

Lab1

1.0

Generated by Doxygen 1.8.9.1

Wed Mar 11 2015 15:21:46

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	Benchmark Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Function Documentation	5
3.1.2.1	test	5
3.2	InputFiles Class Reference	6
3.2.1	Detailed Description	6
3.2.2	Constructor & 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 Documentation	11
4.1	src/benchmark_frm.cpp File Reference	11
4.2	src/benchmark_frm.h File Reference	11
4.3	src/inputfile_txt.cpp File Reference	11
4.4	src/inputfile_txt.h File Reference	11
4.4.1	Detailed Description	12
4.4.2	Variable Documentation	12
4.4.2.1	FIRST_ARGUMENT	12
4.4.2.2	PROGRAM_NAME	12

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

Chapter 1

Class Index

1.1 Class List

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

Benchmark	5
InputFiles	6

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

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

Chapter 3

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

```
13         {
14             int* tabForData = NULL;
15             int tempValue = 0;
16             int count = 0;
17             std::fstream newFile;
18
19             for (int i = 0; i < files.return_number_files() -
FIRST_ARGUMENT; i++){
20
21                 tabForData = new int[files.return_file_size(i)];
22                 newFile.open((files.return_file_name(i) + ".txt"), std::ios::in);
23
24                 assert(newFile.is_open() && ("I can't open file."));
25
26                 for (int j = 0; j < files.return_file_size(i); j++){
27                     newFile >> tempValue;
28                     tabForData[j] = tempValue;
29                 }
30                 newFile.close();
31
32                 measureTime(tabForData, files.return_file_size(i));
33                 delete[] tabForData;
34             }
35         }
```

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

```
4         {
5     filesNumber = UNDEF_VALUE;
6
7     //TODO: EXCEPTIONS HANDLING
8     std::string TempName = std::tmpnam(nullptr);
9
10    fileNameTab.push_back(TempName);
11    fileSizeTab.push_back(UNDEF_VALUE);
12 }
```

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

A constructor.

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

Parameters

<i>filNr</i>	number of files
<i>filSiz</i>	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

```

14                                     {
15     filesNumber = filNr;
16     filesSizes = filSiz;
17
18     std::string TempName;
19     for (int i = 1; i < filesNumber; i++){
20         TempName = std::tmpnam(nullptr);
21
22         boost::algorithm::erase_all(TempName, "/");
23         boost::algorithm::erase_all(TempName, "\\");
24
25         filesNamesTab.push_back(TempName);
26     }
27
28     std::ofstream NewFile;
29     for (int i = 1; i < filesNumber; i++){
30         NewFile.open(filesNamesTab[i - PROGRAM_NAME] + ".txt");
31         assert(NewFile.is_open() && "I can't open this file.");
32         NewFile.close();
33     }
34 }

```

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
57     int seedGen = time(NULL);
58
59     std::mt19937 randomNumbr(seedGen);
60
61     //std::cout << std::numeric_limits<int>::max() << std::endl;
62     std::uniform_int_distribution<>newDistr;
63
64     std::ofstream NewFile;
65
66     for (int i = 1; i < filesNumber; i++){
67         NewFile.open((filesNamesTab[i - PROGRAM_NAME] + ".txt"), std::ios::in);
68         assert(NewFile.is_open() && ("I can't open file."));
69         for (int j = 0; j < filesSizes[i - FIRST_ARGUMENT]; j++){
70             NewFile << newDistr(randomNumbr) << "\n";
71         }
72         NewFile.close();
73     }
74 }

```

3.2.3.2 `const std::string InputFiles::return_file_name (int Nmbr)` `[inline]`

Return names of files (only for read purpose)

Parameters

<i>Nmbr</i>	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

<i>Nmbr</i>	number of the file
-------------	--------------------

```

77                                     {
78         return filesSizes[Nmbr];
79     }

```

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
42     std::cout << "-----" << std::endl;
43     std::cout << filesNumber - FIRST_ARGUMENT << std::endl;
44
45     for (int i = 0; i < (signed)filesNamesTab.size(); i++){
46         std::cout << filesNamesTab[i] << std::endl;
47     }
48     for (int i = 0; i < (signed)filesSizes.size(); i++){
49         std::cout << filesSizes[i] << std::endl;
50     }
51
52     std::cout << "-----" << std::endl;
53 }

```

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

- [src/inputfile_txt.h](#)
- [src/inputfile_txt.cpp](#)

Chapter 4

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>
```

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


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](#)