# 3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

В этом разделе описывается функционирование разрабатываемого программного обеспечения, классы и их методы.

Диаграмма классов предоставлена в Приложении Б.

## 3.1 Описание функционирования программы

Работу данного программного обеспечения можно описать следующей цепочкой действий:

1) Выбор файла с помощью удобного отображения файлов в древовидной структуре, что обеспечивается блоком пользовательского интерфейса.

2) Выбор действия (архивация, разархивация или удаление)

3) Если для архивации была выбрана директория, а не файл, то первым делом происходит поиск всех вложенных в директорию файлов. Это происходит в блоке обработки входных данных для архивации.

4) Обработка содержимого файла (составление бинарного дерева для архивации и чтение таблицы кодировок для разархивации), данные действия обрабатываются в блоке обработки содержимого файлов и блоке обработки содержимого архива соответственно.

5) 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), это происходит в блоке архивации на основе обработанных данных. Разархиватор начинает с чтения дерева каталогов из архива и создания папок, далее все файлы архива по-одному разархивируются в соответствующие папки на основе кодировок байтов, которые записаны в архиве перед каждой сжатой записью (рисунок 3.2), это обеспечивается блоком разархивации файла на основе обработанных данных.

7) Если Пользователь прервал операцию архивации или разархивации, то программа перейдет к соответствующему слоту обработки этого сигнала, так при прерывании операции архивации файл архива будет удален.

8) Далее программа переходит к блоку пользовательского интерфейса.

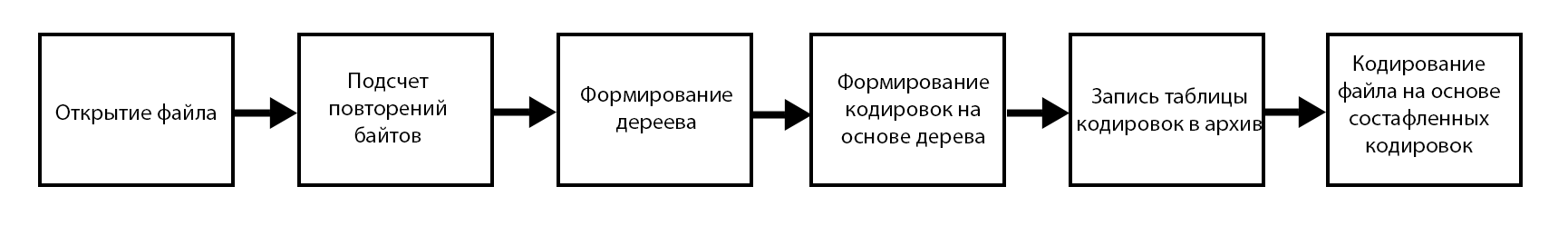


Рисунок 3.1.1 – Логика архивации файла

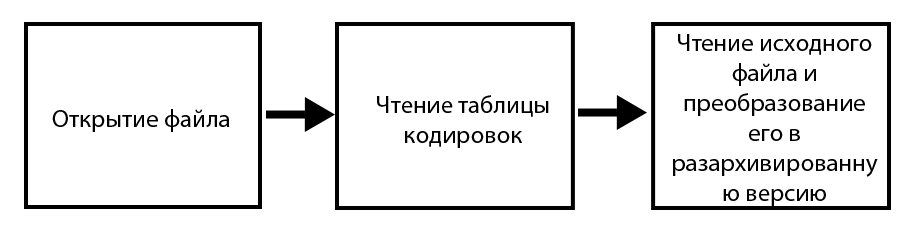


Рисунок 3.1.2 – Логика разархивации файла

## 3.2 Описание структуры взаимодействия внутри приложения

### 3.2.1 Шаблонный класс Node

Данный класс является узлом структуры данных двоичного дерева поиска (класс bTree), является шаблоном двух типов: T и N, где T – тип, значение которого будет использовано для сортировки дерева, а N – тип поля, которое будет иметь значение только в листах дерева (лист не имеет наследников и в нем определено значение поля value).

