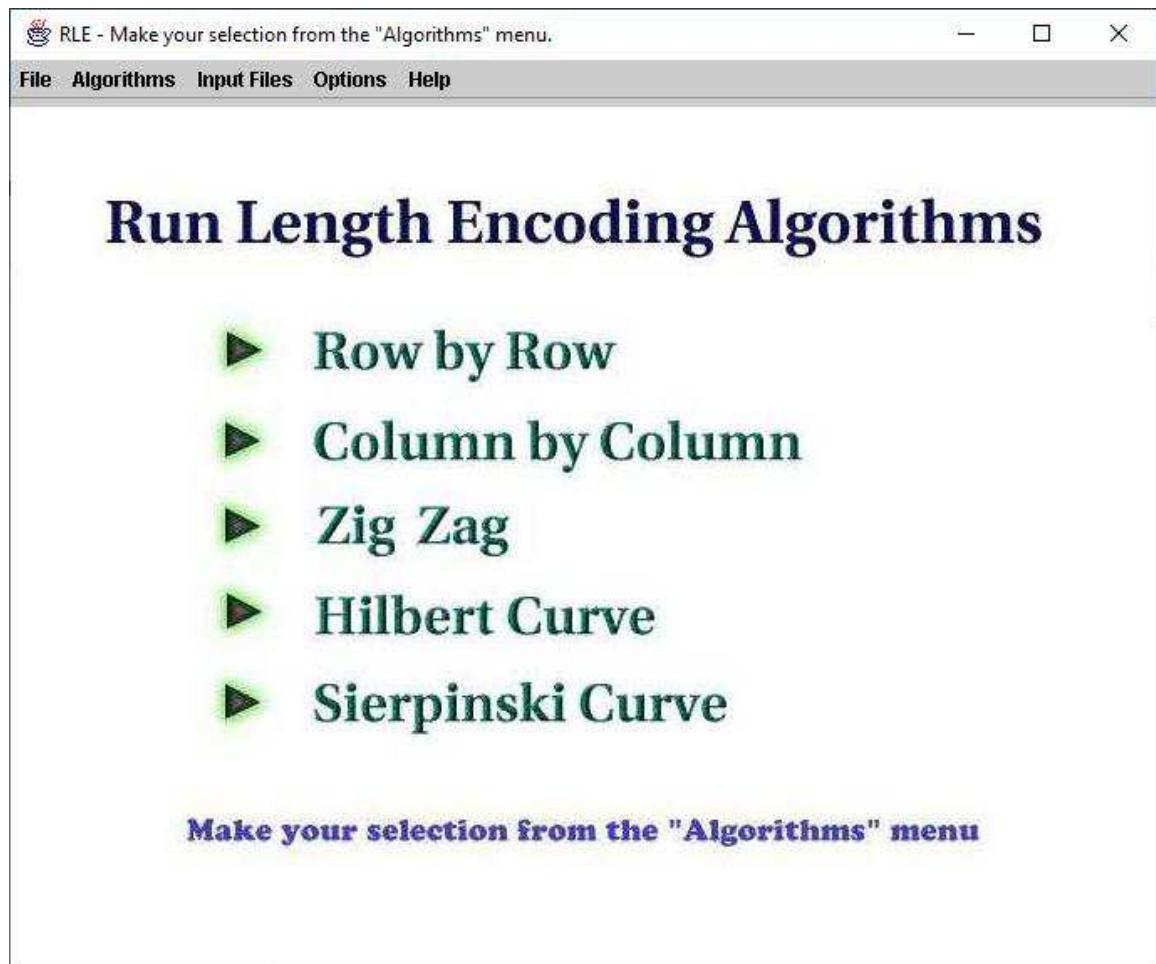


Рисунок 8 – Главное окно программы Run Length Encoding

На рисунке 9 продемонстрирован «обход» файла зигзагом. Заготовленные файлы в данной программе представляют собой битовые массивы различных размеров.

Приложение содержит справочную информацию об алгоритме RLE и инструкцию по использованию.

Программа имеет модульную архитектуру и распространяется в виде проекта, который можно запустить как на Windows, так и на Linux. Однако исходный код недоступен.

В данном решении не предусмотрен возврат к предыдущим шагам. Интерфейс пользователя только на английском языке.

