Lab1

1.0

Generated by Doxygen 1.8.9.1

Wed Mar 11 2015 15:21:46

# **Contents**

1	Clas	s Index				1
	1.1	Class	List		 	 1
2	File	Index				3
	2.1	File Lis	st		 	 3
3	Clas	s Docu	mentation	n		5
	3.1	Bench	mark Clas	ss Reference	 	 5
		3.1.1	Detailed	Description	 	 5
		3.1.2	Member	Function Documentation	 	 5
			3.1.2.1	test	 	 5
	3.2	InputF	iles Class	Reference	 	 6
		3.2.1	Detailed	Description	 	 6
		3.2.2	Construc	ctor & Destructor Documentation	 	 6
			3.2.2.1	InputFiles	 	 6
			3.2.2.2	InputFiles	 	 7
		3.2.3	Member	Function Documentation	 	 8
			3.2.3.1	generate_random_int_data	 	 8
			3.2.3.2	return_file_name	 	 9
			3.2.3.3	return_file_size	 	 10
			3.2.3.4	return_number_files	 	 10
			3.2.3.5	show_info	 	 10
4	File	Docum	entation			11
	4.1	src/bei	nchmark f	frm.cpp File Reference	 	 11
	4.2			frm.h File Reference		
	4.3		_	cpp File Reference		
	4.4	•		h File Reference		
		4.4.1		Description		
		4.4.2		Documentation		
			4.4.2.1	FIRST ARGUMENT		
				PROGRAM NAME	 	 12

iv	CONTENTS

		4.4.2.3	UNDEF_VALUE	. 12
4.5	src/ma	in.cpp File	Reference	. 12
	4.5.1	Function	Documentation	. 12
		4.5.1.1	main	. 12
Index				13

# **Class Index**

4	4		NI.		1	:-4
1	. 1	(	แล	22		IST

Here are the classes, structs, unions and interfaces with brief descriptions:	
Benchmark	 5
InputFiles	 6

2 Class Index

# File Index

## 2.1 File List

Here	ic a	list of	all fi	les with	hrief	descriptions
Hele	is a	1151 01	all II	ies willi	Dilei	descriptions

src/benchmark_frm.cpp	1
src/benchmark_frm.h	11
src/inputfile_txt.cpp	1
src/inputfile_txt.h	
A new input files class	11
src/main.cpp	12

File Index

## **Class Documentation**

#### 3.1 Benchmark Class Reference

```
#include <benchmark_frm.h>
```

#### **Public Member Functions**

void test (InputFiles files)

#### 3.1.1 Detailed Description

Making a framework for testing inserted data structure. Using time to estimate computational complexity.

#### 3.1.2 Member Function Documentation

#### 3.1.2.1 void Benchmark::test ( InputFiles files )

Opening file + making new table with content

Check if file is opened correctly

Testing time here

```
14
       int* tabForData = NULL;
1.5
       int tempValue = 0;
int count = 0;
16
17
       std::fstream newFile;
        for (int i = 0; i < files.return_number_files() -</pre>
       FIRST_ARGUMENT; i++){
20
            tabForData = new int[files.return_file_size(i)];
newFile.open((files.return_file_name(i) + ".txt"), std::ios::in);
22
23
             assert(newFile.is_open() && ("I can't open file."));
             for (int j = 0; j < files.return_file_size(i); j++) {</pre>
28
                 newFile >> tempValue;
tabForData[j] = tempValue;
29
30
            newFile.close();
            measureTime(tabForData, files.return_file_size(i));
35
36
             delete[] tabForData;
38
        }
```

6 Class Documentation

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

- src/benchmark\_frm.h
- src/benchmark\_frm.cpp

### 3.2 InputFiles Class Reference

```
#include <inputfile_txt.h>
```

#### **Public Member Functions**

• InputFiles ()

A default constructor.

InputFiles (int filNr, std::vector< int >filSiz)

A constructor.

· void show\_info ()

Show info about files.

· void generate\_random\_int\_data ()

Create random integers data into files.