Таблица 3.2.1 – 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 |
| 1 | 2 | 3 |
| code | QString | Содержит кодировку элемента в листе |
| count | T | Содержит число, которое в листе значит число повторений байта в файле, а в узле является суммой значений этого поля в наследниках |

Продолжение таблицы 3.2.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | | 3 |
|  |  | |  |
| value | N | | Значение узла дерева |
| left | Node<T, N>\* | | Указатель на левого наследника узла |
| right | Node<T, N>\* | | Указатель на правого наследника |
| endNode | bool | | 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 |
| 1 | 2 | 3 | 4 |
| Node | T cnt | - | Конструктор узла с одним параметром – значением поля count |
| Node | T cnt, N val | - | Конструктор нового узла, устанавливает значение поля has\_value true |

Продолжение таблицы 3.2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | | 3 | 4 |
| Node | Node<T, N>\* lef, Node<T, N>\* rgt | | - | 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 |
| getCount | - | | T | Возвращает значение поля count |
| getValue | | - | N | Возвращает значение поля value |
| getCode | | - | QString | Возвращает битовую 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< | |

Продолжение таблицы 3.2.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| operator== | Node<T, N> other | bool | Сравнивают два узла по значению поля count |
| operator!= |

### 3.2.2 Класс NodeComparator

Класс-компаратор, предназначен для упорядочивания узлов по возрастанию значения поля count в различных контейнерах, в частности в очереди с приоритетом.

Таблица 3.2.2 – 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 возрастанию значения |

### 3.2.3 Класс bTree

Структура данных Бинарное дерево поиска, необходимо в алгоритме Хаффмана для составления битовых кодировок узлов.

Таблица 3.2.3 – Класс bTree

|  |  |  |
| --- | --- | --- |
| Private поля класса | | |
| Имя | Тип | Описание |
| 1 | 2 | 3 |
| root | Node<int, char> \* | Главный узел дерева |

Продолжение таблицы 3.2.3

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | | 3 |
| dictionary | QMap<char, QString> | | Словарь кодировок |
| endCode | QString | | Двоичный код маркера конца архива |
| 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

Данный класс предназначен для подсчета числа повторений каждого уникально байта в входном файле.

Таблица 3.2.4 – 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 | QByteArray info | - | Конструктор, принимающий информацию, которой класс дополнит свое поле catalog |
| add | QByteArray info | QMap<char, int>& | Пересчитывает переданные байты |
| getCatalog | - | QMap<char, int>& | Возвращает словарь catalog |

### 3.2.5 Класс Coder

Класс Coder предназначен для перевода входной информации в сжатую форму при помощи словаря кодировок, полученного с помощью бинарного дерева поиска bTree.

Таблица 3.2.5 – 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 | | Число бит в prev |
| len | int | | Длина полученного из словаря кода |
| dictionary | QMap<char, QString> | | Словарь кодировок |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| 1 | 2 | 3 | 4 |
| Coder | - | - | Конструктор кодировщика, инициализирует поля leftPrev и prevSize |
| hasPrev | - | bool | Возвращает значение поля leftPrev |
| getPrev | - | char | Возвращает значение поля prev |
| setDictionary | QMap<char, QString>& dict | - | Устанавливает значение поля dictionary |

Продолжение таблицы 3.2.5

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 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 |
| clear | - | - | Очищает все поля класса Coder, позволяя использовать тот же объект заново |
| 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

Класс предназначен для диалога с пользователем. Окно запрашивает имя файла. Кнопка Ok становится доступна только при вводе хотя бы одного символа. Это реализуется с помощью системы слотов и сигналов. При завершении вода, объект окна посылает сигнал о завершении операции, который поступает в соответствующий ему слот класса MainWindow.

Таблица 3.2.6 – Класс DialogWindow

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | | Описание |
| message | QLabel \* | | Информация в 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 |
| Private слоты | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| 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 |

Продолжение таблицы 3.2.6

|  |  |  |  |
| --- | --- | --- | --- |
| Сигналы | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| fileNameEntered | QString str | - | Сигнал конца ввода |

### 3.2.7 Класс FileCollector

Класс для сбора всех файлов в директории, а также их абсолютных и относительных путей.

Таблица 3.2.7 – 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<FileEntry> | Функция обертка |
| collect\_files | QDir current, QString relativePath | - | Рекурсивно собирает все файлы в папке |

Продолжение таблицы 3.2.7

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |
| isCorrectFileName | QString path | bool | Проверяет переданный путь на содержание запрещенных символов, а также на некорректные имена |

### 3.2.8 Класс FileEntry

Класс FileEntry содержит полный путь к файлу, относительный путь к файлу и имя файла, используется в классе FileCollector.

Таблица 3.2.8 – Класс FileEntry

|  |  |  |
| --- | --- | --- |
| Private поля класса | | |
| Имя | Тип | Описание |
| path | QString | Путь к файлу |
| relativePath | QString | относительный путь к файлу |
| fileName | QString | Имя файла |

Продолжение таблицы 3.2.8

|  |  |  |  |
| --- | --- | --- | --- |
| 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

Класс для разархивации сжатой записи. Сначала он считывает и создает нужные папки, затем по-одному считывает и распаковывает файлы, создавая их в нужных папках.

