**3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

В этом разделе описывается функционирование и структура разрабатываемого программного обеспечения

**3.1 Описание функционирования программы**

Работу данного программного обеспечения можно описать следующей цепочкой действий:

* Выбор файла с помощью удобного отображения файлов в древовидной структуре
* Выбор действия (архивация или разархивация)
* Обработка содержимого файла (составление бинарного дерева для архивации и чтение таблицы кодировок для разархивации)
* Запись в новый файл, причем Архиватор читает исходный файл небольшими порциями (к примеру, по 256 байт), каждый байт заменяется кодировкой, полученной путем составления бинарного дерева из всех уникальных байтов исходного. Полученный в ходе этого процесса буфер сжатой информации записывается в выходной файл. Так программа продолжает считывать новые порции исходных данных до тех пор, пока весь исходный файл не будет переведен в свою сжатую форму (рисунок 3.1).

Разархиватор также создает новый файл, но, в отличие от архивации, теперь программа считывает сжатые данные, и, на основе таблицы кодировок байтов, преобразовывает сжатую запись в код байта (рисунок 3.2).

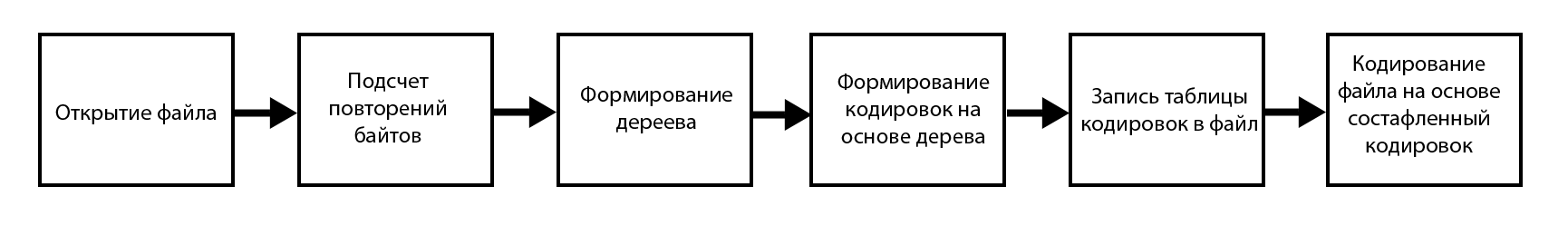


Рисунок 3.1 – Логика архивации файла

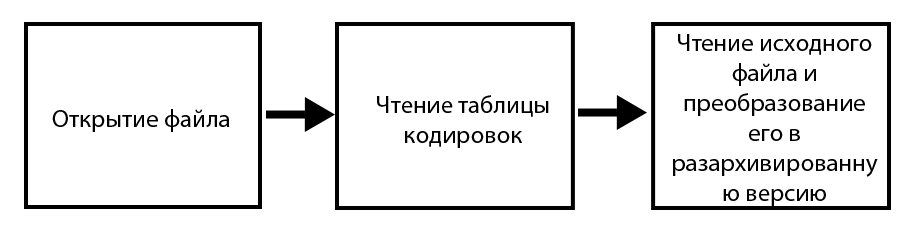


Рисунок 3.2 – Логика разархивации файла

**3.2 Описание структуры взаимодействия внутри приложения**

**3.2.1 Класс Node**

Таблица 3.1 – Шаблонный класс Node

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| has\_value | bool | | Необходимо для различия между листом и узлом дерева |
| Public поля класса | | | |
| Имя | Тип | | Описание |
| code | QString | | Содержит кодировку элемента в листе |
| count | T | | Содержит число, которое в листе значит число повторений байта в файле, а в узле является весом узла |
| value | N | | Значение узла дерева |
| left | Node<T, N>\* | | Указатель на левого наследника узла |
| right | Node<T, N>\* | | Указатель на правого наследника |
| bool | endNode | | Если эта переменная true, то данный узел создан с целью получения уникального кода, который используется как маркер конца сжатой записи в файле |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| Node | T cnt, N val | Node<T, N> | Конструктор новой ноды, устанавливает поле has\_value в true |
| Node | Node<T, N>\* lef, Node<T, N>\* rgt | Node<T, N> | Конструктор нового узла на основе двух существующих. Особенность в том, что has\_value = false, а значение поля count вычисляется как сумма аналогичных полей обоих наследников |
| ~Node | - | - | Вызывает delete для наследников, если они существуют |
| getCount | - | T | Возвращает значение поля count |