• const std::string return\_file\_name (int Nmbr)

Return names of files (only for read purpose)

· const int return\_file\_size (int Nmbr)

Return sizes of files (only for read purpose)

• const int return\_number\_files ()

Return number of files.

#### 3.2.1 Detailed Description

Making an object which contain text files with generated random integer numbers.

#### 3.2.2 Constructor & Destructor Documentation

```
3.2.2.1 InputFiles::InputFiles ( )
```

A default constructor.

Adding number of files(UNDEF\_VALUE = 1); Generating file name; Adding size of file (UNDEF\_VALUE = 1); Just in case, when program starts without any parameters.

A new file object class source code

```
filesNumber = UNDEF_VALUE;

filesNumber = UNDEF_VALUE;

//TODO: EXCEPTIONS HANDLING
std::string TempName = std::tmpnam(nullptr);

filesNamesTab.push_back(TempName);
filesSizes.push_back(UNDEF_VALUE);

}
```

3.2.2.2 InputFiles::InputFiles ( int  $\mathit{filNr}$ ,  $\mathit{std}$ ::vector< int >  $\mathit{filSiz}$  )

A constructor.

Adding number of files; Generating files names; Adding sizes of files; Parameters inherit from list of arguments from command prompt

8 Class Documentation

#### **Parameters**

filNr	number of files
filSiz	sizes of files

Create new names for files

Delete all prohibit char from string

Open files with new names

Check if file is opened correctly

```
15
         filesNumber = filNr;
         filesSizes = filSiz;
16
17
19
         std::string TempName;
         for (int i = 1; i < filesNumber; i++) {
    TempName = std::tmpnam(nullptr);</pre>
20
21
              boost::algorithm::erase_all(TempName, "/");
boost::algorithm::erase_all(TempName, "\\");
24
25
26
27
              filesNamesTab.push_back(TempName);
         }
29
31
         std::ofstream NewFile;
         for (int i = 1; i < filesNumber; i++) {
   NewFile.open(filesNamesTab[i - PROGRAM_NAME] + ".txt");</pre>
32
33
               assert(NewFile.is_open() && "I can't open this file.");
35
36
              NewFile.close();
37
38 }
```

#### 3.2.3 Member Function Documentation

3.2.3.1 void InputFiles::generate\_random\_int\_data()

Create random integers data into files.

Seed for Mersenne Twister 19937 generator

Mersenne Twister 19937 generator

More info about this generator: http://pl.wikipedia.org/wiki/Mersenne\_Twister

Uniform distribution random number

Max number: uncomment next line More info about this distribution: http://pl.wikipedia.org/wiki/← Rozk%C5%82ad\_jednostajny

Check if file is opened correctly

```
55
56
          int seedGen = time(NULL);
59
61
          std::mt19937 randomNumbr(seedGen);
65
66
68
73
          //std::cout << std::numeric_limits<int>::max() << std::endl;</pre>
74
          std::uniform_int_distribution<>newDistr;
75
76
          std::ofstream NewFile;
78
          for (int i = 1; i < filesNumber; i++) {</pre>
                NewFile.open((filesNamesTab[i - PROGRAM_NAME] + ".txt"),std::ios::in);
assert(NewFile.is_open() && ("I can't open file."));
for (int j = 0; j < filesSizes[i - FIRST_ARGUMENT]; j++){
    NewFile << newDistr(randomNumbr) << "\n";</pre>
81
82
83
84
85
                NewFile.close();
87 }
```

**3.2.3.2** const std::string InputFiles::return\_file\_name( int Nmbr ) [inline]

Return names of files (only for read purpose)

10 Class Documentation

#### **Parameters**

Nmbr | number of the file

```
69
70         return filesNamesTab[Nmbr];
71    }
```

3.2.3.3 const int InputFiles::return\_file\_size ( int Nmbr ) [inline]

Return sizes of files (only for read purpose)

#### **Parameters**

```
Nmbr number of the file
```

3.2.3.4 const int InputFiles::return\_number\_files() [inline]

Return number of files.

```
82
83     return filesNumber;
84 }
```

3.2.3.5 void InputFiles::show\_info()

Show info about files.