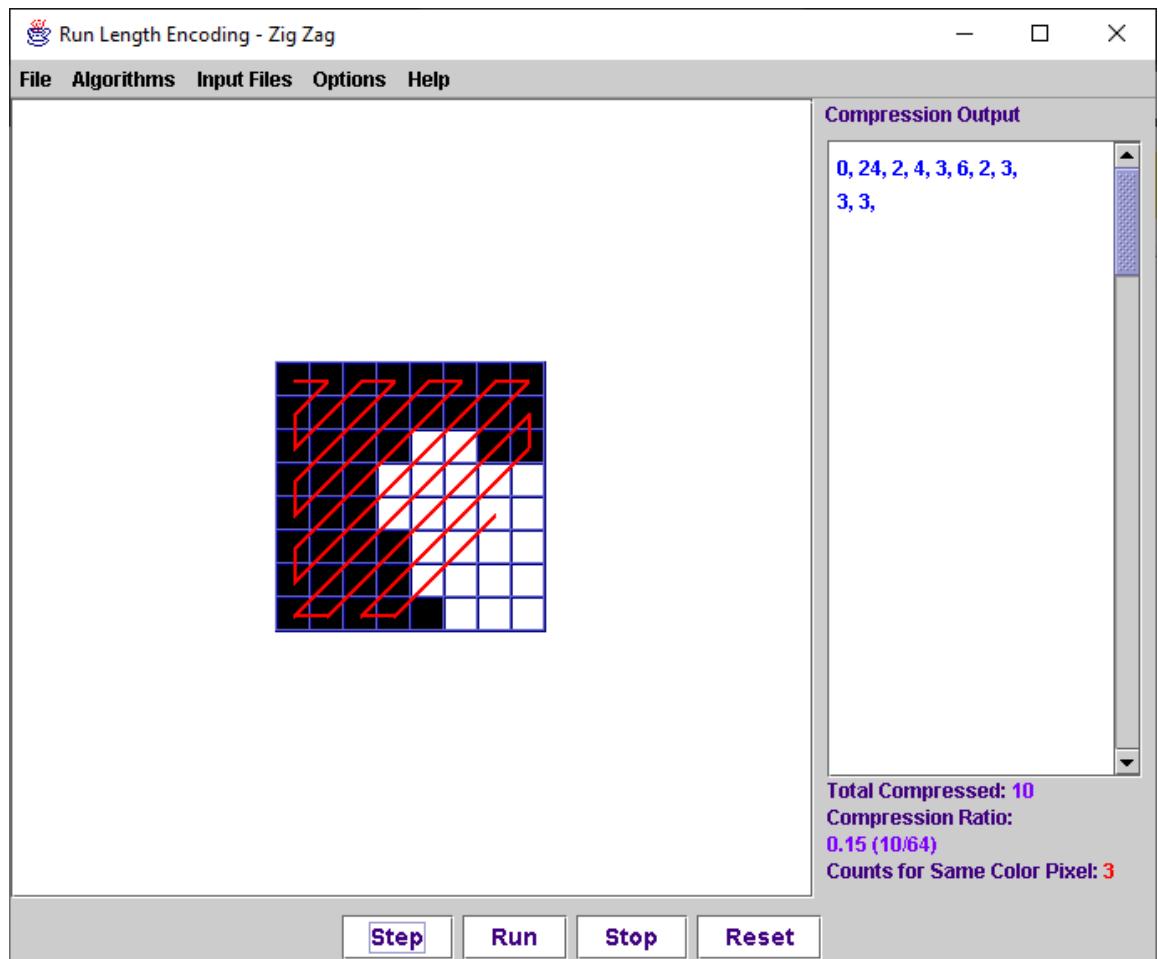


Рисунок 9 – «Обход» файла зигзагом

### 1.1.7 Dasher

Данная программа является информационно-эффективным интерфейсом для ввода текста естественными жестами. Dasher конкурентоспособен в тех случаях, когда речь идет о вводе текста без клавиатуры, например, с помощью джойстика, мыши, тачпада. Разработчиком является The Dasher Project [6]. Интерфейс программы представлен на рисунке 10.

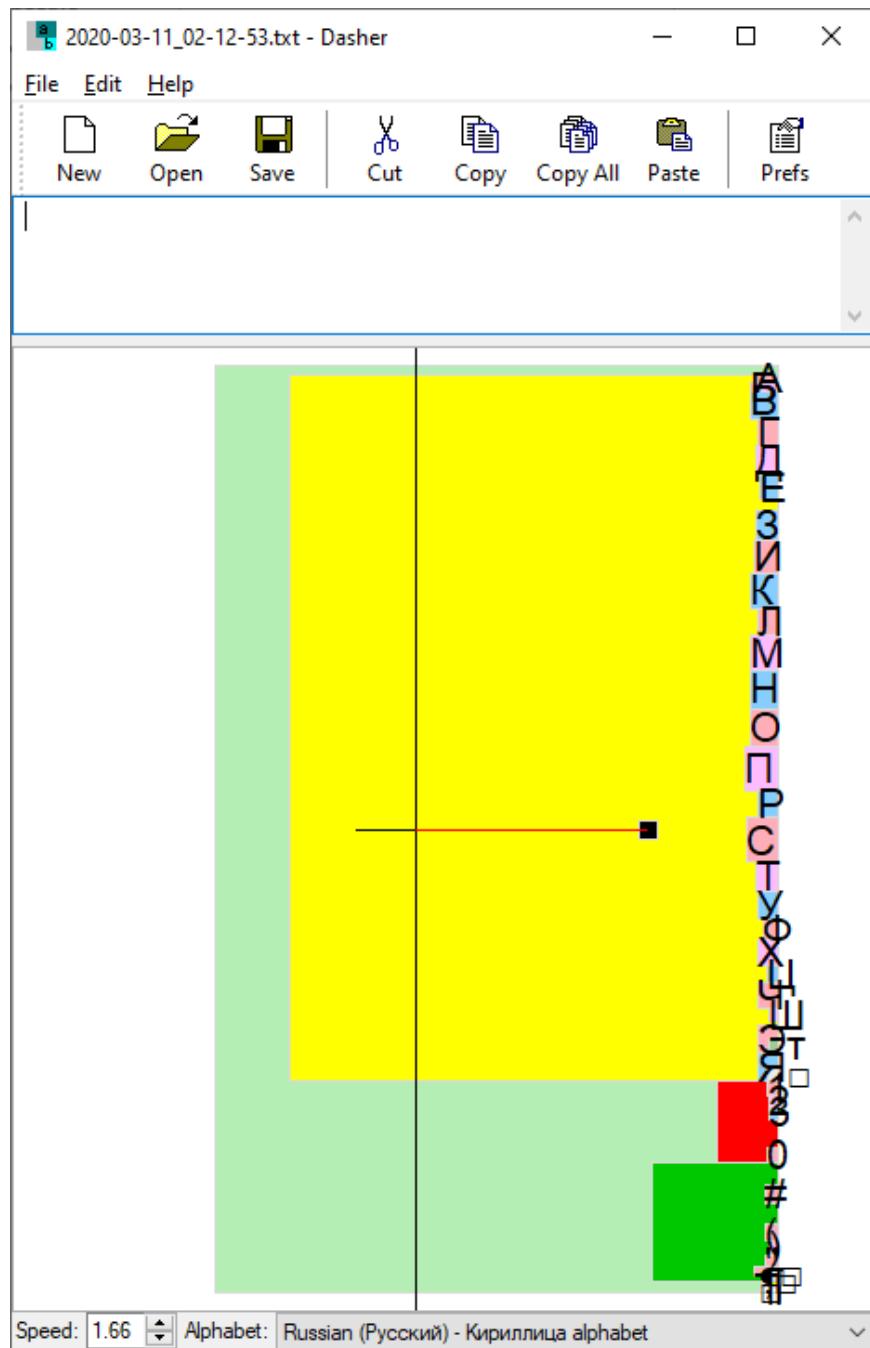


Рисунок 10 – Интерфейс Dasher

Данное приложение не демонстрирует ни один из методов сжатия, однако в алгоритме присутствует арифметическое кодирование. Как можно заметить на рисунке 11, оно позволяет графически выделять наиболее вероятные варианты продолжения слова, основываясь на данных о том или ином алфавите. Также, как утверждает автор, данный алгоритм способен

анализировать ранее написанный текст и улучшать подсказки с каждым новым словом [7].