Таблица 3.2.9 – 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. Нужна для определения повреждений в архиве |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| 1 | 2 | 3 | 4 |
| FileDecoder | - | - | Конструктор разархиватора, инициализирует ReadBuffer input |
| dearchive | QString path, QString fileName | - | Распаковывает архив |
| decodeDictionary | - | bool | Вернет true, если файл не пуст |
| decodeFile | QFile& outf | - | Распаковывает сжатую запись в файл |

Продолжение таблицы 3.2.9

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| readDirectoryTree | - | - | Читает метаданные из архива и создает необходимые папки |
| getNum() | - | int | Читает число из файла |
| getPath() | - | QByteArray | Читает строку из файла |
| toCode | char c, int length | QString | Преобразовывает байт длиной не более восьми в строку нулей и единиц |

### 3.2.10 Класс FileTranslator

Предназначен для сжатия входного файла или файлов в один архив, сохраняя при этом информацию о папках и директориях, если они есть.

Таблица 3.2.10 – Класс FileTranslator

|  |  |  |
| --- | --- | --- |
| Private поля класса | | |
| Имя | Тип | Описание |
| fout | QFile | Файл архива |
| fin | QFile | Текущий входной файл |
| allFiles | QList<FileEntry> | Список всех файлов на архивацию |
| coder | Coder | Кодировщик для сжатия информации |

Продолжение таблицы 3.2.10

|  |  |  |  |
| --- | --- | --- | --- |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| FileTranslator | QString path | - | Конструктор переводчика файла |
| ~FileTranslator | - | - | Закрывает входной и выходной файлы |
| openFile | QString path\_ | - | Открывает указанный в аргументе метода файл |
| 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 |
| clear | - | - | Очищает поля объекта и закрывает открытые файлы |

### 3.2.11 Класс MainWindow

Класс основного окна приложения. В нем осуществляются все взаимодействие с пользователем, обработка сигналов и создание потоков архивации и разархивации.

Таблица 3.2.11 – Класс MainWindow

|  |  |  |  |
| --- | --- | --- | --- |
| Пространства имен | | | |
| Имя | Описание | | |
| Ui | Пространство имени для класса MainWindow | | |
| Private поля класса | | | |
| Имя | | Тип | Описание |
| 1 | | 2 | 3 |
| ui | | Ui::MainWindow\* | Поле графической модели программы |
| model | | QFileSystemModel\* | Модель файловой системы дерева каталогов |
| activeFile | | QFileInfo\* | Текущий выбранный файл |
| dw | | DialogWindow\* | Всплывающее окно ввода имени архива |
| archiveWait | | WaitBox\* | Всплывающее окно ожидания конца архивации |
| dearchiveWait | | WaitBox\* | Всплывающее окно ожидания конца разархивации |
| at | | ArchivationThread\* | Поток архивации |
| dt | | DearchivationThread\* | Поток разархивации |
| fileName | | QString | Имя архива |

Продолжение таблицы 3.2.11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | | 3 | |
| MAX\_DEPTH | const int | | Максимально разрешенное количество файлов в папке для удаления | |
| processing | bool | | Флаг, указывающий на то, производятся в данный момент операции создания или распаковки архива или нет | |
| Public методы класса | | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | | Описание |
| MainWindow | QWidget\* parent = nullptr | - | | Конструктор главного окна, инициализи-рует поля класса |
| ~MainWindow | - | - | | Деструктор |
| Private слоты | | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | | Описание |
| 1 | 2 | 3 | | 4 |
| 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 |

Продолжение таблицы 3.2.11

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| on\_dearchiveButton\_clicked | - | - | Слот для сигнала о нажатии на кнопку разархивации |
| on\_fileName\_entered | QString input | - | Слот для сигнала об окончании ввода имени архива |
| when\_archivation\_complete | - | - | Слот для сигнала о конце операции архивации |
| when\_archivation\_canceled | - | - | Слот для сигнала о прерывании операции архивации |
| when\_dearchivation\_complete | - | - | Слот для сигнала о конце операции разархивации |
| when\_dearchivation\_canceled | - | - | Слот для сигнала о прерывании операции разархивации |
| on\_delete\_button\_clicked | - | - | Слот для сигнала нажатия на кнопку удаления |
| when\_thread\_exception\_handled | - | - | Слот для сигнала об исключении в потоке |

Продолжение таблицы 3.2.11

|  |  |  |  |
| --- | --- | --- | --- |
| Сигналы | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| close\_waitBox | - | - | Сигнал закрытия окна ожидания конца операции |

### 3.2.12 Класс ReadBuffer

Класс для чтения архива. Его удобство в том, что он считывает буфер информации, размером, указанном в конструкторе класса, и потом дает возможность получать доступ к отдельным байтам буфера.

