

DTConverter

Wygenerowano przez Doxygen 1.8.9.1

Pt, 6 lut 2015 11:11:31

Spis treści

1	Indeks hierarchiczny	1
1.1	Hierarchia klas	1
2	Indeks klas	3
2.1	Lista klas	3
3	Dokumentacja klas	5
3.1	Dokumentacja klasy Compressor	5
3.1.1	Dokumentacja konstruktora i destruktora	5
3.1.1.1	Compressor	5
3.1.2	Dokumentacja funkcji składowych	5
3.1.2.1	getDictionaryIndex	5
3.1.2.2	getMaxIndex	6
3.1.2.3	getPixel	6
3.1.2.4	getPixels	6
3.2	Dokumentacja klasy Converter	6
3.2.1	Dokumentacja funkcji składowych	7
3.2.1.1	run	7
3.3	Dokumentacja klasy Decoder	7
3.3.1	Dokumentacja funkcji składowych	8
3.3.1.1	bin2dec	8
3.3.1.2	binaryPixelToRGB	8
3.3.1.3	charToString	8
3.3.1.4	drawPicture	8
3.3.1.5	power	8
3.3.1.6	saveBMP	9
3.3.1.7	setPixel	9
3.4	Dokumentacja klasy FileWriter	9
3.4.1	Dokumentacja konstruktora i destruktora	9
3.4.1.1	FileWriter	9
3.4.2	Dokumentacja funkcji składowych	10
3.4.2.1	convertValueToBinary	10

3.4.2.2	saveFile	10
3.5	Dokumentacja klasy MainWindow	10
3.6	Dokumentacja klasy Picture	11
3.6.1	Dokumentacja konstruktora i destruktora	12
3.6.1.1	Picture	12
3.6.2	Dokumentacja funkcji składowych	13
3.6.2.1	checkBMPFile	13
3.6.2.2	getPictureColors	13
3.6.2.3	getPictureHeight	13
3.6.2.4	getPictureWidth	13
3.6.2.5	getPixelColor	13
3.7	Dokumentacja klasy Reader	14
3.7.1	Dokumentacja funkcji składowych	14
3.7.1.1	open	14
3.7.1.2	readDictionary	14
3.7.1.3	readIndexesFromPixels	14
Indeks		15

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Compressor	5
Decoder	7
Reader	14
FileWriter	9
Picture	11
QMainWindow	
MainWindow	10
QObject	
Converter	6

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Compressor	5
Converter	6
Decoder	7
FileWriter	9
MainWindow	10
Picture	11
Reader	14

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy Compressor

Metody publiczne

- **Compressor** (const vector< SDL_Color > &startColors, **Picture** *picture)
- vector< int > **getPixels** ()
- int **getMaxIndex** ()

Metody prywatne

- SDL_Color **getPixel** (const int &x, const int &y)
- int **getDictionaryIndex** (const vector< SDL_Color > &c)

Atrybuty prywatne

- map< int, vector< SDL_Color > > **dictionary**
- **Picture** * **picture**
- int **dictionarySize**
- int **maxIndex**

3.1.1 Dokumentacja konstruktora i destruktor

3.1.1.1 Compressor::Compressor (const vector< SDL_Color > & startColors, Picture * picture)

Konstruktor klasy tworzący początkowe hasła słownika

Parametry

<i>startColors</i>	lista kolorów obrazka
<i>picture</i>	obiekt klasy Picture (str. 11) będący reprezentacją obrazka wejściowego (DI)

3.1.2 Dokumentacja funkcji składowych

3.1.2.1 int Compressor::getDictionaryIndex (const vector< SDL_Color > & c) [private]

Parametry

<i>c</i>	lista kolorów, którą należy odszukać w słowniku
----------	---

Zwraca

pozycja w słowniku lub -1 jeżeli słowa nie ma w słowniku

3.1.2.2 `int Compressor::getMaxIndex ()`

Zwraca

największy użyty indeks ze słownika LZW

3.1.2.3 `SDL_Color Compressor::getPixel (const int & x, const int & y)` `[private]`

Parametry

<i>x</i>	położenie piksela na osi x
<i>y</i>	położenie piksela na osi y

Zwraca

obiekt klasy `SDL_Color` będący reprezentacją piksela o współrzędnych (x,y) w odpowiednim trybie koloru