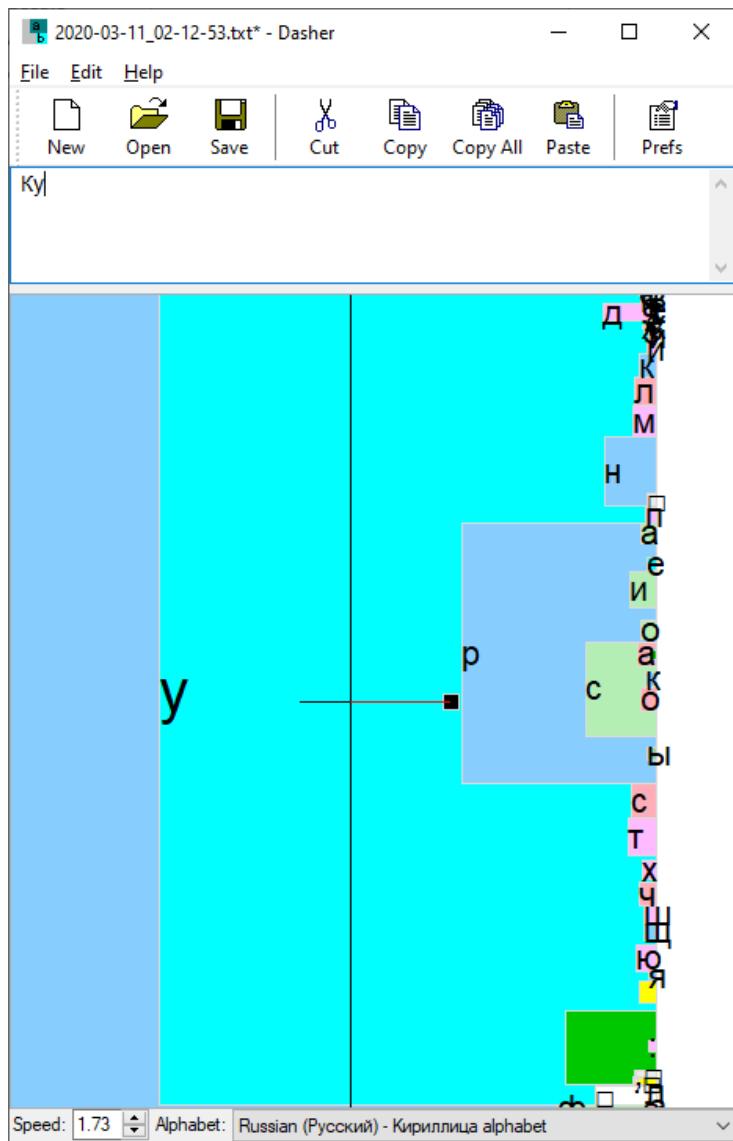


Рисунок 11 – Наиболее вероятные варианты

Интерфейс пользователя только на английском языке. Приложение работает с более чем 100 алфавитами различных языков. Внутри содержится справочная информация о работе алгоритма, инструкция к программе и прочая информация о проекте. Программа распространяется в виде дистрибутива для

ОС Windows и в качестве проекта для ОС Linux. Исходный код доступен для просмотра.

### 1.1.8 Итоги обзора

После обзора найденных решений было произведено их сравнение по ряду параметров. Результаты представлены в таблице 1.

Таблица 1 – Сравнение найденных решений

Название	Откры- тый исх. код	Крос- плат- формен- ность	Графи- ческий интерфейс пользо- вателя	Лицензия	Язык програм- мирования
Gzip in Julia	Да	Нет	Нет	Открытая*	Julia
Тренажер «RLE»	Нет	Нет	Да	Бесплатно с ограни- чениями**	Нет данных
Тренажер «Huffman»	Нет	Нет	Да	Бесплатно с ограни- чениями**	Нет данных
CodSH	Нет	Нет	Да	Нет данных	Нет данных
LZ Dictionary Algorithms	Нет	Да	Да	Нет данных	Java 1.2

Окончание таблицы 1

Название	Открытый исх. код	Кросс-платформенность	Графический интерфейс пользователя	Лицензия	Язык
Run Length Encoding	Нет	Да	Да	Нет данных	Java 1.2
Dasher	Нет	Да	Да	GNU GPL [8]	C++

\* Автор предоставляет исходный код и разрешает копировать и/или изменять его под собственные нужды [1].

\*\* Бесплатно для некоммерческого использования. Программа поставляется как есть, автор не несет ответственности за всевозможные последствия использования. Запрещено включение программы в сборники

Из рассмотренных решений LZ Dictionary Algorithms удовлетворяет большинству требований к выпускной работе, за исключением:

- открытый исходный код;
- поддержка русского и английского языков;
- возможность вернуться к предыдущим шагам;
- возможность модифицировать состояние алгоритмов вручную.

Было решено проектировать и реализовывать приложение с нуля, возможно, заимствуя часть кода из решения, рассмотренного в п. 1.1.1, а также используя в качестве примера пользовательский интерфейс из аналогов, рассмотренных в п. 1.1.3 и п. 1.1.5.

## **1.2 Выбор средств разработки**

В качестве языка программирования, на котором будет написана программа, был выбран язык C++. Во-первых, язык программирования C++ представляет собой высокоуровневый компилируемый язык программирования общего назначения со статической типизацией, который подходит для создания самых различных приложений [9]. Во-вторых, язык C++ является одним из самых популярных языков программирования [10]. Принимая во внимание вышеуказанные тезисы и то, что разрабатываемое приложение является приложением с открытым исходным кодом, оно будет более дружественно к внешним доработкам и модернизациям, чем другие языки программирования.

