AntiPlag Software Design Document

Zhang Huimeng
2015011280

Liu Wentong
????????

June 25, 2016

Contents

Part I

Introduction

Chapter 1

Purpose

Chapter 2

Scope

Chapter 3

Overview

Chapter 4

Reference Material

Part II

System Overview

Part III

System Architecture

Chapter 6

Architectural Design

Chapter 7

Decomposition Description

Chapter 8

Design Rationale

Part IV

Data Design

Chapter 9

Data Description

Chapter 10

Data Dictionary

Part V

Component Design

Chapter 11

Namespace Index

11.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 12

Hierarchical Index

12.1 Class Hierarchy

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

Chapter 13

Class Index

13.1 Class List

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

Chapter 14

Namespace Documentation

14.1 Ui Namespace Reference

Classes

- class Widget

Chapter 15

Class Documentation

15.1 Document Class Reference

Public Member Functions

- Document (std::string address)
- void RabinKarp ()

  Perform Rabin-Karp algorithm for this document.
- void KMP ()

  Perdorm KMP algorithm for this document.
- std::string getAddress ()

Protected Member Functions

- void makePattern ()

  Make the patterns with winnowing algorithm.
- void preprocess ()
- bool isValid (char c)

Private Member Functions

- Document & operator= (const Document &other)

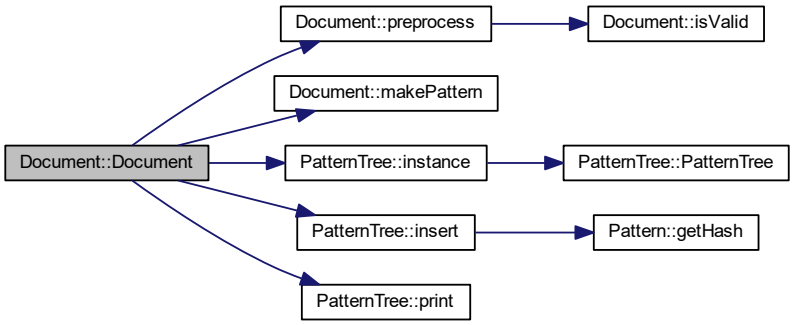
Private Attributes

- std::string m_address
- std::string m_content
- std::vector< Pattern > m_patterns

15.1.1 Constructor & Destructor Documentation

Document::Document (std::string *address*)

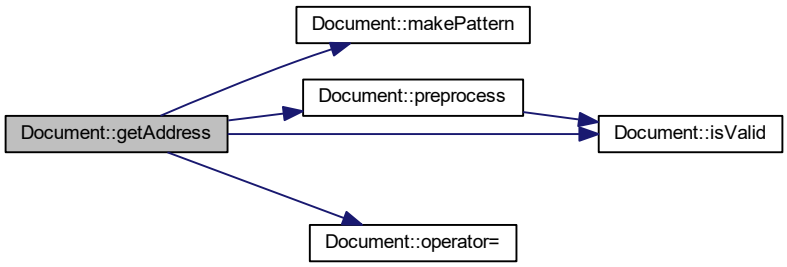
Here is the call graph for this function:



15.1.2 Member Function Documentation

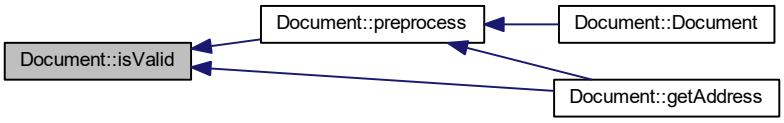
std::string Document::getAddress ( ) `[inline]`

Here is the call graph for this function:



bool Document::isValid (char *c*) [protected]

Here is the caller graph for this function:



void Document::KMP ( )

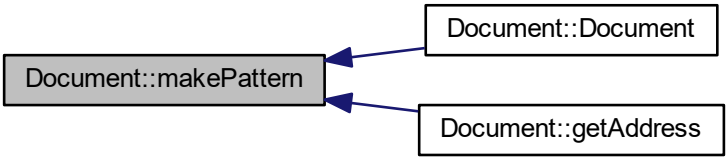
Perdorm KMP algorithm for this document.

Here is the call graph for this function:

void Document::makePattern ( ) [protected]

Make the patterns with winnowing algorithm.

