

Objektinio programavimo projektas

Generated by Doxygen 1.11.0

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 stud Class Reference	7
4.1.1 Constructor & Destructor Documentation	8
4.1.1.1 stud() [1/5]	8
4.1.1.2 stud() [2/5]	8
4.1.1.3 stud() [3/5]	8
4.1.1.4 ~stud()	9
4.1.1.5 stud() [4/5]	9
4.1.1.6 stud() [5/5]	9
4.1.2 Member Function Documentation	9
4.1.2.1 AddMark()	9
4.1.2.2 CalculateFinalMark()	9
4.1.2.3 compareMark()	9
4.1.2.4 compareName()	9
4.1.2.5 compareSurname()	9
4.1.2.6 GenerateRandomGrades() [1/2]	10
4.1.2.7 GenerateRandomGrades() [2/2]	10
4.1.2.8 GenerateRandomName() [1/2]	10
4.1.2.9 GenerateRandomName() [2/2]	10
4.1.2.10 getEgzamRez()	10
4.1.2.11 getFinalMark()	10
4.1.2.12 getHomeWorkRez()	10
4.1.2.13 getHomeWorkSize()	10
4.1.2.14 getName()	10
4.1.2.15 getSurname()	11
4.1.2.16 operator=() [1/2]	11
4.1.2.17 operator=() [2/2]	11
4.1.2.18 ReadStudent()	11
4.1.2.19 setEgzamRez()	11
4.1.2.20 setHomeWorkRez()	11
4.1.2.21 setName()	11
4.1.2.22 setSurname()	11
4.1.3 Friends And Related Symbol Documentation	12
4.1.3.1 operator<<	12

4.1.3.2 operator>>	12
4.1.4 Member Data Documentation	12
4.1.4.1 egz	12
4.1.4.2 FinalMark	12
4.1.4.3 nd	12
4.2 zmogus Class Reference	12
4.2.1 Constructor & Destructor Documentation	13
4.2.1.1 ~zmogus()	13
4.2.2 Member Function Documentation	13
4.2.2.1 getName()	13
4.2.2.2 getSurname()	13
4.2.2.3 setName()	13
4.2.2.4 setSurname()	13
4.2.3 Member Data Documentation	14
4.2.3.1 pav	14
4.2.3.2 vard	14
5 File Documentation	15
5.1 build/CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c File Reference	15
5.1.1 Macro Definition Documentation	15
5.1.1.1 ARCHITECTURE_ID	15
5.1.1.2 C_DIALECT	16
5.1.1.3 COMPILER_ID	16
5.1.1.4 DEC	16
5.1.1.5 HEX	16
5.1.1.6 PLATFORM_ID	16
5.1.1.7 STRINGIFY	16
5.1.1.8 STRINGIFY_HELPER	16
5.1.2 Function Documentation	17
5.1.2.1 main()	17
5.1.3 Variable Documentation	17
5.1.3.1 info_arch	17
5.1.3.2 info_compiler	17
5.1.3.3 info_language_dialect_default	17
5.1.3.4 info_platform	17
5.2 build/CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference	17
5.2.1 Macro Definition Documentation	18
5.2.1.1 ARCHITECTURE_ID	18
5.2.1.2 COMPILER_ID	18
5.2.1.3 CXX_STD	18
5.2.1.4 DEC	18
5.2.1.5 HEX	18