Графический интерфейс пользователя планируется реализовать с помощью библиотеки для C++ – Qt. Данная библиотека обладает огромным потенциалом в области создания кроссплатформенных приложений с графическим интерфейсом, к тому же она распространяется по лицензии GNU GPL (GNU General Public License), т. е. свободно для создания ПО с открытым исходным кодом [11].

## **1.3 Диаграмма прецедентов**

На рисунке 12 представлена диаграмма прецедентов.

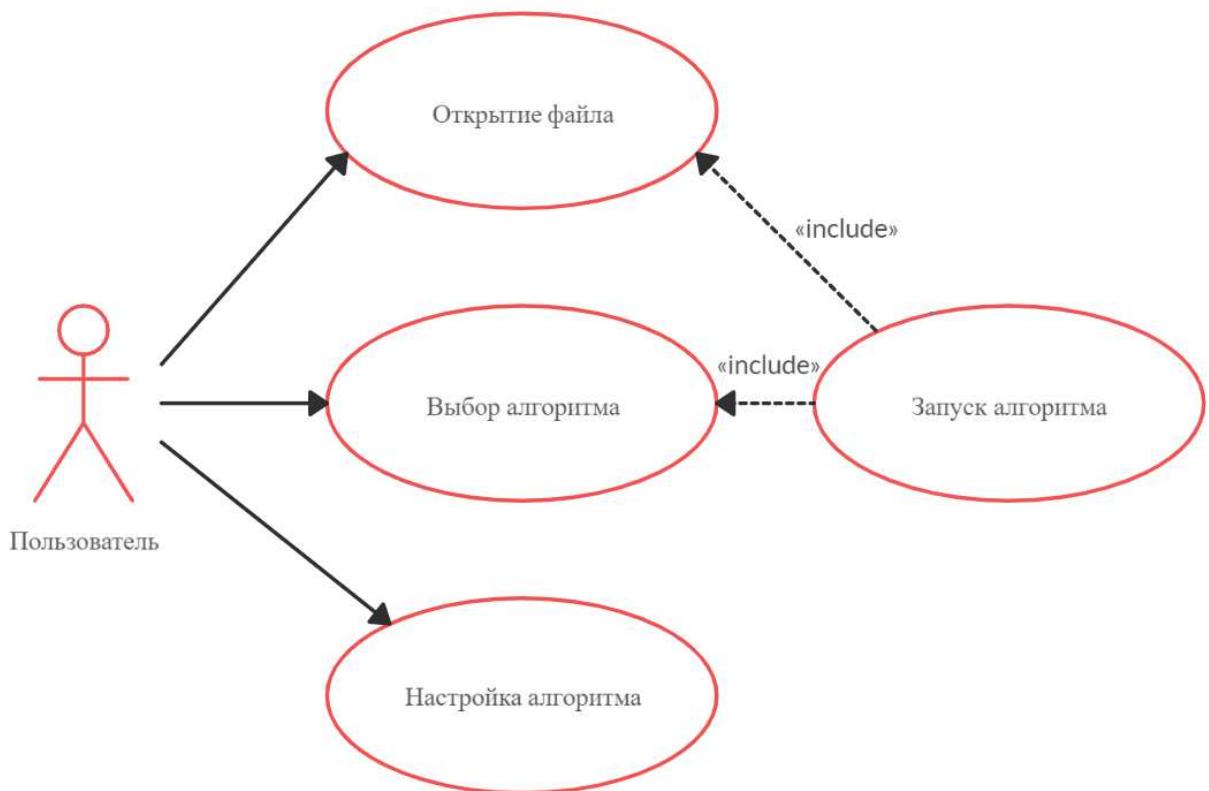


Рисунок 12 – Диаграмма прецедентов

Текстовое описание прецедентов:

**Название прецедента:** Открытие файла.

Предусловия: пользователь находится на главном экране, нажимает кнопку «Открыть» в меню «Файл», при этом программа не выполняет какой-либо процесс.

Основной сценарий:

- A. Создается окно выбора текстового файла.
- B. Пользователь выбирает файл и нажимает кнопку «OK».
- C. Выбранный файл открывается только для чтения.
- D. Заголовок окна обновляется, отображая имя открытого файла.

Постусловия: появляется возможность увидеть содержимое файла, выполняется одно из условий для запуска алгоритма.

Условие ввода в действие альтернативных сценариев

Условие 1. Пользователь закрыл окно выбора файла, не подтвердив выбор, либо файл не может быть открыт для чтения.

Альтернативный сценарий:

А. Появляется окно, информирующее о том, что файл не открыт.

Постусловия: программа возвращается в исходное состояние.

**Название прецедента:** Выбор алгоритма.

Предусловия: пользователь находится на главном экране, нажимает одну из трех кнопок (LZ77, LZ78, LZW) в выпадающем меню «Алгоритм» в меню «Настройки».

Основной сценарий:

А. В качестве активного алгоритма выступает выбранный.

Постусловия: выполняется одно из условий запуска алгоритма.

**Название прецедента:** Настройка алгоритма.

Предусловия: пользователь находится на главном экране с выбранным алгоритмом, нажимает кнопку «Настройка алгоритма» в меню «Настройки».

Основной сценарий:

А. Создается окно настройки выбранного алгоритма.

Б. Пользователь вводит свои настройки и нажимает кнопку «OK».

С. Введенные параметры передаются классу выбранного алгоритма.

Постусловия: параметры алгоритма изменены.

Условие ввода в действие альтернативных сценариев

Условие 1. Пользователь нажал кнопку «Отмена» или закрыл окно.

Альтернативный сценарий:

А. Пользователь информируется о том, что настройки не сохранены.

Постусловия: параметры алгоритма остаются исходными.

**Название прецедента:** Запуск алгоритма.

Предусловия: пользователь находится на главном экране с выбранными файлом и алгоритмом, нажимает кнопку «Старт» на панели инструментов.

Основной сценарий:

- A. Выполняется процесс компрессии в автоматическом режиме.
- B. Пользователю предлагается создать выходной файл.
- C. Выполняется процесс декомпрессии в автоматическом режиме.
- D. Пользователю предлагается создать второй выходной файл.