Продолжение таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| getValue | - | N | Возвращает значение поля value |
| getCode | - | QString | Возвращает кодировку листа |
| hasValue | - | bool | Возвращает значение поля has\_value |
| setEndNode | bool t | void | Устанавливает текущий узел как узел с кодом маркировки конца архива |
| isEndNode | - | bool | Возвращает значение поля endNode |
| operator> | Node<T, N> other | bool | Сравнивают два узла по значению поля count |
| operator< |
| operator== |
| operator!= |

**3.2.2 Класс NodeComparator**

Таблица 3.2 – Класс NodeComparator

|  |  |  |  |
| --- | --- | --- | --- |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| operator() | Node<int, char> \*first,  Node<int, char> \*second | bool | Компаратор, который используется в приоритетной очереди, для создания приоритета по значению (Впереди будут узлы с наименьшим значением count) |

**3.2.3 Класс bTree**

Таблица 3.3 – Класс bTree

|  |  |  |
| --- | --- | --- |
| Private поля класса | | |
| Имя | Тип | Описание |
| root | Node<int, char> \* | Главный узел дерева |
| dictionary | QMap<char, QString> | Словарь кодировок |
| endCode | QString | Двоичный код маркера конца архива |

Продолжение таблицы 3.3

|  |  |  |  |
| --- | --- | --- | --- |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| bTree | Node<int, char> \*nd | bTree | Конструктор дерева из существующего узла |
| ~bTree | - | - | Вызывает рекурсивный метод destroyTree |
| destroyTree | Node<int, char> \*&node | - | рекурсивно удаляет все узлы дерева |
| formCodes | - | - | Вызывает метод formCodesRec |
| formCodesRec | Node<int, char>\* node,  QString tempCode | - | Проходит по листьям дерева и составляет для них коды ,которые записываются в поле dictionary |
| getDictionary | - | QMap<char, QString>& | Возвращает ссылку на заполненный словарь – поле dictionary |
| getEndCode | - | QString | Возвращает строку двоичного кода маркера конца |

**3.2.4 Класс Catalog**

Таблица 3.4 – Класс Catalog

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| catalog | QMap<char, int> | | Словарь, где хранится байт и число его повторений в файле |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| Catalog | - | - | Конструктор каталога |
| Catalog | QByteArray info | - | Конструктор, принимающий информацию, которую каталог пересчитает |
| add | QByteArray info | QMap<char, int>& | Пересчитывает переданные байты |
| getCatalog | - | QMap<char, int>& | Возвращает словарь catalog |

**3.2.5 Класс Coder**

Таблица 3.5 – Класс Coder

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| endCode | QString | | Строка с бинарным кодом маркера конца архива |
| prev | char | | Незаполненный байт с прошлого вызова метода кодировки буфера |
| leftPrev | bool | | true, если prev не пуст |
| prevSize | short | | Насколько prev заполнен информацией |
| len | int | | Длина полученного из словаря кода |
| dictionary | QMap<char, QString> | | Словарь кодировок |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| Coder | - | - | Конструктор кодировщика |
| hasPrev | - | bool | Возвращает значение поля leftPrev |
| getPrev | - | char | Возвращает значение поля prev |
| setDictionary | QMap<char, QString>& dict | - | Устанавливает значение поля dictionary |
| getDictionary | - | QMap<char, QString>& dict | Возвращает значение поля dictionary |
| setEndCode | QString code | - | Устанавливает значение поля endCode |
| getEndCode | - | QString | Возвращает значение поля endCode |
| getEof | - | QByteArray | Возвращает битовый код, содержащий незаконченный байт prev и endCode |
| encode | char sb | QByteArray | Возвращает закодированный по словарю байт |
| getNextCodeBuffer | QByteArray source | QByteArray | Возвращает сжатую информацию из source |
| static Public методы | | | |
| formByte | QString code | char | Переводит кодировку из строки из 8 нулей и единиц в байты |
| formBytes | QString code | QByteArray | Переводит кодировку из строки любой длины в байты |