5.2.1.6 PLATFORM_ID	19
5.2.1.7 STRINGIFY	19
5.2.1.8 STRINGIFY_HELPER	19
5.2.2 Function Documentation	19
5.2.2.1 main()	19
5.2.3 Variable Documentation	19
5.2.3.1 info_arch	19
5.2.3.2 info_compiler	19
5.2.3.3 info_language_dialect_default	19
5.2.3.4 info_platform	20
5.3 DataManip.cpp File Reference	20
5.3.1 Function Documentation	20
5.3.1.1 sorting()	20
5.3.1.2 SplitVector()	20
5.4 DataManip.h File Reference	20
5.4.1 Function Documentation	20
5.4.1.1 sorting()	20
5.4.1.2 SplitVector()	21
5.5 DataManip.h	21
5.6 ErrorHandling.cpp File Reference	21
5.6.1 Function Documentation	21
5.6.1.1 ClearCin()	21
5.6.1.2 ExamParameters()	21
5.6.1.3 GradingParameters()	21
5.6.1.4 StartParameters()	22
5.6.1.5 StringParameters()	22
5.7 ErrorHandling.h File Reference	22
5.7.1 Function Documentation	22
5.7.1.1 ClearCin()	22
5.7.1.2 ExamParameters()	22
5.7.1.3 GradingParameters()	22
5.7.1.4 StartParameters()	22
5.7.1.5 StringParameters()	22
5.8 ErrorHandling.h	23
5.9 IOmanip.cpp File Reference	23
5.9.1 Function Documentation	23
5.9.1.1 Assign()	23
5.9.1.2 CreateFile()	23
5.9.1.3 OutputTime()	23
5.9.1.4 Print()	23
5.9.1.5 PrintFile()	24
5.9.1.6 readFile()	24

5.10 IManip.h File Reference	24
5.10.1 Function Documentation	24
5.10.1.1 Assign()	24
5.10.1.2 CreateFile()	24
5.10.1.3 OutputTime()	24
5.10.1.4 Print()	25
5.10.1.5 PrintFile()	25
5.10.1.6 readFile()	25
5.11 IManip.h	25
5.12 lib.h File Reference	25
5.12.1 Variable Documentation	26
5.12.1.1 AverageTime	26
5.12.1.2 Marktime	26
5.12.1.3 Names	26
5.12.1.4 Surnames	26
5.13 lib.h	26
5.14 student.cpp File Reference	27
5.14.1 Variable Documentation	27
5.14.1.1 Names	27
5.14.1.2 Surnames	27
5.15 student.hpp File Reference	27
5.15.1 Variable Documentation	28
5.15.1.1 CountByAvg	28
5.16 student.hpp	28
5.17 Tests.cpp File Reference	30
5.17.1 Function Documentation	30
5.17.1.1 CinInputTest()	30
5.17.1.2 CopyOpConTest()	31
5.17.1.3 IstreamInputTest()	31
5.17.1.4 MoveOpConTest()	31
5.17.1.5 OutputTest()	31
5.17.1.6 OutputTestFile()	31
5.18 Tests.h File Reference	31
5.18.1 Function Documentation	32
5.18.1.1 CinInputTest()	32
5.18.1.2 CopyOpConTest()	32
5.18.1.3 IstreamInputTest()	32
5.18.1.4 MoveOpConTest()	32
5.18.1.5 OutputTest()	32
5.18.1.6 OutputTestFile()	32
5.19 Tests.h	32
5.20 vector.cpp File Reference	33

5.20.1 Function Documentation	33
5.20.1.1 main()	33
5.20.2 Variable Documentation	33
5.20.2.1 AverageTime	33
5.20.2.2 CountByAvg	33
5.20.2.3 Marktime	33
Index	35

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

zmogus	12
stud	7

Chapter 2

Class Index

2.1 Class List

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

stud	7
zmogus	12

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

DataManip.cpp	20
DataManip.h	20
ErrorHandling.cpp	21
ErrorHandling.h	22
IOmanip.cpp	23
IOmanip.h	24
lib.h	25
student.cpp	27
student.hpp	27
Tests.cpp	30
Tests.h	31
vector.cpp	33
build/CMakeFiles/3.16.3/CompilerIdC/ CMakeCCompilerId.c	15
build/CMakeFiles/3.16.3/CompilerIdCXX/ CMakeCXXCompilerId.cpp	17

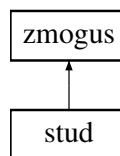
Chapter 4

Class Documentation

4.1 stud Class Reference

```
#include <student.hpp>
```

Inheritance diagram for stud:



Public Member Functions

- `stud ()` noexcept
- `stud (std::istream &is)` noexcept
- `stud (string vard_, string pav_, int egz_, vector< int > nd_)` noexcept
- `float getEgzamRez ()` const
- `void setEgzamRez (int i)`
- `string getName ()` const
- `void setName (string str)`
- `string getSurname ()` const
- `void setSurname (string str)`
- `float getFinalMark ()` const
- `vector< int > & getHomeWorkRez ()`
- `int getHomeWorkSize ()`
- `void setHomeWorkRez (vector< int > vec)`
- `~stud ()`
- `float CalculateFinalMark (int sum)`
- `std::istream & ReadStudent (std::istream &is)`
- `void AddMark (int grade)`
- `void GenerateRandomGrades (int Quantity, int counter)`
- `void GenerateRandomGrades (int Quantity)`
- `void GenerateRandomName (int counter)`
- `void GenerateRandomName ()`
- `stud & operator= (const stud &obj)`
- `stud & operator= (stud &&obj)` noexcept
- `stud (const stud &obj)`
- `stud (stud &&obj)` noexcept

Public Member Functions inherited from [zmogus](#)

- virtual [~zmogus](#) ()

Static Public Member Functions

- static bool [compareName](#) (const [stud](#) &a, const [stud](#) &b)
- static bool [compareSurname](#) (const [stud](#) &a, const [stud](#) &b)
- static bool [compareMark](#) (const [stud](#) &a, const [stud](#) &b)

Private Attributes

- float [egz](#)
- vector< int > [nd](#)
- float [FinalMark](#)

Friends

- std::ostream & [operator<<](#) (std::ostream &out, const [stud](#) &obj)
- std::istream & [operator>>](#) (std::istream &in, [stud](#) &obj)

Additional Inherited Members

Protected Attributes inherited from [zmogus](#)

- string [vard](#)
- string [pav](#)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 [stud\(\)](#) [1/5]

```
stud::stud () [inline], [noexcept]
```

4.1.1.2 [stud\(\)](#) [2/5]

```
stud::stud (
    std::istream & is) [noexcept]
```

4.1.1.3 [stud\(\)](#) [3/5]

```
stud::stud (
    string vard_,
    string pav_,
    int egz_,
    vector< int > nd_) [inline], [noexcept]
```


4.1.1.4 ~stud()

```
stud::~stud () [inline]
```

4.1.1.5 stud() [4/5]

```
stud::stud (  
    const stud & obj) [inline]
```

4.1.1.6 stud() [5/5]

```
stud::stud (  
    stud && obj) [inline], [noexcept]
```

4.1.2 Member Function Documentation

4.1.2.1 AddMark()

```
void stud::AddMark (  
    int grade) [inline]
```

4.1.2.2 CalculateFinalMark()

```
float stud::CalculateFinalMark (  
    int sum)
```

4.1.2.3 compareMark()

```
static bool stud::compareMark (  
    const stud & a,  
    const stud & b) [inline], [static]
```

4.1.2.4 compareName()

```
static bool stud::compareName (  
    const stud & a,  
    const stud & b) [inline], [static]
```

4.1.2.5 compareSurname()

```
static bool stud::compareSurname (  
    const stud & a,  
    const stud & b) [inline], [static]
```

4.1.2.6 GenerateRandomGrades() [1/2]

```
void stud::GenerateRandomGrades (
    int Quantity)
```

4.1.2.7 GenerateRandomGrades() [2/2]

```
void stud::GenerateRandomGrades (
    int Quantity,
    int counter)
```

4.1.2.8 GenerateRandomName() [1/2]

```
void stud::GenerateRandomName ()
```

4.1.2.9 GenerateRandomName() [2/2]

```
void stud::GenerateRandomName (
    int counter)
```

4.1.2.10 getEgzamRez()

```
float stud::getEgzamRez () const [inline]
```

4.1.2.11 getFinalMark()

```
float stud::getFinalMark () const [inline]
```

4.1.2.12 getHomeWorkRez()

```
vector< int > & stud::getHomeWorkRez () [inline]
```

4.1.2.13 getHomeWorkSize()

```
int stud::getHomeWorkSize () [inline]
```

4.1.2.14 getName()

```
string stud::getName () const [inline], [virtual]
```

Implements [zmogus](#).