Постусловие: программа выполнила компрессию и декомпрессию выбранным алгоритмом.

Условие ввода в действие альтернативных сценариев

Условие 1. Пользователь прервал процедуру создания и сохранения закодированного выходного файла.

Альтернативный сценарий:

- A. Программа возвращается в исходное состояние.

Постусловия: программа выполнила компрессию выбранным алгоритмом, но результат не сохранен.

Условие 2. Пользователь прервал процедуру создания и сохранения раскодированного выходного файла.

Альтернативный сценарий:

- A. Программа возвращается в исходное состояние.

Постусловия: программа выполнила компрессию и декомпрессию выбранным алгоритмом, но результат декомпрессии не сохранен.

Условие 3. Пользователь нажал кнопку «Следующий шаг» на панели инструментов.

Альтернативный сценарий:

- A. Программа выполняет один шаг компрессии выбранным методом.
- B. Пользователь пошагово производит компрессию.
- C. Пользователю предлагается создать выходной файл.
- D. Пользователь пошагово производит декомпрессию.
- E. Пользователю предлагается создать второй выходной файл.

Постусловия: программа пошагово выполнила компрессию и декомпрессию выбранным алгоритмом под управлением пользователя.

## 2 Проектирование и реализация приложения

### 2.1 Интерфейс программы

При запуске приложения появляется главное окно приложения, изображенное на рисунке 13. С помощью меню настроек пользователь может выбрать один из трех методов сжатия – LZ77, LZ78 или LZW. После выбора одного из методов сжатия заголовок окна сменится на соответствующий, отображая выбранный метод. Открыть файл для сжатия можно через меню «Файл». После выбора файла заголовок окна будет отображать выбранный метод сжатия, либо исходный заголовок окна, и имя выбранного файла. С помощью кнопки на панели инструментов или меню настроек можно вызвать окно конфигурации алгоритма, изображенное на рисунке 14.

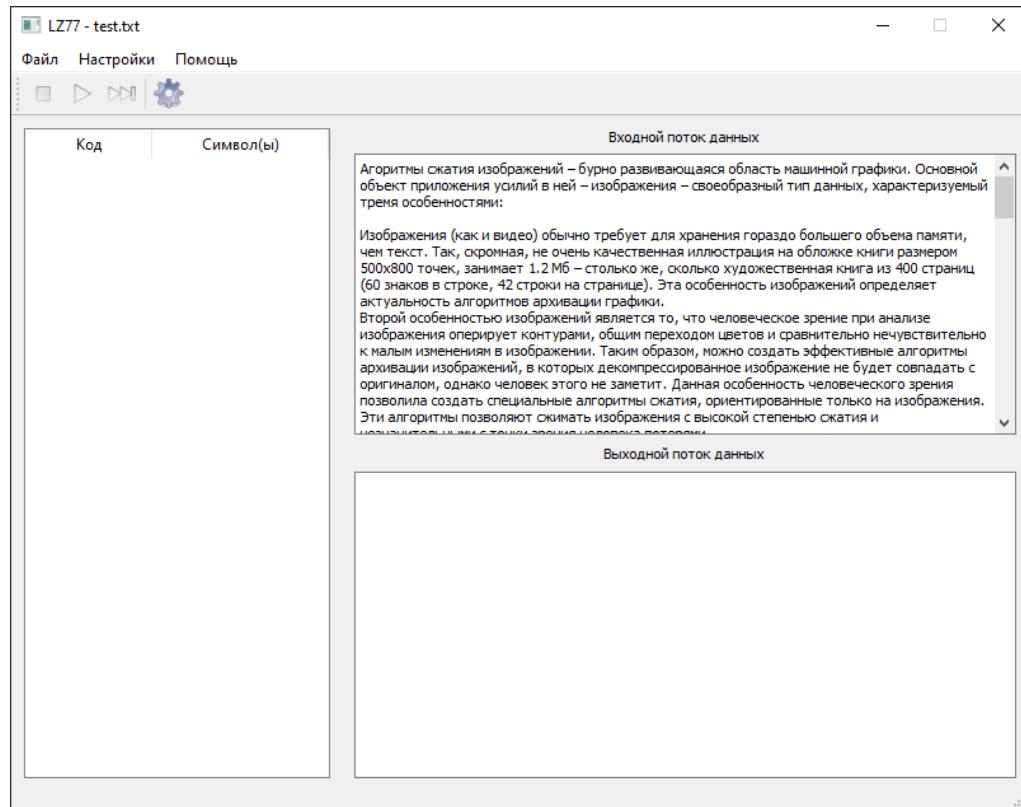


Рисунок 13 – Главное окно программы

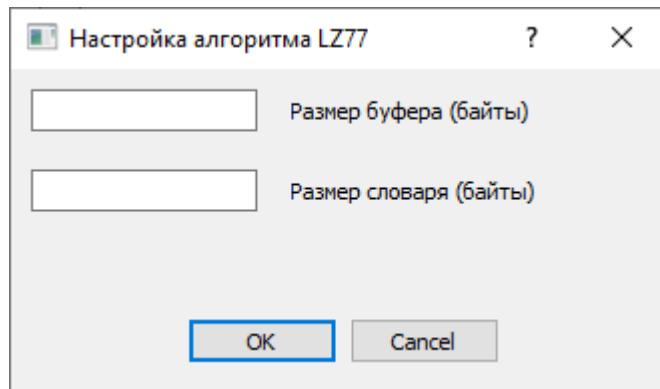


Рисунок 14 – Окно конфигурации алгоритма LZ77

Настройка алгоритма необязательна. Если пользователь введет некорректные настройки алгоритма или не введет их вовсе, то программа будет использовать настройки по умолчанию. Количество настроек зависит от активного в данный момент метода сжатия.

После выбора алгоритма и открытия исходного файла, пользователь может использовать кнопки на панели инструментов для управления процессом. Предусмотрены запуск алгоритма в автоматическом режиме, ручное пошаговое управление алгоритмом, полная остановка работы алгоритма и возврат к исходному состоянию.

