|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт искусственного интеллекта | | |
| Кафедра программного обеспечения систем радиоэлектронной аппаратуры | | |

|  |  |
| --- | --- |
| **ОТЧЕТ** | |
| **ПО НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ** | |
|  | |
| **Тема: «Парсинг оглавления книги»** | |
| Студент группы           КМБО-02-22 | *И.О.Кочнов* |
| Руководитель практики | *А.В.Завьялов* |

|  |  |  |
| --- | --- | --- |
| «Отчет представлен к рассмотрению» | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2024 г. | *(подпись студента)* |
|  |  |  |
| «Отчет утвержден.  Допущен к защите» | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2024 г. | *(подпись руководителя)* |

Москва 2024

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc168664497)

[1 ПОСТАНОВКА ЗАДАЧИ 3](#_Toc168664498)

[1.1 Задание 3](#_Toc168664499)

[2 ОПИСАНИЕ РАБОТ И РЕЗУЛЬТАТОВ 3](#_Toc168664500)

[2.1 Используемые средства разработки 3](#_Toc168664501)

[2.2 Инструкция по использованию 3](#_Toc168664502)

[2.3 Содержание и результат выполненных работ 5](#_Toc168664503)

[2.3 Доступ к результатм. 9](#_Toc168664504)

[ЗАКЛЮЧЕНИЕ 9](#_Toc168664505)

[СПИСОК ИСТОЧНИКОВ 9](#_Toc168664506)

# ВВЕДЕНИЕ

Создается Desktop-приложения для парсинга оглавления книги, с помощью которого можно в удобной форме получить всю информацию об разделах в книге (например, номера страниц, вложенность разделов и т.д.)

# 1 ПОСТАНОВКА ЗАДАЧИ

## Задание

Необходимо реализовать алгоритм считывания оглавления любой книги, предварительно введя ряд параметров в специальных полях, после чего результат должен быть сохранен в CSV формате.

# 2 ОПИСАНИЕ РАБОТ И РЕЗУЛЬТАТОВ

## 2.1 Используемые средства разработки

Для реализации проекты был выбран язык программирования C++, фреймворк Qt. Для имплементации кода использовался редактор Qt Creator.

## 2.2 Инструкция по использованию

При запуске пользователь увидит следующий интерфейс:

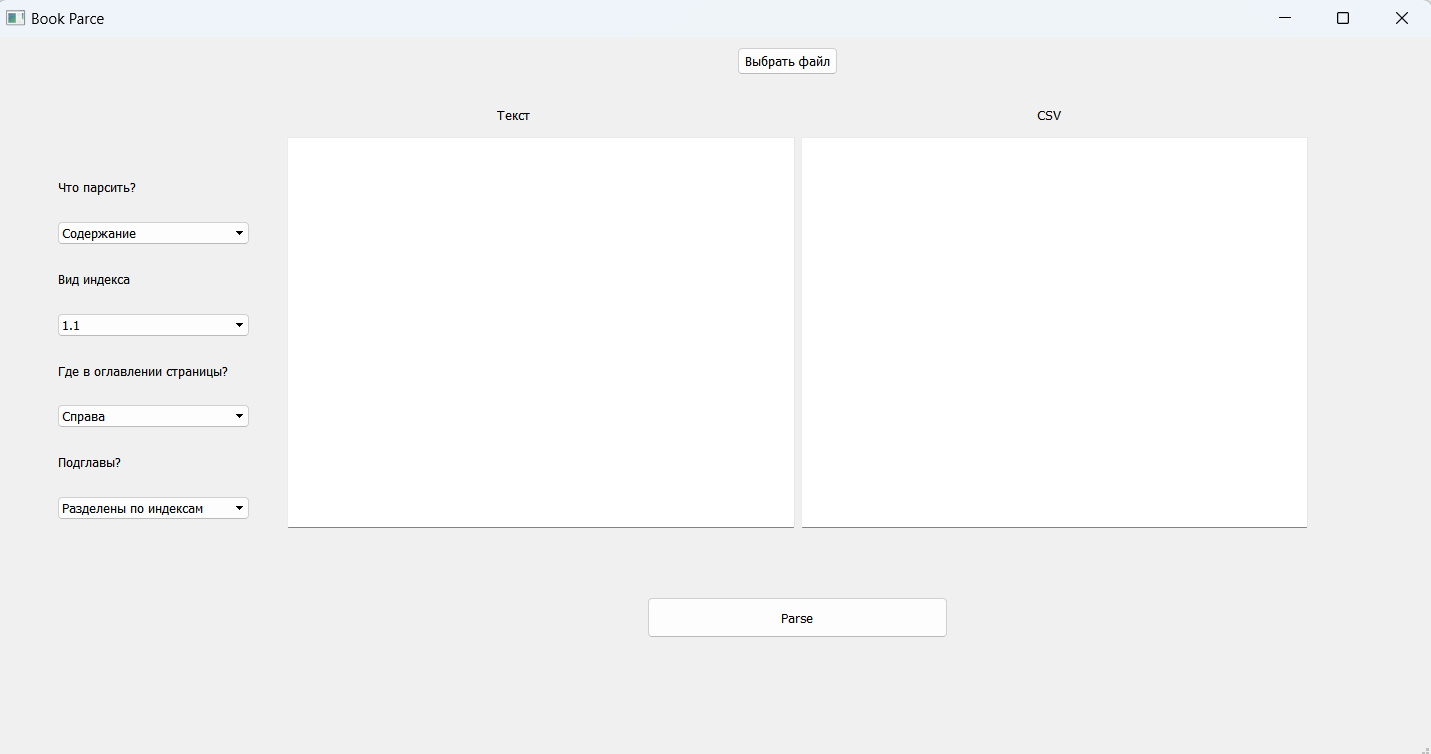


Рисунок 1 – интерфейс программы

Опишем параметры слева и их значение:

* Что парсить? (Содержание или Оглавление) – что именно будет искаться в книге
* Вид индекса (1.1, Глава 1, Часть 1 Глава 1) – вид символов или слов, которые могут стоять перед главой (если в книге присутствуют есть главы, но нет частей, то нужно выбрать второе; если в книге встречаются числовые индексы, то нужно выбрать первый вариант)
* Где в оглавлении страницы? (Слева или Справа) – где по отношению к названиям находятся номера страниц
* Подглавы? (Разделены по индексам или Идут сразу после глав) – если в книге все подглавы идут строго по нумерации, то нужно выбрать первый вариант, если же оглавление построено таким образом, что сначала идет глава, а следом за ней идут подглавы, относящиеся к ней, то следует выбрать второй вариант)

Затем нужно нажать на кнопку “Выбрать файл”, и далее, в выпадающем окне выбрать необходимый файл в формате txt. После этого следует нажать на кнопку “Parce”, отвечающую за парсинг. После вывода результатов появится кнопка “Save to Csv”, при нажатии на которую пользователю предложат выбрать папку, в которую сохранится результат в виде файла “content.csv” (пример на рисунке 2).

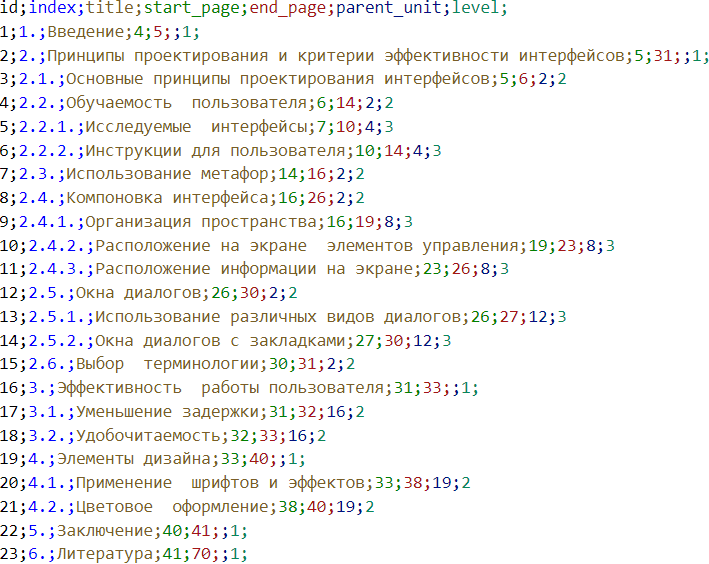


Рисунок 2 – результат работы программы

## 2.3 Содержание и результат выполненных работ

Для дальнейшего описания введем следующие термины:

* Глава – раздел книги
* Подглава – раздел книги, который содержится в другом разделе
* Индекс – чаще цифры или особые слова перед названием разделов (например, 1.1, Глава 1, Часть 1 Глава 1)

Итак, программа начинается с ввода пользователем и дальнейшего считывания программой полей QComboBox (выпадающий список) и поля “Выбрать файл”, реализованный через QPushButton и QFileDialog. По нажатию на кнопку “Parce”, на основе этих параметров, создается экземпляр класса Book а затем запускается его метод parce(). Обозначим самые нетривиальные моменты этого метода:

* Параметр firstChapter – первая глава в книге. Если мы ее встретим еще раз, значит, мы уже вышли из оглавления и читаем саму книгу, а значит, можно заканчивать выполнение метода.
* Использование функций isFind(…) (Работает на основе std::string::find(), но независима от регистра благодаря использованию QString::toLower()); getIdx(…) (Достает индекс в виде цифр из названия главы. Если пользователь указал буквенный вид индекса, то он предварительно меняется на цифровой для удобства); isSingleIndex(…) (Определяет, единичный ли перед нами индекс или нет: 1. – единичный, 1.1. – не единичный); getChapterPageAndName(…) (Возвращает пару значений, полученных из строки, а именно – номер страницы и имя глава без страницы); а так же другие функции, написанные в functions.cpp
* Параметр parceNeed – true, когда мы в оглавлении, false, когда мы еще не в нем
* На основе комбинации индекса текущей главы, предыдущей, кол-ва глав в книге, а так же других параметров, определяется, глава перед нами (тогда мы ее добавляем в вектор глав), или же подглава (тогда используем специальный метод, смотри ниже)

Структура классов: класс Book хранит в себе параметры, полученные при создании (вводятся в начале работы пользователем), а также вектор, состоящий из экземпляров класса Chapter; класс Chapter хранит в себе параметры, определяющие главу (имя, страница, индекс), а также, как и Book, вектор, состоящий из экземпляров класса Chapter.

Добавление подглав: на основе ранее описанного метода, программа выбирает, глава это или подглава. Если подглава, то используется метод addSubchapter(…) (рисунок 4), который применяется к последней главе. Внутри него мы рекурсивно проходимся под главам, и когда спускаемся на уровень ниже, откидываем первую часть индекса, и на основе остатка определяем, куда идти дальше (используем метод getChapter(…) – рисунок 5). Так продолжается до тех пор, пока индекс не закончится, т.е., пока мы не окажемся на нужном уровне. Туда и мы добавляем нашу главу.