4.1.2.15 getSurname()

```
string stud::getSurname () const [inline], [virtual]
```

Implements [zmogus](#).

4.1.2.16 operator=() [1/2]

```
stud & stud::operator= (  
    const stud & obj) [inline]
```

4.1.2.17 operator=() [2/2]

```
stud & stud::operator= (  
    stud && obj) [inline], [noexcept]
```

4.1.2.18 ReadStudent()

```
std::istream & stud::ReadStudent (  
    std::istream & is)
```

4.1.2.19 setEgzamRez()

```
void stud::setEgzamRez (  
    int i) [inline]
```

4.1.2.20 setHomeWorkRez()

```
void stud::setHomeWorkRez (  
    vector< int > vec) [inline]
```

4.1.2.21 setName()

```
void stud::setName (  
    string str) [inline], [virtual]
```

Reimplemented from [zmogus](#).

4.1.2.22 setSurname()

```
void stud::setSurname (  
    string str) [inline], [virtual]
```

Reimplemented from [zmogus](#).

4.1.3 Friends And Related Symbol Documentation

4.1.3.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & out,  
    const stud & obj) [friend]
```

4.1.3.2 operator>>

```
std::istream & operator>> (  
    std::istream & in,  
    stud & obj) [friend]
```

4.1.4 Member Data Documentation

4.1.4.1 egz

```
float stud::egz [private]
```

4.1.4.2 FinalMark

```
float stud::FinalMark [private]
```

4.1.4.3 nd

```
vector<int> stud::nd [private]
```

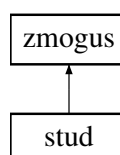
The documentation for this class was generated from the following files:

- [student.hpp](#)
- [student.cpp](#)

4.2 zmogus Class Reference

```
#include <student.hpp>
```

Inheritance diagram for zmogus:



Public Member Functions

- virtual string [getName](#) () const =0
- virtual void [setName](#) (string str)
- virtual string [getSurname](#) () const =0
- virtual void [setSurname](#) (string str)
- virtual [~zmogus](#) ()

Protected Attributes

- string [vard](#)
- string [pav](#)

4.2.1 Constructor & Destructor Documentation

4.2.1.1 ~zmogus()

```
virtual zmogus::~~zmogus () [inline], [virtual]
```

4.2.2 Member Function Documentation

4.2.2.1 getName()

```
virtual string zmogus::getName () const [pure virtual]
```

Implemented in [stud](#).

4.2.2.2 getSurname()

```
virtual string zmogus::getSurname () const [pure virtual]
```

Implemented in [stud](#).

4.2.2.3 setName()

```
virtual void zmogus::setName (  
    string str) [inline], [virtual]
```

Reimplemented in [stud](#).

4.2.2.4 setSurname()

```
virtual void zmogus::setSurname (  
    string str) [inline], [virtual]
```

Reimplemented in [stud](#).

4.2.3 Member Data Documentation

4.2.3.1 pav

```
string zmogus::pav [protected]
```

4.2.3.2 vard

```
string zmogus::vard [protected]
```

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

- [student.hpp](#)

Chapter 5

File Documentation

5.1 build/CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c File Reference

Macros

- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define C_DIALECT`

Functions

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

Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_dialect_default`

5.1.1 Macro Definition Documentation

5.1.1.1 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

5.1.1.2 C_DIALECT

```
#define C_DIALECT
```

5.1.1.3 COMPILER_ID

```
#define COMPILER_ID ""
```

5.1.1.4 DEC

```
#define DEC(  
    n)
```

Value:

```
('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + ((n) % 10))
```

5.1.1.5 HEX

```
#define HEX(  
    n)
```

Value:

```
('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))
```

5.1.1.6 PLATFORM_ID

```
#define PLATFORM_ID
```

5.1.1.7 STRINGIFY

```
#define STRINGIFY(  
    X) STRINGIFY_HELPER(X)
```

5.1.1.8 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(  
    X) #X
```


5.1.2 Function Documentation

5.1.2.1 main()

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

5.1.3 Variable Documentation

5.1.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

5.1.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

5.1.3.3 info_language_dialect_default

```
const char* info_language_dialect_default
```

Initial value:

```
=  
"INFO" ":" "dialect_default[" C_DIALECT "]"
```

5.1.3.4 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

5.2 build/CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

Macros

- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define CXX_STD __cplusplus`

Functions

- int [main](#) (int argc, char *argv[])

Variables

- char const * [info_compiler](#) = "INFO" ":" "compiler[" COMPILER_ID "]"
- char const * [info_platform](#) = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const * [info_arch](#) = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
- const char * [info_language_dialect_default](#)

5.2.1 Macro Definition Documentation

5.2.1.1 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

5.2.1.2 COMPILER_ID

```
#define COMPILER_ID ""
```

5.2.1.3 CXX_STD

```
#define CXX_STD __cplusplus
```

5.2.1.4 DEC

```
#define DEC(  
    n)
```

Value:

```
('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + ((n) % 10))
```

5.2.1.5 HEX

```
#define HEX(  
    n)
```

Value:

```
('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))
```

5.2.1.6 PLATFORM_ID

```
#define PLATFORM_ID
```

5.2.1.7 STRINGIFY

```
#define STRINGIFY(  
    X) STRINGIFY_HELPER(X)
```

5.2.1.8 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(  
    X) #X
```

5.2.2 Function Documentation

5.2.2.1 main()

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

5.2.3 Variable Documentation

5.2.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

5.2.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

5.2.3.3 info_language_dialect_default

```
const char* info_language_dialect_default
```

Initial value:

```
= "INFO" ":" "dialect_default["
```

```
"98"
```

```
"]"
```

5.2.3.4 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

5.3 DataManip.cpp File Reference

```
#include "DataManip.h"
```

Functions

- void [sorting](#) (vector< [stud](#) > &obj)
- void [SplitVector](#) (vector< [stud](#) > &obj, vector< [stud](#) > &SAD, vector< [stud](#) > &COOL)

5.3.1 Function Documentation

5.3.1.1 sorting()

```
void sorting (  
    vector< stud > & obj)
```

5.3.1.2 SplitVector()

```
void SplitVector (  
    vector< stud > & obj,  
    vector< stud > & SAD,  
    vector< stud > & COOL)
```

5.4 DataManip.h File Reference

```
#include "lib.h"  
#include "student.hpp"  
#include "ErrorHandling.h"
```

Functions

- void [sorting](#) (vector< [stud](#) > &obj)
- void [SplitVector](#) (vector< [stud](#) > &obj, vector< [stud](#) > &SAD, vector< [stud](#) > &COOL)

5.4.1 Function Documentation

5.4.1.1 sorting()

```
void sorting (  
    vector< stud > & obj)
```

5.4.1.2 SplitVector()

```
void SplitVector (
    vector< stud > & obj,
    vector< stud > & SAD,
    vector< stud > & COOL)
```

5.5 DataManip.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include "lib.h"
00004 #include "student.hpp"
00005 #include "ErrorHandling.h"
00006
00007 void sorting(vector<stud>& obj);
00008 void SplitVector(vector<stud>& obj, vector<stud> &SAD, vector<stud> &COOL);
```

5.6 ErrorHandling.cpp File Reference

```
#include "ErrorHandling.h"
```

Functions

- int [StartParameters](#) ()
- string [StringParameters](#) ()
- int [GradingParameters](#) ()
- int [ExamParameters](#) ()
- void [ClearCin](#) ()