**3.2.6 Класс DialogWindow**

Таблица 3.6 – Класс DialogWindow

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| message | QLabel \* | | Информация во всплывающем окне |
| input | QLineEdit \* | | Строка ввода |
| ok | QPushButton \* | | Кнопка окончания ввода |
| cancel | QPushButton \* | | Кнопка прерывания ввода |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| DialogWindow | QWidget \*parent = nullptr | - | Конструктор диалогового окна |
| Pricate slots | | | |
| on\_text\_changed | QString str | - | Слот для сигнала изменения текста в строке ввода |
| on\_ok\_button\_clicked | - | - | Слот для сигнала нажатия кнопки Ok |
| on\_cancel\_button\_clicked | - | - | Слот для сигнала нажатия на кнопку отмены |
| Signals | | | |
| fileNameEntered | QString str | - | Сигнал конца ввода |

**3.2.7 Класс FileCollector**

Таблица 3.7 – Класс FileCollector

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| startDir | QDir | | Папка, с которой начнется сбор файлов |
| allFiles | QList<FileEntry> | | Список информации о файлах в директории |
| allDirs | QStringList | | Список относительных путей папок |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| FileCollector | QDir dir | - | Конструктор сборщика файлов, устанавливает стартовую папку |
| collectFiles | - | QList<DileEntry> | Функция обертка |
| collect\_files | QDir current, QString relativePath | - | Рекурсивно собирает все файлы в папке |
| Static public методы | | | |
| dirSize | QString dirPath, int size, cons tint max | int | вернет число меньше max, если папка достаточно неглубокая, и max+1, если папка слишком глубокая |

**3.2.8 Класс Entry**

Таблица 3.8 – Класс FileEntry

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| path | QString | | Путь к файлу |
| relativePath | QString | | относительный путь к файлу |
| fileName | QString | | Имя файла |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| FileEntry | QString path\_, QString relativePath\_, QString filename\_ | - | Конструктор файловой записи, устанавливает все поля класса |
| setPath | QString path\_ | - | Устанавливает значение поля path |
| getPath | - | QString | Возвращает значение поля path |
| setRelativePath | QString relativePath\_ | - | Устанавливает значение поля relativePath |
| getRelativePath | - | QString | Возвращает значение поля relativePath |
| setFileName | QString fileName\_ | - | Устанавливает значение поля fileName |
| getFileName | - | QString | Возвращает значение поля fileName |

**3.2.9 Класс FileDecoder**

Таблица 3.9 – Класс FileDecoder

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| input | ReadBuffer | | Буфер считывания архива |
| dictionary | QMap<QString, char> | | Словарь кодировок |
| dirName | QString | | Имя папки |
| endCode | QString | | Код маркера конца архива |
| outpPath | QString | | Путь к папке, в которую необходимо записать файл |
| longestCodeSize | int | | Длина самого длинного кода в словаре |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| FileDecoder | - | - | Конструктор разархиватора, инициализирует ReadBuffer input |
| dearchive | QString path, QString fileName | - | Разархивирует архив |
| getPath | - | QString | Возвращает значение поля path |
| decodeDictionary | - | bool | Вернет true, если файл не пуст |
| decodeFile | QFile& outf | - | Разархивирует в файл |
| readDirectoryTree | - | - | Читает метаданные из архива и создает необходимые папки |
| getNum() | - | int | Читает число из файла |
| getPath() | - | QString | Читает строку из файла |
| toCode | char c, int length | QString | Преобразовывает байт длиной не более восбми в строку нулей и единиц |