## 2.2 Структура программы

Работа главного окна приложения реализована с помощью класса `MainWindow`.

Для реализации каждого из алгоритмов был создан отдельный класс: `lz77`, `lz78` и `lzw` соответственно для алгоритмов LZ77, LZ78, LZW.

В классе реализации каждого алгоритма сжатия данных присутствуют методы для компрессии и декомпрессии – `encode()` и `decode()` соответственно.

Так как в программе предусмотрены ручной (пошаговый) и автоматический режимы работы алгоритмов, в классах присутствует атрибут mode, принимающий разные значения в зависимости от вызывающего слота (start() или nextStep()), который отвечает за режим работы декомпрессора и декомпрессора. В функциях encode() и decode() каждого класса работа алгоритма реализована в виде цикла. При значении mode = false, цикл выполняется до полного завершения процесса компрессии или декомпрессии. При значении mode = true, цикл совершает одну итерацию и программа ничего не выполняет, пока не получит сигнал nextStep() или stop().

Реализация каждого алгоритма представляет собой собственную адаптацию реализаций алгоритмов с открытым исходным кодом под возможности программы [12;13;14].

При вызове методов start() или nextStep() любого из классов алгоритмов, им передаются настройки, которые можно задать через отдельное окно. Реализация этого диалогового окна также выделена в отдельный класс ConfigDialog. В этом классе находятся три конструктора диалогового окна – для каждого алгоритма свой конструктор, так как у разных алгоритмов разные параметры, которые можно менять для изменения как поведения алгоритма, так и его эффективности. Также в классе ConfigDialog присутствуют методы для передачи введенных пользователем параметров, либо параметров по умолчанию, соответствующим классам.

## 3 Инструкции

### 3.1 Инструкция пользователя

При открытии программы появляется главное окно. Для начала работы нужно открыть исходный файл через меню «Файл» с помощью диалогового окна выбора файла, которое появляется при нажатии кнопки «Открыть» (рисунок 15). Цифрами на рисунке обозначен порядок действий.

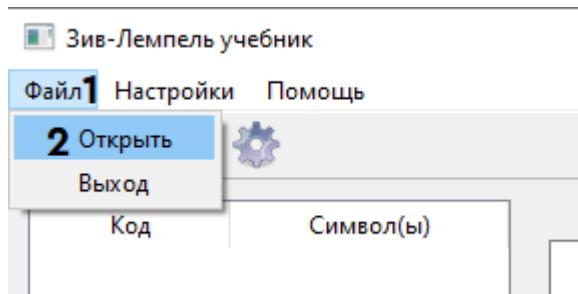


Рисунок 15 – Кнопка «Открыть» в меню «Файл»

После открытия файла в поле «Входной поток данных» появится содержимое открытого файла (рисунок 16).

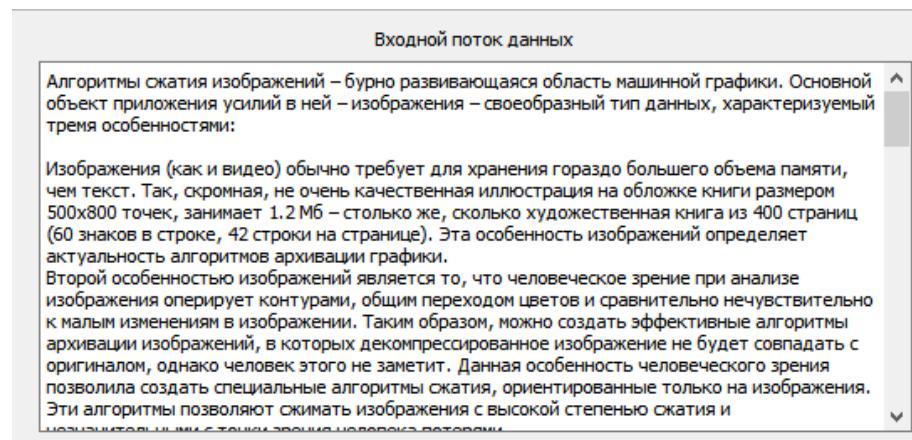


Рисунок 16 – Содержимое открытого файла

Далее следует выбрать алгоритм сжатия, с помощью которого будут происходить процессы компрессии и декомпрессии файла. Сделать это можно через меню «Настройки», нажав на одну из кнопок в выпадающем меню «Алгоритм» (рисунок 17). Цифрами на рисунке обозначен порядок действий.

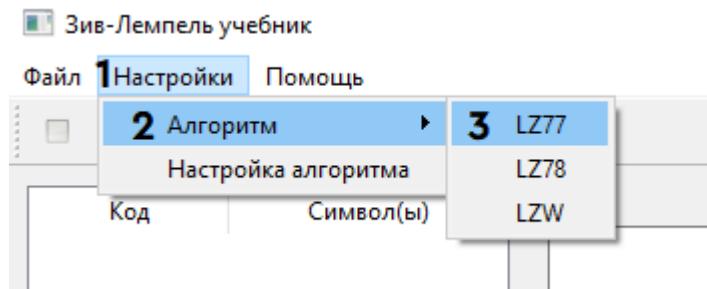


Рисунок 17 – Выбор алгоритма сжатия

После выбора алгоритма сжатия будет доступна его настройка. Вызвать диалоговое окно настройки алгоритма можно через меню «Настройки», нажав на кнопку «Настройка алгоритма». Например, при выбранном алгоритме LZ77 откроется диалоговое окно, изображенное на рисунке 18.

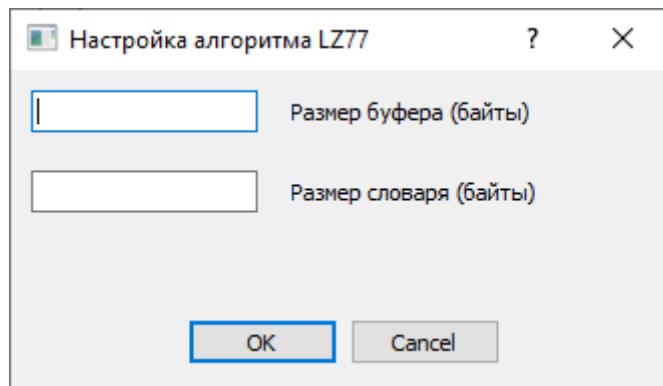


Рисунок 18 – Диалоговое окно изменения параметров алгоритма

После ввода параметров и их применения с помощью нажатия кнопки «OK» введенные параметры применяются к выбранному алгоритму. Процедура

настройки алгоритма необязательна для его запуска. В этом случае алгоритм запустится с исходными предустановленными параметрами.