5.6.1 Function Documentation

5.6.1.1 ClearCin()

```
void ClearCin ()
```

5.6.1.2 ExamParameters()

```
int ExamParameters ()
```

5.6.1.3 GradingParameters()

```
int GradingParameters ()
```

5.6.1.4 StartParameters()

```
int StartParameters ()
```

5.6.1.5 StringParameters()

```
string StringParameters ()
```

5.7 ErrorHandler.h File Reference

```
#include "lib.h"
```

Functions

- int [StartParameters](#) ()
- string [StringParameters](#) ()
- int [GradingParameters](#) ()
- int [ExamParameters](#) ()
- void [ClearCin](#) ()

5.7.1 Function Documentation

5.7.1.1 ClearCin()

```
void ClearCin ()
```

5.7.1.2 ExamParameters()

```
int ExamParameters ()
```

5.7.1.3 GradingParameters()

```
int GradingParameters ()
```

5.7.1.4 StartParameters()

```
int StartParameters ()
```

5.7.1.5 StringParameters()

```
string StringParameters ()
```

5.8 ErrorHandling.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include "lib.h"
00004
00005 int StartParameters();
00006 string StringParameters();
00007 int GradingParameters();
00008 int ExamParameters();
00009 void ClearCin();
```

5.9 IOmanip.cpp File Reference

```
#include "IOmanip.h"
```

Functions

- void [Assign](#) (vector< [stud](#) > &obj, bool countByAvg)
- void [Print](#) (const vector< [stud](#) > &obj)
- void [readFile](#) (vector< [stud](#) > &obj, const string filename, const bool countByAvg)
- void [PrintFile](#) (const vector< [stud](#) > &obj, string filename)
- void [CreateFile](#) (string &filename)
- void [OutputTime](#) ()

5.9.1 Function Documentation

5.9.1.1 Assign()

```
void Assign (
    vector< stud > & obj,
    bool countByAvg)
```

5.9.1.2 CreateFile()

```
void CreateFile (
    string & filename)
```

5.9.1.3 OutputTime()

```
void OutputTime ()
```

5.9.1.4 Print()

```
void Print (
    const vector< stud > & obj)
```

5.9.1.5 PrintFile()

```
void PrintFile (
    const vector< stud > & obj,
    string filename)
```

5.9.1.6 readFile()

```
void readFile (
    vector< stud > & obj,
    const string filename,
    const bool countByAvg)
```

5.10 IOmanip.h File Reference

```
#include "lib.h"
#include "DataManip.h"
#include "student.hpp"
#include "ErrorHandling.h"
```

Functions

- void [Assign](#) (vector< [stud](#) > &obj, bool countByAvg)
- void [Print](#) (const vector< [stud](#) > &obj)
- void [readFile](#) (vector< [stud](#) > &obj, const string filename, const bool countByAvg)
- void [PrintFile](#) (const vector< [stud](#) > &obj, string filename)
- void [CreateFile](#) (string &filename)
- void [OutputTime](#) ()

5.10.1 Function Documentation

5.10.1.1 Assign()

```
void Assign (
    vector< stud > & obj,
    bool countByAvg)
```

5.10.1.2 CreateFile()

```
void CreateFile (
    string & filename)
```

5.10.1.3 OutputTime()

```
void OutputTime ()
```


5.10.1.4 Print()

```
void Print (
    const vector< stud > & obj)
```

5.10.1.5 PrintFile()

```
void PrintFile (
    const vector< stud > & obj,
    string filename)
```

5.10.1.6 readFile()

```
void readFile (
    vector< stud > & obj,
    const string filename,
    const bool countByAvg)
```

5.11 IOmanip.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include "lib.h"
00004 #include "DataManip.h"
00005 #include "student.hpp"
00006 #include "ErrorHandling.h"
00007
00008 void Assign(vector<stud> &obj, bool countByAvg);
00009 void Print (const vector<stud> &obj);
00010 void readFile(vector<stud>& obj, const string filename, const bool countByAvg);
00011 void PrintFile(const vector<stud> &obj, string filename);
00012 void CreateFile(string& filename);
00013 void OutputTime();
```