**3.2.10 Класс FileTranslator**

Таблица 3.10 – Класс FileTranslator

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| four | QFile | | Файл архива |
| fin | QFile | | Текущий входной файл |
| allFiles | QList<FileEntry> | | Список всех файлов на архивацию |
| coder | Coder | | Кодировщик для сжатия информации |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| FileTranslator | QString path | - | Конструктор переводчика файла |
| setAllFiles | QList<FileEntry> allFiles\_ | - | Устанавливает значение поля allFiles |
| getAllFiles | - | QList<FileEntry> | Возвращает значение поля allFiles |
| translateFiles | - | - | Последовательно архивирует каждый элемент списка allFiles |
| translateDictionary | - | - | Записывает словарь текущего файла в архив |

**3.2.11 Класс MainWindow**

Таблица 3.11 – Класс MainWindow

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| ui | Ui::MainWindow\* | | Поле графической модели программы |
| model | QFileSystem\* | | Модель файловой системы дерева каталогов |
| activeFile | QFileInfo\* | | Текущий выбранный файл |
| dw | DialogWindow\* | | Всплывающее окно ввода имени архива |
| wmb | QaitMessageBox\* | | Всплывающее окно ожидания конца архивации |
| fileName | QString | | Имя архива |
| MAX\_DEPTH | const int | | Максимально разрешенное количество файлов в папке |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| MainWindow | QWidget\* parent = nullptr | - | Конструктор главного окна |
| Private slots | | | |
| on\_treeView\_clicked | const QModelIndex& index | - | Слот для сигнала о выборе файла в дереве файлов |
| on\_archiveButton\_clicked | - | - | Слот для сигнала о нажатии на кнопку архивации |
| on\_dearchiveButton\_clicked | - | - | Слот для сигнала о нажатии на кнопку разархивации |
| on\_fileName\_entered | QString input | - | Слот для сигнала об окончании ввода имени архива |

**3.2.12 Класс ReadBuffer**

Таблица 3.12 – Класс ReadBuffer

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| fin | QFile | | Входной файл для считывания буфера |
| buffer | char\* | | Массив для чтения |
| inputSize | const size\_t | | Размер массива |
| index | size\_t | | Индекс в массиве |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| ReadBuffer | int inputSize\_ | - | Конструктор буфера чтения |
| ~ReadBuffer | - | - | Деструктор буфера чтения |
| openFile | QString fileName | - | Открывает переданный файл |
| get | - | char | Читает следующий байт файла |
| peek | - | - | Демонстрирует следующий байт файла |
| nextBuffer | - | - | Читает следующий буфер из файла |
| isEnd | - | bool | Вернет true, если указатель файла на конце |

**3.2.13 Класс TreeFormer**

Таблица 3.13 – Класс TreeFormer

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| nodes | priority\_queue<Node<int, char>\*, vector<Node<int, char>\*>, NodeComparator> | | Очередь с приоритетом, с помощью которой построится дерево |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| TreeFormer | - | - | Конструктор формировщика дерева |
| TreeFormer | QMap<char, int> cat | - | Конструктор формировщика дерева c добавлением каталога в узлы |
| add | Node<int, char>\* nw | - | Добавляет узел в очередь |
| take | - | Node<int, char>\* nw | Берет узел из вершины очереди |
| formBTree | - | Node<int, char>\* nw | Формирует бинарное дерево и возвращает указатель на вершину |
| nextBuffer | - | - | Читает следующий буфер из файла |

**3.2.14 Класс WaitMessageBox**

Таблица 3.14 – Класс WaitMessageBox

|  |  |  |  |
| --- | --- | --- | --- |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| WaitMessageBox | QWidget\* parent = nullptr | - | Конструктор окна ожидания |
| TreeFormer | QMap<char, int> cat | - | Конструктор формировщика дерева c добавлением каталога в узлы |
| add | Node<int, char>\* nw | - | Добавляет узел в очередь |
| take | - | Node<int, char>\* nw | Берет узел из вершины очереди |
| formBTree | - | Node<int, char>\* nw | Формирует бинарное дерево и возвращает указатель на вершину |
| nextBuffer | - | - | Читает следующий буфер из файла |