PA04

Tim Kwist
Version 1.0
9/24/14

Table of Contents

Table of contents

Hierarchical Index

Class Hierarchy

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

Class Index

Class List

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

File Index

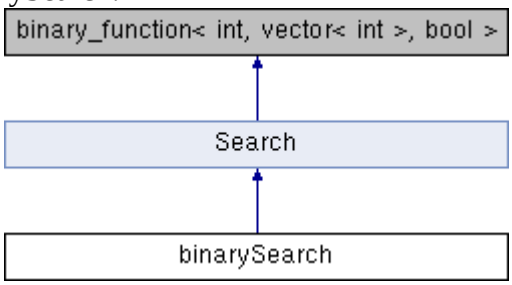
File List

Here is a list of all documented files with brief descriptions:

Class Documentation

binarySearch Class Reference

Inheritance diagram for binarySearch:



Public Member Functions

- bool **operator()** (int searchValue, const vector< int > &keys) const

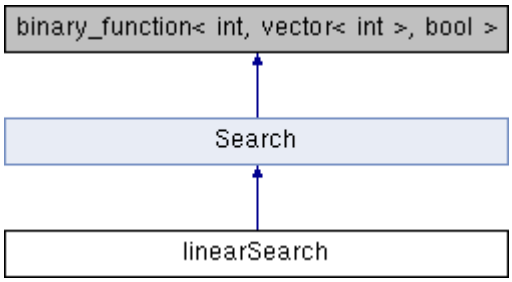
---

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

- search.cpp

linearSearch Class Reference

Inheritance diagram for linearSearch:



Public Member Functions

- bool **operator()** (int searchValue, const vector< int > &keys) const

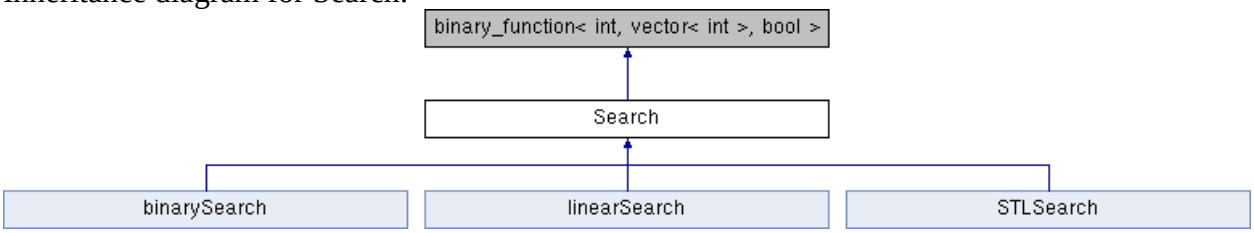
---

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

- search.cpp

Search Class Reference

Inheritance diagram for Search:

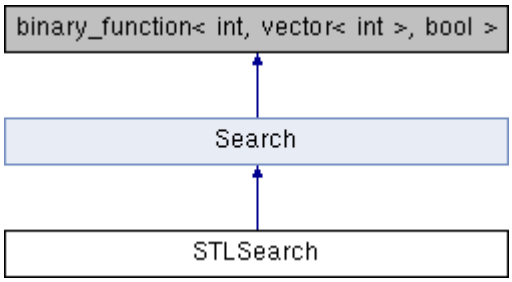


The documentation for this class was generated from the following file:

- search.cpp

STLSearch Class Reference

Inheritance diagram for STLSearch:



Public Member Functions

- bool **operator()** (int searchValue, const vector< int > &keys) const

---

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

- search.cpp

TestVector Class Reference

Public Member Functions

- **TestVector** (int size)
- **TestVector** (const **TestVector** &rhs)
- **TestVector** & **operator++** ()
- **TestVector operator++** (int ignored)
- int **operator[]** (int loc) const

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

- TestVector.h
- TestVector.cpp

Timer Class Reference

Public Member Functions

- **Timer** ()
- void **start** () throw (runtime_error)
- void **stop** () throw (logic_error)
- double **getElapsedTime** () const throw (logic_error)

Constructor & Destructor Documentation

Timer::Timer ()

Method implementation Initialize **Timer** object

Parameters:

none	

Returns:

none

Precondition:

None

Postcondition:

Initialize the internal timer valuesso that the timer is ready to measure time

---

Member Function Documentation

double Timer::getElapsedTime () const throw logic_error)

Find how much time has elapsed between start and end of timer

Parameters:

none	

Returns:

double Length of time interval in seconds

Precondition:

Timer has been started and ended

Postcondition:

Returns the length of the time interval in seconds

void Timer::start () throw runtime_error)

Set timer started to true and record the beginning time

Parameters:

none	

Returns:

none

Precondition:

Timer has not started yet

Postcondition:

Mark the beginning of a time interval; start the timer

void Timer::stop () throw logic_error)

Set timer started to false and record end time of timer

Parameters:

none	

Returns:

none

Precondition:

Timer has started

Postcondition:

Marks the end of a time interval; stop the timer

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

- Timer.h
- **Timer.cpp**
- Timer.cs

File Documentation

Timer.cpp File Reference

This program implements a stopwatch. It starts, stops and stores the time elapsed in between.

```
#include <sys/time.h>
#include <stdexcept>
#include <iostream>
#include "Timer.h"
```

Detailed Description

This program implements a stopwatch. It starts, stops and stores the time elapsed in between.

Author:

Tim Kwist

Version:

1.0

Date:

Wednesday, September 23nd, 2014

The program uses system time to determine elapsed time. Starting stores the current system time. Stopping gets the current system time and subtracts the beginning time to get the elapsed time.

Index

INDEX