Here is the caller graph for this function:



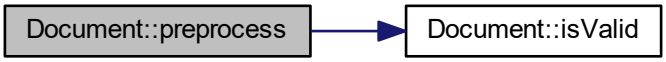
Document& Document::operator= (const Document & *other*) [private]

Here is the caller graph for this function:

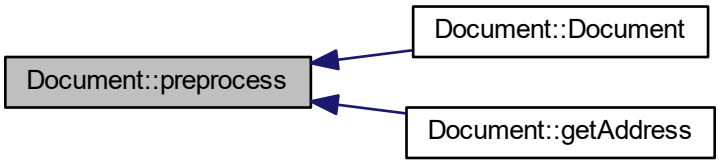
void Document::preprocess ( ) [protected]

Remove spaces and taps and etc; Need to replace comments Need to add replacement

Here is the call graph for this function:



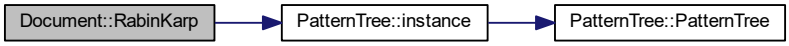
Here is the caller graph for this function:



void Document::RabinKarp ( )

Perform Rabin-Karp algorithm for this document.

Here is the call graph for this function:



15.1.3 Member Data Documentation

std::string Document::m_address `[private]`

std::string Document::m_content `[private]`

std::vector<Pattern> Document::m_patterns `[private]`

15.2 Homework Class Reference

Public Types

- enum HomeworkType { Single, Multiple }

Public Member Functions

- Homework (std::string path, HomeworkType type)

  Initialization and build the whole file system.

Protected Member Functions

- bool findSingle (std::string filePath)
- bool findMultiple (std::string filePath)
- void dfsFolder (std::string folderPath, std::vector< std::string > &address)

Private Attributes

- HomeworkType m_type
- std::string m_path
- std::vector< Project > m_projects

15.2.1 Member Enumeration Documentation

enum Homework::HomeworkType

Enumerator

Single

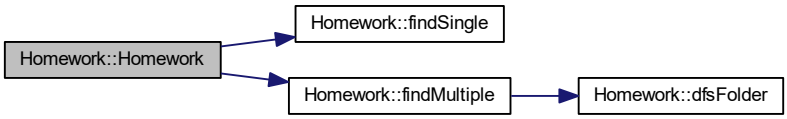
Multiple

15.2.2 Constructor & Destructor Documentation

Homework::Homework (std::string *path,* HomeworkType *type*)

Initialization and build the whole file system.

Here is the call graph for this function:



15.2.3 Member Function Documentation

void Homework::dfsFolder (std::string *folderPath,* std::vector< std::string > & *address*) [protected]

Here is the caller graph for this function:

bool Homework::findMultiple (std::string *filePath*) [protected]

Here is the call graph for this function:

Here is the caller graph for this function:

bool Homework::findSingle (std::string *filePath*) [protected]

Here is the caller graph for this function:

15.2.4 Member Data Documentation

std::string Homework::m_path [private]

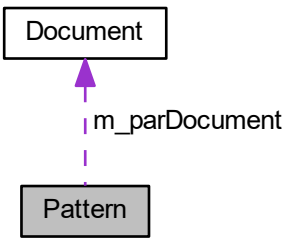
std::vector<Project> Homework::m_projects [private]

HomeworkType Homework::m_type [private]

15.3 Pattern Class Reference

Storing patterns from a document.

Collaboration diagram for Pattern:



Public Member Functions

- Pattern (Document &parDocument, std::string pattern, int pos)

  The only constructor.
- bool operator< (const Pattern &right)

  operator < for inserting into PatternTree
- long long int getHash () const
- int getLength () const
- Document * getParDocument ()
- std::string getPattern () const
- void print () const

  print basic information about the pattern

Protected Member Functions

- void calcHash ()

Private Member Functions

- Pattern & operator= (const Pattern &other)

Private Attributes

- std::string m_pattern
- int m_pos
- long long int m_hash
- Document * m_parDocument

15.3.1 Detailed Description

Storing patterns from a document.

15.3.2 Constructor & Destructor Documentation

Pattern::Pattern (Document & *parDocument*, std::string *pattern*, int *pos*)

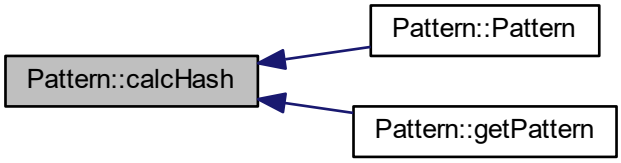
The only constructor.

Here is the call graph for this function:

15.3.3 Member Function Documentation

void Pattern::calcHash ( ) [protected]

Here is the caller graph for this function:



long long int Pattern::getHash ( ) const `[inline]`

Returns

The hash value.

Here is the caller graph for this function:

int Pattern::getLength ( ) const `[inline]`

Returns

the length of the pattern.