Таблица 3.2.12 – 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 |
| readen | size\_t | Число прочитанных в последней операции чтения из файла байтов |

Продолжение таблицы 3.2.12

|  |  |  |  |
| --- | --- | --- | --- |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| ReadBuffer | int inputSize\_ | - | Конструктор буфера чтения, инициализирует массив buffer |
| ~ReadBuffer | - | - | Деструктор буфера чтения |
| openFile | QString fileName | - | Открывает файл |
| get | - | char | Возвращает 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

Файл предназначен для формирования бинарного дерева поиска из словаря информации о всех уникальных байтах исходного файла и числа их повторений.

Таблица 3.2.13 – 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 строится дерево |

Продолжение таблицы 3.2.13

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

### 3.2.14 Класс WaitBox

Класс представляет окно ожидания окончания операции, в котором пользователь может нажать на кнопку Cancel и прервать текущую операцию сжатия или распаковки.

Таблица 3.2.14 – Класс WaitBox

|  |  |  |
| --- | --- | --- |
| Private поля класса | | |
| Имя | Тип | Описание |
| msgLabel | QLabel\* | Строка с информацией для пользователя |
| cancel | QPushButton\* | Кнопка для отмены текущей операции |

Продолжение таблицы 3.2.14

|  |  |  |  |
| --- | --- | --- | --- |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| WaitBox | QWidget\* parent = nullptr | - | Конструктор окна ожидания |
| show | - | - | Метод отображения окна на экране |
| Сигналы | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| operation\_canceled | - | - | Сигнал, уведомляющий об отмене операции |
| Public слоты | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| on\_cancel\_button\_clicked | - | - | Слот обработки нажатия кнопки отмены |
| on\_signal\_to\_close | - | - | Слот для сигнала о закрытии окна |

### 3.2.15 Класс ArchivationThread

Данный класс необходим для выделения процесса архивации в отдельный поток. В нем происходит сбор информации о папках и передача этой информации классу FileTranslator. Также поток перехватывает исключения и передает информацию о них основному потоку с помощью сигнала.

Таблица 3.2.15 – Класс ArchivationThread

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Private поля класса | | | | | |
| Имя | | Тип | | Описание | |
| fileName | | QString | | Имя архива | |
| activeFile | | QFileInfo\* | | Указатель на выбранный файл | |
| fcList | | QList<FileEntry> | | Список Файлов в папке | |
| ft | | FileTranslator | | Кодировщик файлов | |
| Public методы класса | | | | | |
| Имя | Принимаемые параметры | | Возвращаемое значение | | Описание |
| ArchivationThread | QString filename\_, QFileInfo\* activeFile\_ | | - | | Конструктор потока архивации |
| setFileName | QString fileName\_ | | - | | Устанавливает значение поля filrName |
| setActiveFile | QFileInfo\* | | - | | Устанавливает значение поля activeFile |
| clear | - | | - | | Очищает все поля объекта |
| getFileName | - | | QString | | Возвращает значение поля fileName |
| getFcList | - | | QList<FileEntry> | | Возвращает значение поля fcList |
| getArchiveName | - | | QString | | Возвращает имя архива |
| run | - | | - | | Метод для исполнения потока |

Продолжение таблицы 3.2.17

|  |  |  |  |
| --- | --- | --- | --- |
| Сигналы | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| exception\_executed | QString e | - | Сигнал для главного потока с информацией об исключении |

## 3.2.16 Класс DearchivationThread

Класс необходим для запуска процесса распаковки архива в отдельном потоке. В этом классе происходит перехват исключений из класса FileDecoder и перенаправление их в главный поток с помощью сигнала.

Таблица 3.2.16 – Класс DearchivationThread

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Private поля класса | | | | | |
| Имя | | Тип | | Описание | |
| fileName | | QString | | Имя архива | |
| filePath | | QString | | Путь к архиву | |
| Public методы класса | | | | | |
| Имя | Принимаемые параметры | | Возвращаемое значение | | Описание |
| DearchivationThread | QString filename\_, QString filePath\_ | | - | | Конструктор потока разархивации |
| setFileName | QString fileName\_ | | - | | Устанавливает значение поля filrName |
| setFilePath | QString filePath\_ | | - | | Устанавливает значение поля filePath |
| run | - | | - | | Метод для исполнения потока |

Продолжение таблицы 3.2.16

|  |  |  |  |
| --- | --- | --- | --- |
| Сигналы | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| exception\_executed | QString e | - | Сигнал для главного потока с информацией об исключении |