3.1.2.4 `vector< int > Compressor::getPixels ()`

Właściwa kompresja LZW

Zwraca

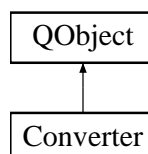
lista kolejno zapisanych pixeli wg. indeksów słownika LZW

Dokumentacja dla tej klasy została wygenerowana z plików:

- `Compressor.h`
- `Compressor.cpp`

3.2 Dokumentacja klasy Converter

Diagram dziedziczenia dla Converter



Sloty publiczne

- `void run ()`

Sygnaly

- void **conversionStart** ()
- void **conversionEnd** ()
- void **conversionProgress** (int value)
- void **conversionSuccess** ()
- void **conversionFailed** ()
- void **finished** ()
- void **error** (QString err)

Metody publiczne

- **Converter** (QObject *parent=0)
- **Converter** (const std::string &openPath, const std::string &savePath, const std::string &saveName, const int &colorType)

Atrybuty prywatne

- std::string **openPath**
- std::string **savePath**
- std::string **saveName**
- int **colorType**

3.2.1 Dokumentacja funkcji składowych

3.2.1.1 void Converter::run () [slot]

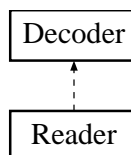
Funkcja przeprowadzająca cały proces konwersji pliku BMP do pliku DT.

Dokumentacja dla tej klasy została wygenerowana z plików:

- Converter.h
- Converter.cpp

3.3 Dokumentacja klasy Decoder

Diagram dziedziczenia dla Decoder



Metody publiczne

- int **power** (int liczba, int dopotegi)
- int **bin2dec** (string input)
- string **charToString** (char *buffer, int lenght)
- void **binaryPixelToRGB** (string binaryPixel)
- void **drawPicture** (string name_save)
- void **saveBMP** (string name_save)
- void **setPixel** (int x, int y, Uint8 R, Uint8 G, Uint8 B)

Atrybuty publiczne

- SDL_Surface * **screen**
- int **height**
- int **width**
- char const * **tytul** = "BitMapPreview"

Atrybuty chronione

- int **maxColors**
- int **pixelWidth**
- int **dictionaryStart**
- int **pictureStart**
- vector< SDL_Color > **pixels**
- map< int, string > **dictionaryColors**

3.3.1 Dokumentacja funkcji składowych

3.3.1.1 int Decoder::bin2dec (string *input*)

Funkcja zamiany liczby zapisanej w systemie binarnym na system dziesiętny

Parametry

<i>input</i>	liczba zapisana w systemie binarnym
--------------	-------------------------------------

3.3.1.2 void Decoder::binaryPixelToRGB (string *binaryPixel*)

Funkcja zmieniająca odczyt binarny piksela na obiekt SDL_Color

Parametry

<i>binaryPixel</i>	binarny zapis piksela
--------------------	-----------------------

3.3.1.3 string Decoder::charToString (char * *buffer*, int *length*)

Funkcja rzutująca typ char na string

Parametry

<i>buffer</i>	bufor danych
<i>length</i>	długość bufora

3.3.1.4 void Decoder::drawPicture (string *name_save*)

Funkcja tworząca obraz

Parametry

<i>name_save</i>	nazwa pliku wynikowego
------------------	------------------------

3.3.1.5 int Decoder::power (int *liczba*, int *dopotegi*)

Funkcja potęgowa używana do zamiany liczby z systemu binarnego na dziesiętny

3.3.1.6 void Decoder::saveBMP (string *name_save*)

Funkcja odpowiedzialna za zapis do pliku

Parametry

<i>name_save</i>	nazwa pliku do którego zapisujemy obraz
------------------	---

3.3.1.7 void Decoder::setPixel (int *x*, int *y*, Uint8 *R*, Uint8 *G*, Uint8 *B*)

Funkcja ustawiająca pixel

Parametry

<i>x</i>	współrzędna x obrazka
<i>y</i>	współrzędna y obrazka
<i>R</i>	składowa R koloru ustawianego piksela
<i>G</i>	składowa G koloru ustawianego piksela
<i>B</i>	składowa B koloru ustawianego piksela

Dokumentacja dla tej klasy została wygenerowana z plików:

- decoder.h
- Decoder.cpp

3.4 Dokumentacja klasy FileWriter

Metody publiczne

- **FileWriter** (const string &savePath, const string &saveName)
- void **saveFile** (**Picture** *picture, const vector< int > &pixelListAfterCompression, const vector< SDL_Color > &colorsList, const int &maxIndex)

Metody prywatne

- string **convertValueToBinary** (int value, const int &precision)

Atrybuty prywatne

- string **savePath**
- string **saveName**

3.4.1 Dokumentacja konstruktora i destruktor

3.4.1.1 FileWriter::FileWriter (const string & *savePath*, const string & *saveName*)

Konstruktor klasy **FileWriter** (str. 9)

Parametry

<i>savePath</i>	miejsce, gdzie ma zostać zapisany plik wynikowy
-----------------	---

<i>saveName</i>	nazwa pliku wynikowego
-----------------	------------------------

3.4.2 Dokumentacja funkcji składowych

3.4.2.1 `string FileWriter::convertValueToBinary (int value, const int & precision)` [private]

Konwersja liczby w systemie dziesiętnym na system binarny

Parametry

<i>value</i>	wartość, która zostanie zamieniona na system binarny
<i>precision</i>	długość ciągu wynikowego

Zwraca

bitowy zapis *value* o długości *precision*

3.4.2.2 `void FileWriter::saveFile (Picture * picture, const vector< int > & pixelListAfterCompression, const vector< SDL_Color > & colorsList, const int & maxIndex)`

Funkcja odpowiedzialna za prawidłowy zapis do pliku

Parametry

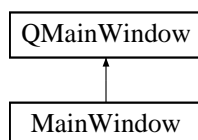
<i>picture</i>	obiekt klasy Picture (str. 11) będący reprezentacją obrazka wejściowego (DI)
<i>pixelListAfterCompression</i>	lista indeksów słownika LZW po kompresji (lista pikseli)
<i>colorsList</i>	lista kolorów obrazu wejściowego
<i>maxIndex</i>	maksymalny indeks słownika LZW użyty przy kompresji LZW

Dokumentacja dla tej klasy została wygenerowana z plików:

- FileWriter.h
- FileWriter.cpp

3.5 Dokumentacja klasy MainWindow

Diagram dziedziczenia dla MainWindow



Metody publiczne

- **MainWindow** (QWidget *parent=0)

Sloty prywatne

- void **conversionStartHandle** ()
- void **conversionEndHandle** ()

- void **conversionProgressHandle** (int value)
- void **conversionSuccessHandle** ()
- void **conversionFailedHandle** ()
- void **errorStringHandle** (QString error)
- void **on_bdFile_selectionChanged** ()
- void **on_bdPath_selectionChanged** ()
- void **on_bdFileButton_released** ()
- void **on_bdPathButton_released** ()
- void **on_bdConvertButton_released** ()
- void **on_dbFile_selectionChanged** ()
- void **on_dbPath_selectionChanged** ()
- void **on_dbFileButton_released** ()
- void **on_dbPathButton_released** ()
- void **on_dbConvertButton_released** ()

Metody prywatne

- QString **getFilePath** (int mode)
- QString **getFolderPath** (int mode)

Atrybuty prywatne

- Ui::MainWindow * **ui**
- QProgressDialog * **progress** = NULL
- int **qLineMaxLength** = 500

Dokumentacja dla tej klasy została wygenerowana z plików:

- mainwindow.h
- mainwindow.cpp

3.6 Dokumentacja klasy Picture

Metody publiczne

- **Picture** (const string &openPath, const int &colorType)
- vector< SDL_Color > **getPictureColors** ()
- SDL_Color **getPixelColor** (const int &x, const int &y)
- int **getPictureWidth** ()
- int **getPictureHeight** ()
- bool **checkBMPFile** ()

Metody prywatne

- bool **isInList** (const vector< SDL_Color > &ListOfColors, const SDL_Color &color)

Atrybuty prywatne

- SDL_Surface * **BMP** = NULL
- int **colorType**

3.6.1 Dokumentacja konstruktora i destruktora

3.6.1.1 `Picture::Picture (const string & openPath, const int & colorType)`

Konstruktor klasy **Picture** (str. 11)

Parametry

<i>openPath</i>	ścieżka do pliku wejściowego BMP
<i>colorType</i>	konwersja w kolorze lub skali szarości

3.6.2 Dokumentacja funkcji składowych

3.6.2.1 bool Picture::checkBMPFile ()