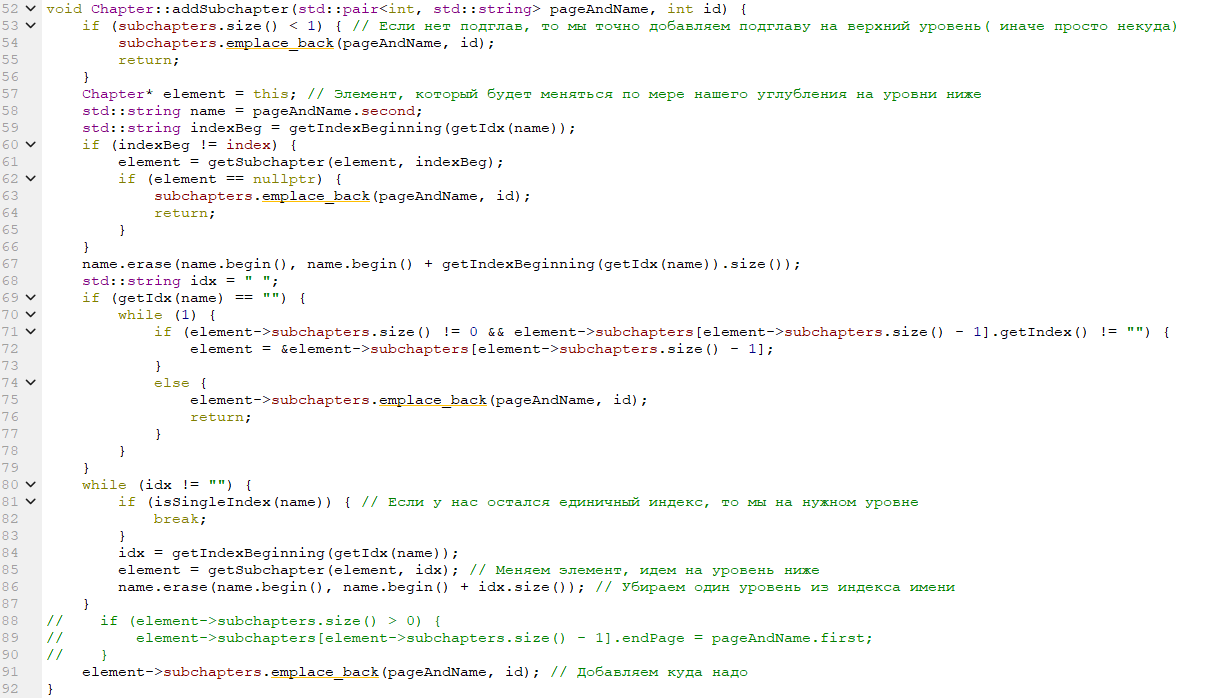


Рисунок 4 - метод addSubchapter(…) класс Chapter

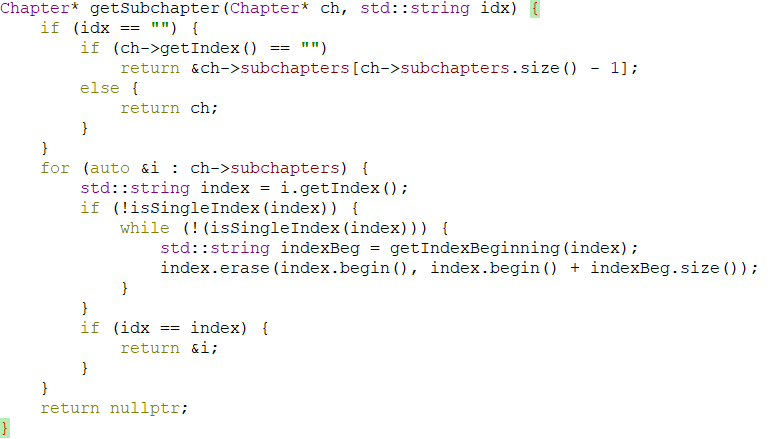


Рисунок 5 – метод getSubchapter(…) класс Chapter

Вывод результатов: первичный(упрощенный) вариант оглавления выводится в поле посередине окна программы. Пользователь может сохранить полное оглавление в csv формате. В таком случае используется метод saveToCsv(…). В нем для начала используется метод fillEndPages(), который, проходя по главам, заполняет их последние страницы (последняя страница – это первая страница следующей главы, в случае последней главы – считаем книгу с конца, и ищем номер страницы, первый который нашли и будет последним, если не нашли, то просто добавим плюс 50 к начальной странице главы), затем проходясь по всем главам, будем последовательно записывать их в csv. Затем отсортируем главы в нужном порядке и получим итоговый вариант. Параметры полученного файла:

id – порядковый номер

index - численный индекс, если его нет, то пустое поле

title – название главы

start\_page – стартовая страница

end\_page – конечная страница

parent\_unit – id главы, в которую помещена текущая глава, пусто, если выше нет главы

level – уровень глубины главы (если нет главы, в которую вложена, то 1, если вложена в одну главу, то 2, если вложена в главу, которая вложена в главу, то 3, и т.д.)

## 2.3 Доступ к результатм.

Отчет и весь исходный код сайта расположен в репозитории:  
https://github.com/dodle4532/BookParce

# ЗАКЛЮЧЕНИЕ

В ходе выполнения проекта была изучена работа с фреймворком Qt, а также получена практика программирования на языке C++ в прикладных целях

# СПИСОК ИСТОЧНИКОВ

1. <https://doc.qt.io>
2. <https://en.cppreference.com>