Запустить алгоритм можно в двух режимах – автоматическом и ручном. Автоматический режим запускается кнопкой «Старт», отмеченной на рисунке 19 буквой «А». Ручной режим запускается кнопкой «Следующий шаг», отмеченной на рисунке 19 буквой «Б». Этой же кнопкой осуществляется управление алгоритмом (т. е. движение по шагам алгоритма вперед). В ручном режиме доступна кнопка «Стоп», отложенная на рисунке 19 буквой «В». При нажатии на неё выполнение алгоритма прекратится, и программа вернется в исходное состояние.

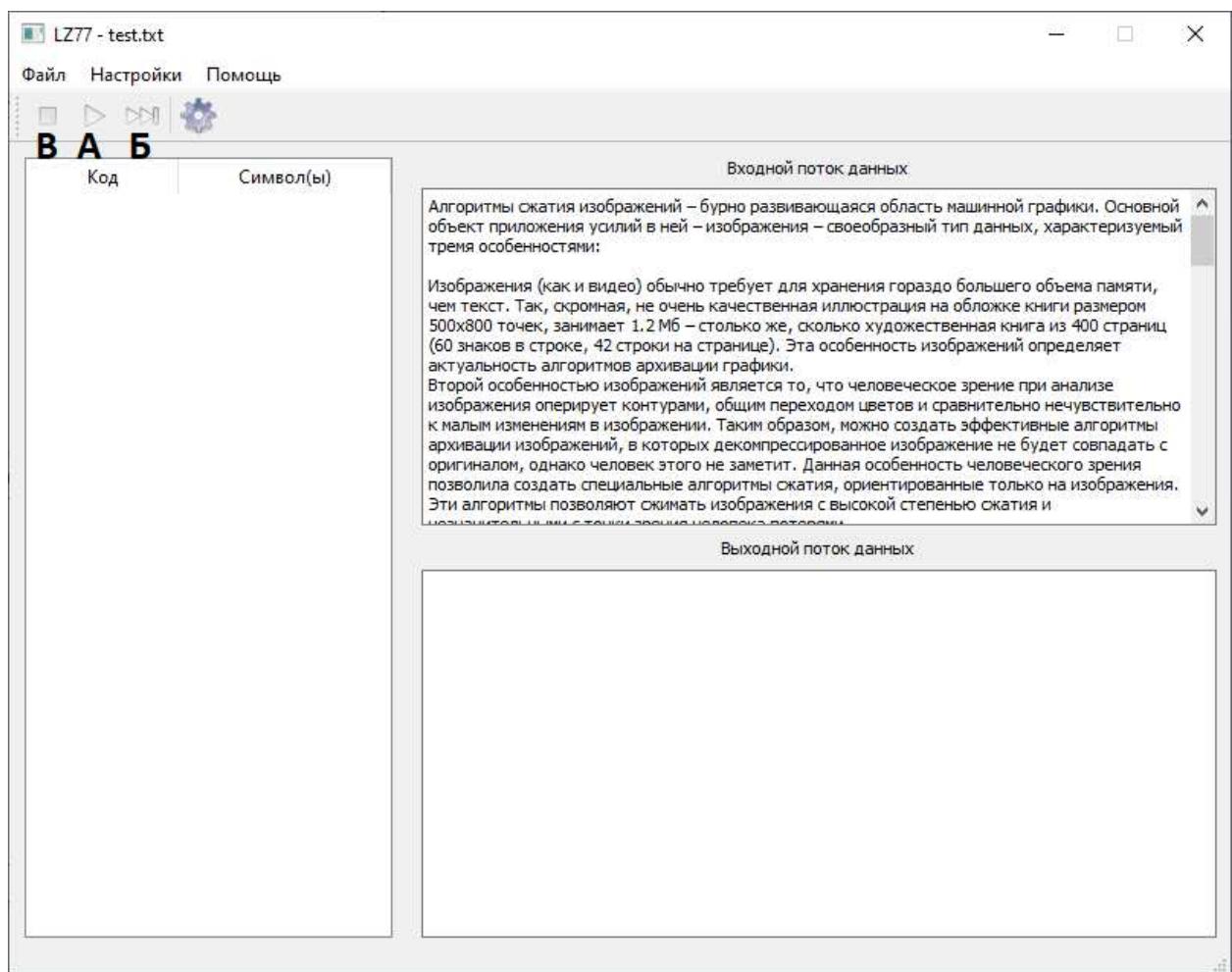


Рисунок 19 – Кнопки на панели инструментов

### **3.2 Инструкция разработчика**

Для компиляции проекта следует использовать библиотеку Qt версии 5.12 или новее. В качестве IDE рекомендуется использовать Qt Creator (рекомендуемая версия 4.10.2). Загрузить библиотеку Qt и среду разработки Qt Creator можно на официальном сайте Qt [15]. В качестве компилятора рекомендуется использовать GCC версии 7.3.0 или новее.

Список файлов и их назначение:

- mainwindow.h и mainwindow.cpp – реализация класса MainWindow;
- lz77.h и lz77.cpp – описание и реализация алгоритма LZ77;
- lz78.h и lz78.cpp – описание и реализация алгоритма LZ78;
- lzw.h и lzw.cpp – описание и реализация алгоритма LZW;
- configdialog.h и configdialog.cpp – реализация класса ConfigDialog;
- mainwindow.ui и configdialog.ui – файлы форм, из которых формируются окна программы;
- token.h – заголовочный файл структуры, используемой при кодировании информации;
- resources.qrc – файл, содержащий в себе пути к иконкам;
- VKR.pro – информация для сборки проекта с помощью qmake;
- icons – каталог, содержащая в себе файлы иконок, используемых приложением.