Funkcja sprawdzająca poprawność otwarcia pliku BMP

Zwraca

informacja o tym, czy plik wejściowy BMP został poprawnie otworzony

3.6.2.2 vector< SDL_Color > Picture::getPictureColors ()

Funkcja generująca listę wszystkich kolorów obrazu

Zwraca

lista wszystkich kolorów obrazu

3.6.2.3 int Picture::getPictureHeight ()

Funkcja zwracająca wysokość obrazka wejściowego

Zwraca

wysokość wejściowego obrazu BMP

3.6.2.4 int Picture::getPictureWidth ()

Funkcja zwracająca szerokość obrazka wejściowego

Zwraca

szerokość wejściowego obrazu BMP

3.6.2.5 SDL_Color Picture::getPixelColor (const int & x, const int & y)

Funkcja odpowiedzialna za zwrócenie koloru piksela

Parametry

<i>x</i>	współrzędna x piksela
<i>y</i>	współrzędna y piksela

Zwraca

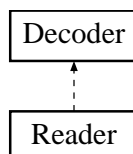
referencja do obiektu klasy SDL_Color będącego reprezentacją piksela obrazu o współrzędnych (x,y)

Dokumentacja dla tej klasy została wygenerowana z plików:

- Picture.h
- Picture.cpp

3.7 Dokumentacja klasy Reader

Diagram dziedziczenia dla Reader



Metody publiczne

- void **readIndexesFromPixels** (string name_save)
- void **readDictionary** (string name_save)
- void **open** (string name, string name_save)

Atrybuty prywatne

- fstream **plik**

Dodatkowe Dziedziczone Składowe

3.7.1 Dokumentacja funkcji składowych

3.7.1.1 void Reader::open (string name, string name_save)

Funkcja otwierająca plik z rozszerzeniem dt

Parametry

<i>name</i>	nazwa otwieranego pliku
<i>name_save</i>	nazwa pliku wynikowego

end of pictureStart

3.7.1.2 void Reader::readDictionary (string name_save)

Funkcja czytająca słownik z pliku

Parametry

<i>name_save</i>	nazwa pliku DT
------------------	----------------

3.7.1.3 void Reader::readIndexesFromPixels (string name_save)

Funkcja dekodująca plik DT alorytmem LZW

Parametry

<i>name_save</i>	nazwa pliku DT
------------------	----------------

Dokumentacja dla tej klasy została wygenerowana z plików:

- reader.h
- Reader.cpp

Skorowidz

- bin2dec
 - Decoder, 8
- binaryPixelToRGB
 - Decoder, 8
- charToString
 - Decoder, 8
- checkBMPFile
 - Picture, 13
- Compressor, 5
 - Compressor, 5
 - getDictionaryIndex, 5
 - getMaxIndex, 6
 - getPixel, 6
 - getPixels, 6
- convertValueToBinary
 - FileWriter, 10
- Converter, 6
 - run, 7
- Decoder, 7
 - bin2dec, 8
 - binaryPixelToRGB, 8
 - charToString, 8
 - drawPicture, 8
 - power, 8
 - saveBMP, 8
 - setPixel, 9
- drawPicture
 - Decoder, 8
- FileWriter, 9
 - convertValueToBinary, 10
 - FileWriter, 9
 - saveFile, 10
- getDictionaryIndex
 - Compressor, 5
- getMaxIndex
 - Compressor, 6
- getPictureColors
 - Picture, 13
- getPictureHeight
 - Picture, 13
- getPictureWidth
 - Picture, 13
- getPixel
 - Compressor, 6
- getPixelColor
 - Picture, 13
- getPixels
 - Compressor, 6
- MainWindow, 10
- open
 - Reader, 14
- Picture, 11
 - checkBMPFile, 13
 - getPictureColors, 13
 - getPictureHeight, 13
 - getPictureWidth, 13
 - getPixelColor, 13
 - Picture, 12
- power
 - Decoder, 8
- readDictionary
 - Reader, 14
- readIndexesFromPixels
 - Reader, 14
- Reader, 14
 - open, 14
 - readDictionary, 14
 - readIndexesFromPixels, 14
- run
 - Converter, 7
- saveBMP
 - Decoder, 8
- saveFile
 - FileWriter, 10
- setPixel
 - Decoder, 9