Szperacz dyskowy

Generated by Doxygen 1.8.11

Contents

Chapter 1

Szperacz dyskowy

Author

Damian Świerk

Date

8.11.2017

1.1 Założenia

Założenia projektowe Cel projektu:

1. przeglądanie zawartości systemu plików,
2. wizualizacja hierarchii plików w postaci drzewa,
3. wyszukiwanie i "katalogowanie" plików konkretnego rodzaju (np. robienie spisu filmów, książek czy muzyki),
4. obliczanie zajętego przez katalog i pliki miejsca,
5. podgląd wybranych rodzajów plików (np. tekstowych)
6. program ma mieć menu z zaimplementowanymi opcjami np. do odczytywania/zapisywania plików, okienko "o programie" oraz podstawową pomoc (np. opisy przeznaczenia widgetów w postaci "dymków"),
7. program ma zapisywać swój "stan" (np. rozmiar i pozycja okna programu, aktualnie analizowany katalog) i po kolejnym włączeniu wracać do niego (np. minimum zapisywanie pozycji i rozmiaru okna programu).

Wnioski:

1. Zadanie pozwoliło na poznanie działania i metod obsługi widgetów, modeli i obiektów odpowiadających za obsługę danych.
2. Zdobyto wiedzę na temat sposobu dostępu do danych oraz możliwości ich odczytu.
3. Nie zaimplementowano możliwości edycji plików tekstowych oraz rekurencyjnego wyszukiwania plików

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 Ui Namespace Reference

6.1.1 Detailed Description

Date

8.11.2017r.

Author

Damian Świerk

Chapter 7

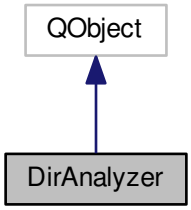
Class Documentation

7.1 DirAnalyzer Class Reference

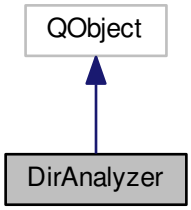
Klasa DirAnalyzer Klasa służy do przetwarzania informacji o danych zamieszczonych w podanej lokalizacji. Pozwala na pozyskanie listy plików znajdujących się w katalogu. Posiada funkcjonalność otwierania plików tekstowych.

`#include <diranalyzer.h>`

Inheritance diagram for DirAnalyzer:



Collaboration diagram for DirAnalyzer:



Public Member Functions

- DirAnalyzer (QObject ∗parent=0)

  Konstruktor.
- void passNewDir (const QModelIndex &index)

  metoda przygotowywująca nowe informacje o danych w podanej lokalizacji
- bool isDir ()

  metoda sprawdzająca czy podana lokalizacja jest folderem
- bool isTextFile ()

  metoda sprawdzająca czy podana lokalizacja jest plikiem tekstowym
- QString getTextFileContent ()

  metoda zwracająca zawartość pliku tekstowego
- QString getDirName ()

  metoda zwracająca nazwę katalogu
- QStringList getFileList ()

  metoda zwracająca zawartość katalogu
- QString getFileName ()

  metoda zwracająca nazwe pliku
- QString getPath ()

  metoda zwracająca podaną ścieżkę
- void setFilter (QString suffix)

  metoda ustawiająca filtr plików

7.1.1 Detailed Description

Klasa DirAnalyzer Klasa służy do przetwarzania informacji o danych zamieszczonych w podanej lokalizacji. Pozwala na pozyskanie listy plików znajdujących się w katalogu. Posiada funkcjonalność otwierania plików tekstowych.

Date

8.11.2017r.

Author

Damian Świerk

7.1.2 Constructor & Destructor Documentation

7.1.2.1 DirAnalyzer::DirAnalyzer (QObject ∗ *parent* = 0) [explicit]

Konstruktor.

DirAnalyzer::DirAnalyzer Konstruktor klasy.

Parameters

parent	- wskaźnik na widget nadrzędny

7.1.3 Member Function Documentation

7.1.3.1 QString DirAnalyzer::getDirName ( )

metoda zwracająca nazwę katalogu

DirAnalyzer::getDirName Metoda zwracająca nazwę katalogu.

7.1.3.2 QStringList DirAnalyzer::getFileList ( )

metoda zwracająca zawartość katalogu

DirAnalyzer::getFileList Metoda zwracająca listę plików w katalogu wraz z ich rozmiarami.

7.1.3.3 QString DirAnalyzer::getFileName ( )

metoda zwracająca nazwe pliku

DirAnalyzer::getFileName Metoda zwracająca nazwę wybranego pliku.

7.1.3.4 QString DirAnalyzer::getPath ( )

metoda zwracająca podaną ścieżkę

DirAnalyzer::getPath Metoda zwracająca bezwzględny adres do pliku.

7.1.3.5 QString DirAnalyzer::getTextFileContent ( )

metoda zwracająca zawartość pliku tekstowego

DirAnalyzer::getTextFileContent Metoda zwracająca zawartość pliku tekstowego.