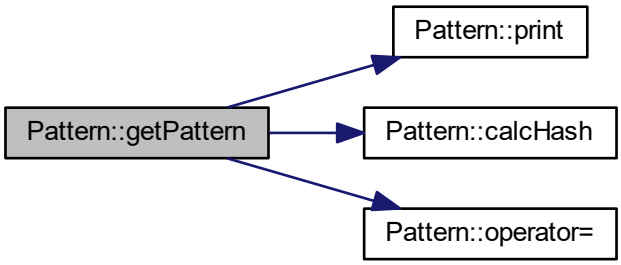
Document* Pattern::getParDocument ( ) `[inline]`

Returns

the address of the parent document. Note: need to use const here

std::string Pattern::getPattern ( ) const `[inline]`

Here is the call graph for this function:



bool Pattern::operator< (const Pattern & *right*) `[inline]`

operator < for inserting into PatternTree

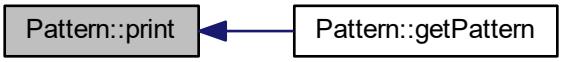
Pattern& Pattern::operator= (const Pattern & *other*) [private]

Here is the caller graph for this function:

void Pattern::print ( ) const

print basic information about the pattern

Here is the caller graph for this function:



15.3.4 Member Data Documentation

long long int Pattern::m_hash [private]

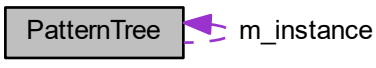
Document* Pattern::m_parDocument [private]

std::string Pattern::m_pattern [private]

int Pattern::m_pos [private]

15.4 PatternTree Class Reference

Collaboration diagram for PatternTree:



Public Member Functions

- PatternTree ()
- void destroy ()
- void insert (const Pattern &pattern)

  insert a pattern into the tree
- std::vector< Pattern > find (const long long int hash)
- std::vector< Pattern > getAll ()
- void print ()

  print some basic information about the tree

Static Public Member Functions

- static PatternTree * instance ()

Private Member Functions

- PatternTree (const PatternTree &other)
- PatternTree & operator= (const PatternTree &right)

Private Attributes

- patternMmap m_tree

Static Private Attributes

- static PatternTree * m_instance = NULL

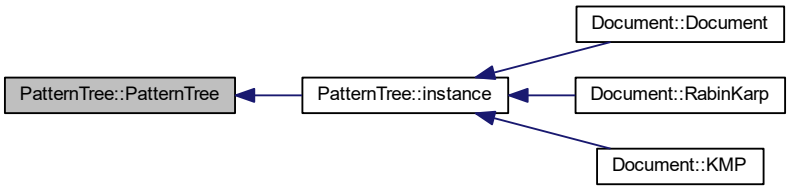
15.4.1 Detailed Description

A tree with all patterns stored in it Singleton

15.4.2 Constructor & Destructor Documentation

PatternTree::PatternTree ( )

Here is the caller graph for this function:



PatternTree::PatternTree (const PatternTree & *other*) [private]

15.4.3 Member Function Documentation

void PatternTree::destroy ( )

std::vector< Pattern > PatternTree::find (const long long int *hash*)

find a set of patterns with the same hash value in the tree Note: I cannot make it const...

std::vector< Pattern > PatternTree::getAll ( )

void PatternTree::insert (const Pattern & *pattern*)

insert a pattern into the tree

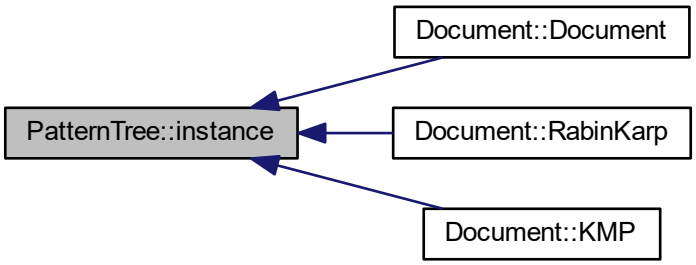
Here is the call graph for this function:

Here is the caller graph for this function:

PatternTree * PatternTree::instance ( ) [static]

Here is the call graph for this function:

Here is the caller graph for this function:

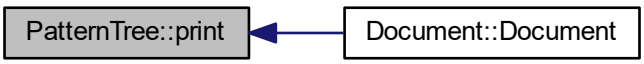


PatternTree& PatternTree::operator= (const PatternTree & *right*) [private]

void PatternTree::print ( )

print some basic information about the tree

Here is the caller graph for this function:



15.4.4 Member Data Documentation

PatternTree * PatternTree::m_instance = NULL [static], [private]

patternMmap PatternTree::m_tree [private]

15.5 Project Class Reference

Public Member Functions

- Project (std::string path, const std::vector< std::string > &address)

  Note: need to think about reference here.