Интерфейс главного окна содержит в себе два текстовых поля без возможности редактировать текст – inputStream и outputStream – соответственно для отображения входного и выходного потока данных. Для отображения текста в этих полях используются сигналы setInText() и setOutText(). Также в главном окне есть элемент таблицы, используемый для отображения кодов символов или строк. Для добавления элементов в таблицу используется сигнал setDictSymb().

Для добавления другого алгоритма в программу следует убедиться, что добавляемый алгоритм можно продемонстрировать, используя описанные выше элементы интерфейса. Далее нужно добавить отдельный класс алгоритма, в котором реализованы алгоритмы компрессии и декомпрессии тем же образом, что и уже существующие, т. е. в цикле, в котором можно выполнить одну итерацию за одно нажатие кнопки. Также нужно добавить новый конструктор и методы передачи параметров алгоритма в классе ConfigDialog. Интерфейс также нужно доработать, добавив в выбор алгоритмов новый экземпляр и добавить новое значение в функцию обновления заголовка окна.

## **ЗАКЛЮЧЕНИЕ**

В результате выполненной работы было спроектировано и реализовано интерактивное электронное учебное пособие по алгоритмам сжатия данных.

Был разработан графический интерфейс программы, структура приложения. Адаптированы три алгоритма сжатия без потерь: LZ77, LZ78 и LZW. Была добавлена возможность настраивать алгоритмы.

Разработанное приложение не соответствует следующим требованиям:

- поддержка английского языка;
- демонстрация состояния компрессора и декомпрессора вместе (одновременно);
- возможность модифицировать состояние алгоритмов вручную;
- возможность вернуться к предыдущим шагам;
- возможность определения степени сжатия.

Направления для дальнейшей доработки:

- доработать приложение до состояния соответствия требованиям;
- добавление новых алгоритмов;
- добавление возможности выбора скорости автоматического воспроизведения алгоритма.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. gzip + poetry = awesome [Электронный ресурс]: Julia Evans. – Режим доступа: <https://jvns.ca/blog/2013/10/24/day-16-gzip-plus-poetry-equals-awesome/> (дата обращения: 03.03.2020).
2. gzip + poetry = awesome [Электронный ресурс]: YouTube. – Режим доступа: <https://www.youtube.com/watch?v=SWBkneyTyPU> (дата обращения: 09.03.2020).
3. Тренажеры для изучения алгоритмов сжатия [Электронный ресурс]: сайт Константина Полякова. – Режим доступа: <https://www.kroyakov.spb.ru/prog/compress.htm> (дата обращения: 05.03.2020).
4. Тренажеры по методам оптимального кодирования [Электронный ресурс]: Система электронного обучения СФУ. – Режим доступа: <https://e.sfu-kras.ru/mod/assign/view.php?id=712764> (дата обращения: 06.03.2020).
5. Sami Khuri's Visualization and Animation Packages [Электронный ресурс]: Sami Khuri: Educational tools. – Режим доступа: <https://www.cs.sjsu.edu/~khuri/animation.html> (дата обращения: 09.03.2020).
6. What is Dasher for? [Электронный ресурс]: Inference Group: Dasher Project. – Режим доступа: <http://www.inference.org.uk/dasher/DasherSummary.html> (дата обращения 10.03.2020).
7. How does Dasher works? [Электронный ресурс]: Inference Group: Dasher Project. – Режим доступа: <http://www.inference.org.uk/dasher/DasherSummary2.html> (дата обращения: 10.03.2020).
8. dasher-project [Электронный ресурс]: GitHub. – Режим доступа: <https://github.com/dasher-project/dasher> (дата обращения: 10.03.2020).
9. C++ | Введение [Электронный ресурс]: Сайт о программировании. – Режим доступа: <https://metanit.com/cpp/tutorial/1.1.php> (дата обращения 05.03.2020).

10. index | TIOBE [Электронный ресурс]: The Software Quality Company. – Режим доступа: <https://www.tiobe.com/tiobe-index/> (дата обращения 05.03.2020).
11. Why Qt – Get started today [Электронный ресурс]: The Qt Company. – Режим доступа: <https://www.qt.io/why-qt> (дата обращения: 05.03.2020).
12. MihrimatriX/LZ77-QT [Электронный ресурс]: GitHub. – Режим доступа: <https://github.com/MihrimatriX/LZ77-QT> (дата обращения 10.06.2020).
13. Обзор алгоритмов сжатия без потерь [Электронный ресурс]: Алгоритмы сжатия. – Режим доступа: [https://mf.grsu.by/UchProc/livak/po/comprsite/pract\\_lz78.html#Алгоритм%20LZ78](https://mf.grsu.by/UchProc/livak/po/comprsite/pract_lz78.html#Алгоритм%20LZ78) (дата обращения 10.06.2020).
14. Обзор алгоритмов сжатия без потерь [Электронный ресурс]: Алгоритмы сжатия. – Режим доступа: [https://mf.grsu.by/UchProc/livak/po/comprsite/pract\\_lzw.html#Алгоритм%20LZW](https://mf.grsu.by/UchProc/livak/po/comprsite/pract_lzw.html#Алгоритм%20LZW) (дата обращения 10.06.2020).
15. Get Qt – Download now [Электронный ресурс]: The Qt Company. – Режим доступа: <https://www.qt.io/download> (дата обращения 20.06.2020).

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
**«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой  
*О. В. Непомнящий*  
подпись

« \_\_\_\_\_ » 2020 г.

## БАКАЛАВРСКАЯ РАБОТА

09.03.01 – «Информатика и вычислительная техника»

Интерактивное электронное учебное пособие по алгоритмам сжатия данных

Руководитель

*Постников*  
подпись, дата

доцент,  
канд. техн. наук

А. И. Постников

Выпускник

*Дедик*  
29.06.2020г.  
подпись, дата

Д. А. Дедик

Консультант

*Пушкарев*  
подпись, дата

ст. преподаватель

К. В. Пушкарев

Нормоконтролер

*Постников*  
подпись, дата

А. И. Постников

Красноярск 2020