Display: number of files, names of files, sizes of files

```
40
41
        std::cout << "----" << std::endl;
43
        std::cout << filesNumber - FIRST_ARGUMENT << std::endl;</pre>
        for (int i = 0; i < (signed) filesNamesTab.size(); i++) {
    std::cout << filesNamesTab[i] << std::endl;</pre>
45
46
47
        for (int i = 0; i < (signed) filesSizes.size(); i++) {</pre>
49
            std::cout << filesSizes[i] << std::endl;</pre>
50
51
        std::cout << "----" << std::endl;
52
53 }
```

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

- src/inputfile\_txt.h
- src/inputfile\_txt.cpp

## **File Documentation**

### 4.1 src/benchmark\_frm.cpp File Reference

```
#include "benchmark_frm.h"
```

### 4.2 src/benchmark\_frm.h File Reference

```
#include <vector>
#include <fstream>
#include <boost\timer\timer.hpp>
#include "inputfile_txt.h"
```

#### Classes

• class Benchmark

### 4.3 src/inputfile\_txt.cpp File Reference

```
#include "inputfile_txt.h"
```

### 4.4 src/inputfile\_txt.h File Reference

#### A new input files class.

```
#include <iostream>
#include <string>
#include <fstream>
#include <vector>
#include <cstdio>
#include <cassert>
#include <ctime>
#include <random>
#include <boost/algorithm/string/erase.hpp>
```

12 File Documentation

#### Classes

class InputFiles

#### **Variables**

```
• const int FIRST_ARGUMENT = 1
```

- First argument from command prompt (name of the program)
   const int UNDEF\_VALUE = 1
- const int PROGRAM\_NAME = 1

#### 4.4.1 Detailed Description

A new input files class.

#### 4.4.2 Variable Documentation

```
4.4.2.1 const int FIRST_ARGUMENT = 1
```

First argument from command prompt (name of the program)

```
4.4.2.2 const int PROGRAM_NAME = 1
```

4.4.2.3 const int UNDEF\_VALUE = 1

### 4.5 src/main.cpp File Reference

```
#include "inputfile_txt.h"
#include "benchmark_frm.h"
```

#### **Functions**

• int main (int argc, char \*argv[])

#### 4.5.1 Function Documentation

```
4.5.1.1 int main ( int argc, char * argv[])
```

Container for sizes from command prompt

First argument is a name of the program so i = 1

```
5 {
7     std::vector<int>FilesSizes;
8
10     for (int i = 1; i < argc; i++)
11         FilesSizes.push_back(atoi(argv[i]));
12
13     InputFiles newFilesList(argc, FilesSizes);
14     Benchmark NewTest;
15     newFilesList.generate_random_int_data();
16     NewTest.test(newFilesList);
17 }</pre>
```

## Index

```
Benchmark, 5
    test, 5
FIRST_ARGUMENT
    inputfile_txt.h, 12
generate_random_int_data
    InputFiles, 8
InputFiles, 6
    generate_random_int_data, 8
    InputFiles, 6
    return_file_name, 8
    return_file_size, 10
    return_number_files, 10
    show_info, 10
inputfile_txt.h
    FIRST_ARGUMENT, 12
     PROGRAM_NAME, 12
    UNDEF_VALUE, 12
main
    main.cpp, 12
main.cpp
    main, 12
PROGRAM_NAME
    inputfile_txt.h, 12
return_file_name
    InputFiles, 8
return_file_size
    InputFiles, 10
return_number_files
    InputFiles, 10
show info
    InputFiles, 10
src/benchmark_frm.cpp, 11
src/benchmark_frm.h, 11
src/inputfile_txt.cpp, 11
src/inputfile_txt.h, 11
src/main.cpp, 12
test
    Benchmark, 5
UNDEF_VALUE
```

inputfile\_txt.h, 12