7.1.3.6 bool DirAnalyzer::isDir ( )

metoda sprawdzająca czy podana lokalizacja jest folderem

DirAnalyzer::isDir Metoda sprawdzająca czy przechowywana ścieżka jest adresem folderu.

7.1.3.7 bool DirAnalyzer::isTextFile ( )

metoda sprawdzająca czy podana lokalizacja jest plikiem tekstowym

DirAnalyzer::isTextFile Metoda sprawdzająca czy przechowywana ścieżka jest adresem pliku tekstowego.

7.1.3.8 void DirAnalyzer::passNewDir (const QModelIndex & *index*)

metoda przygotowywująca nowe informacje o danych w podanej lokalizacji

DirAnalyzer::passNewDir Metoda pobierająca index modelu przedstawiającego wybraną przez użytkownika lokalizację, a następnie przetwarza informacje o danych zawierających się pod tą lokalizacją. Algorytm sprawdza czy podana ścieżka przechowuje plik, jeśli tak to sprawdza czy jest on również plikiem tekstowym. Jeżeli ścieżka przechowuje plik tekstowy to jego zawartość zostaje pobrana i zapisana w atrybucie m_txtContent. Jeżeli ścieżka wskazuje na folder, to zostaje wczytana zawartość folderu wraz z rozmarami plików i uwzględnieniem filtru rozszerzenia.

Parameters

index	- indeks modelu przechowującego adres lokalizacji

7.1.3.9 void DirAnalyzer::setFilter (QString *suffix*)

metoda ustawiająca filtr plików

DirAnalyzer::setFilter Metoda ustawiająca nowy filtr rozszerzenia plików.

Parameters

index	- indeks modelu przechowującego adres lokalizacji

The documentation for this class was generated from the following files:

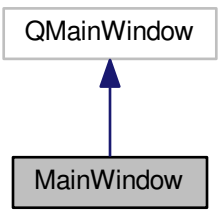
- diranalyzer.h
- diranalyzer.cpp

7.2 MainWindow Class Reference

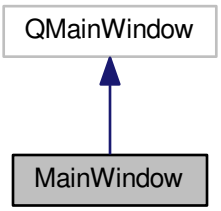
Klasa MainWindow Klasa Main Window - interfejs graficzny szperacza dyskowego. Jest to pomost między operacjami na katalogach i plikach, które wykonywane są w obiekcie m_analyzer, a graficzną wizualizacją danych w głownym oknie programu.

```
#include <mainwindow.h>
```

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



Public Member Functions

- MainWindow (QWidget ∗parent=0)

  Konstruktor.
- ∼MainWindow ()

  Destruktor.

7.2.1 Detailed Description

Klasa MainWindow Klasa Main Window - interfejs graficzny szperacza dyskowego. Jest to pomost między operacjami na katalogach i plikach, które wykonywane są w obiekcie m_analyzer, a graficzną wizualizacją danych w głownym oknie programu.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 MainWindow::MainWindow (QWidget ∗ *parent* = 0) [explicit]

Konstruktor.

MainWindow::MainWindow Konstruktor klasy. W Konstruktorze tym jest zawarta obsługa slotów i sygnałów programu, a także wprowadzone są ustawienia programu, wraz z deklaracją koniecznych obiektów.

Parameters

parent	- wskaźnik na widget nadrzędny

7.2.2.2 MainWindow::∼MainWindow ( )

Destruktor.

MainWindow::∼MainWindow Destruktor klasy.

The documentation for this class was generated from the following files:

- mainwindow.h
- mainwindow.cpp

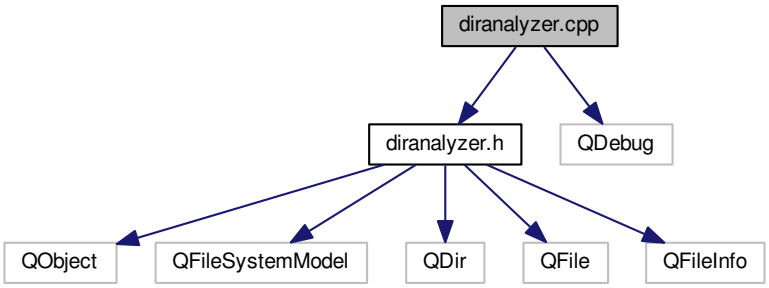
Chapter 8

File Documentation

8.1 diranalyzer.cpp File Reference

```
#include "diranalyzer.h"
#include <QDebug>
```

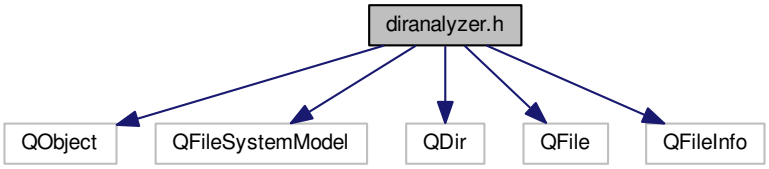
Include dependency graph for diranalyzer.cpp:



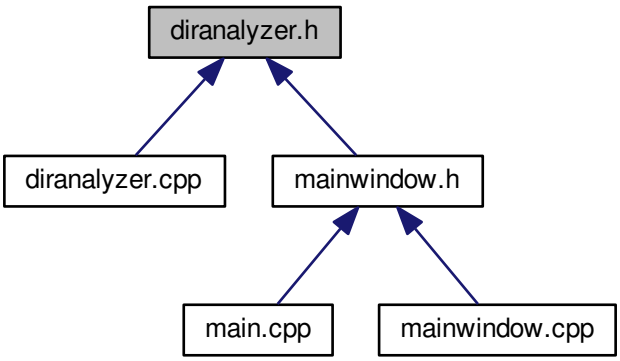
8.2 diranalyzer.h File Reference

```
#include <QObject>
#include <QFileSystemModel>
#include <QDir>
#include <QFile>
#include <QFileInfo>
```

Include dependency graph for diranalyzer.h:



This graph shows which files directly or indirectly include this file:



Classes

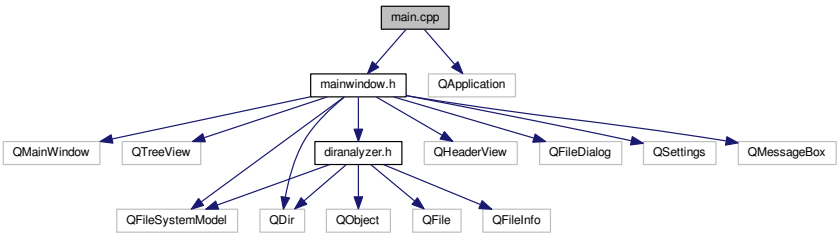
- class DirAnalyzer

  Klasa DirAnalyzer Klasa służy do przetwarzania informacji o danych zamieszczonych w podanej lokalizacji. Pozwala na pozyskanie listy plików znajdujących się w katalogu. Posiada funkcjonalność otwierania plików tekstowych.

8.3 main.cpp File Reference

```
#include "mainwindow.h"
#include <QApplication>
```

Include dependency graph for main.cpp:



Functions

- int main (int argc, char ∗argv[ ])

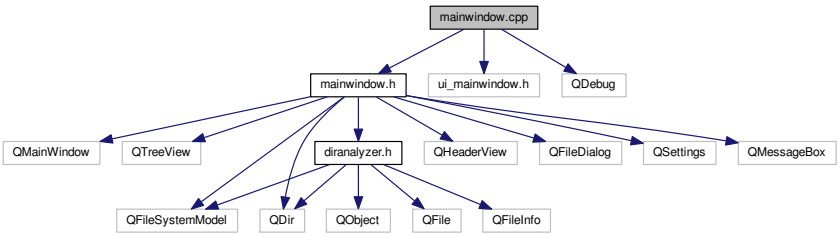
8.3.1 Function Documentation

8.3.1.1 int main (int *argc,* char ∗ *argv[ ]*)

8.4 mainwindow.cpp File Reference

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QDebug>
```

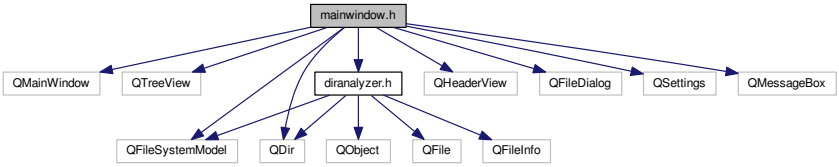
Include dependency graph for mainwindow.cpp:



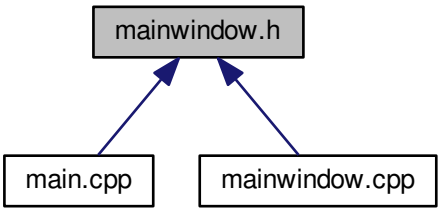
8.5 mainwindow.h File Reference

```
#include <QMainWindow>
#include <QTreeView>
#include <QFileSystemModel>
#include <QDir>
#include <QHeaderView>
#include <QFileDialog>
#include <QSettings>
#include <QMessageBox>
#include "diranalyzer.h"
```

Include dependency graph for mainwindow.h:



This graph shows which files directly or indirectly include this file:



Classes

- class MainWindow

  Klasa MainWindow Klasa Main Window - interfejs graficzny szperacza dyskowego. Jest to pomost między operacjami na katalogach i plikach, które wykonywane są w obiekcie m_analyzer, a graficzną wizualizacją danych w głownym oknie programu.

Namespaces

- Ui