Private Attributes

- std::string m_path
- std::vector< Document > m_documents

15.5.1 Constructor & Destructor Documentation

Project::Project (std::string *path*, const std::vector< std::string > & *address*)

Note: need to think about reference here.

15.5.2 Member Data Documentation

std::vector<Document> Project::m_documents `[private]`

std::string Project::m_path `[private]`

15.6 qt_meta_stringdata_Widget_t Struct Reference

Public Attributes

- QByteArrayData data [3]
- char stringdata0 [14]

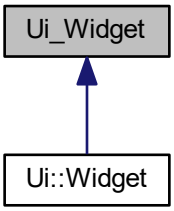
15.6.1 Member Data Documentation

QByteArrayData qt_meta_stringdata_Widget_t::data[3]

char qt_meta_stringdata_Widget_t::stringdata0[14]

15.7 Ui_Widget Class Reference

Inheritance diagram for Ui_Widget:



Public Member Functions

- void setupUi (QWidget *Widget)
- void retranslateUi (QWidget *Widget)

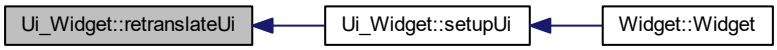
Public Attributes

- QLabel * label
- QLabel * label_2
- QLabel * label_3
- QPushButton * pushButton
- QLabel * label_4

15.7.1 Member Function Documentation

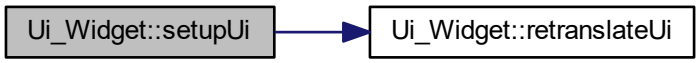
void Ui_Widget::retranslateUi (QWidget * *Widget*) [inline]

Here is the caller graph for this function:

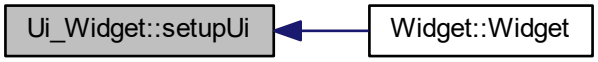


void Ui_Widget::setupUi (QWidget * *Widget*) [inline]

Here is the call graph for this function:



Here is the caller graph for this function:



15.7.2 Member Data Documentation

QLabel* Ui_Widget::label

QLabel* Ui_Widget::label_2

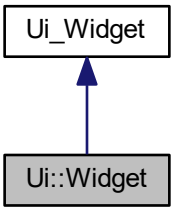
QLabel* Ui_Widget::label_3

QLabel* Ui_Widget::label_4

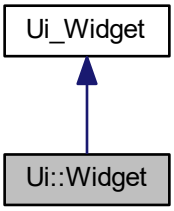
QPushButton* Ui_Widget::pushButton

15.8 Ui::Widget Class Reference

Inheritance diagram for Ui::Widget:



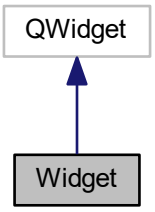
Collaboration diagram for Ui::Widget:



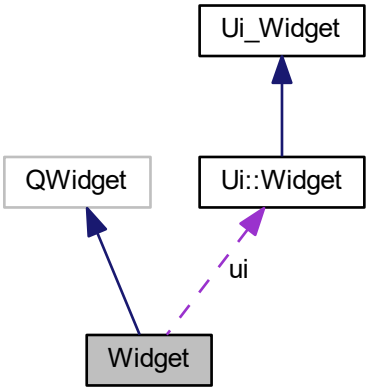
Additional Inherited Members

15.9 Widget Class Reference

Inheritance diagram for Widget:



Collaboration diagram for Widget:



Public Slots

- void Print ()

Public Member Functions

- Widget (QWidget *parent=0)
- ~Widget ()

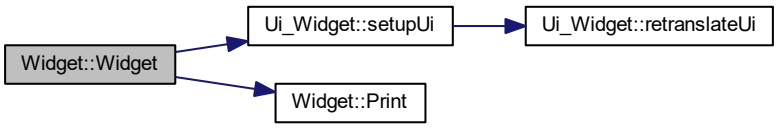
Private Attributes

- Ui::Widget * ui

15.9.1 Constructor & Destructor Documentation

Widget::Widget (QWidget * *parent = 0*) [explicit]

Here is the call graph for this function:



Widget::~Widget ( )

15.9.2 Member Function Documentation

void Widget::Print ( ) [slot]

Here is the caller graph for this function:

15.9.3 Member Data Documentation

Ui::Widget* Widget::ui [private]

Part VI

Human Interface Design

Chapter 16

Overview of Human Interface

Chapter 17

Screen Images

Chapter 18

Screen Objects and Actions

Part VII

Design Patterns

Bibliography

[1] This is an example.