5.12 lib.h File Reference

```
#include <utility>
#include <iostream>
#include <iomanip>
#include <string>
#include <limits>
#include <vector>
#include <algorithm>
#include <random>
#include <chrono>
#include <fstream>
#include <deque>
#include <list>
#include <istream>
```

Variables

- `vector< double >` [Marktime](#)
- `vector< double >` [AverageTime](#)
- `vector< string >` [Names](#)
- `vector< string >` [Surnames](#)

5.12.1 Variable Documentation

5.12.1.1 AverageTime

```
vector<double> AverageTime [extern]
```

5.12.1.2 Marktime

```
vector<double> Marktime [extern]
```

5.12.1.3 Names

```
vector<string> Names [extern]
```

5.12.1.4 Surnames

```
vector<string> Surnames [extern]
```

5.13 lib.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include <utility>
00004 #include <iostream>
00005 #include <iomanip>
00006 #include <string>
00007 #include <limits>
00008 #include <vector>
00009 #include <algorithm>
00010 #include <random>
00011 #include <chrono>
00012 #include <fstream>
00013 #include <deque>
00014 #include <list>
00015 #include <istream>
00016
00017
00018 using std::cout;
00019 using std::cin;
00020 using std::string;
00021 using std::vector;
00022 using std::deque;
00023 using std::list;
00024 using std::endl;
00025 using std::mt19937;
00026 using std::setw;
00027 using std::setprecision;
00028 using std::left;
00029 using std::fixed;
00030 using std::cerr;
```

```

00031 using std::out_of_range;
00032 using std::invalid_argument;
00033 using std::runtime_error;
00034 using std::getline;
00035 using std::numeric_limits;
00036 using std::streamsize;
00037 using std::ifstream;
00038 using std::exception;
00039 using std::stringstream;
00040 using std::ofstream;
00041 using std::ostreamstream;
00042 using std::move;
00043
00044
00045 extern vector<double> Marktime;
00046 extern vector<double> AverageTime;
00047 extern vector<string> Names;
00048 extern vector<string> Surnames;
00049

```

5.14 student.cpp File Reference

```
#include "student.hpp"
```

Variables

- vector< string > [Names](#) {"Audrius", "Edvard", "Ganesh", "Nojus", "Cleophas", "Rodrigo", "Jurgita", "Ugne", "Tatiana", "Sarah"}
- vector< string > [Surnames](#) {"Czerniewicz", "Finch", "Hummel", "McKowen", "Warszawski", "Clery", "Wilbur", "Kennedy", "Nixon", "Obama"}

5.14.1 Variable Documentation

5.14.1.1 Names

```
vector<string> Names {"Audrius", "Edvard", "Ganesh", "Nojus", "Cleophas", "Rodrigo", "Jurgita",
"Ugne", "Tatiana", "Sarah"}
```

5.14.1.2 Surnames

```
vector<string> Surnames {"Czerniewicz", "Finch", "Hummel", "McKowen", "Warszawski", "Clery",
"Wilbur", "Kennedy", "Nixon", "Obama"}
```

5.15 student.hpp File Reference

```
#include "lib.h"
```

Classes

- class [zmogus](#)
- class [stud](#)

Variables

- bool [CountByAvg](#)

5.15.1 Variable Documentation

5.15.1.1 CountByAvg

```
bool CountByAvg [extern]
```

5.16 student.hpp

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include "lib.h"
00004 extern bool CountByAvg;
00005
00006 class zmogus{
00007 protected:
00008     string vard;
00009     string pav;
00010 public:
00011     virtual string getName() const = 0; // setters and getters start
00012     virtual void setName(string str) { vard = str; }
00013
00014     virtual string getSurname() const = 0;
00015     virtual void setSurname(string str) { pav = str; } // setters and getters end
00016
00017     virtual ~zmogus(){}; // destructor
00018 };
00019
00020 class stud : public zmogus {
00021 private:
00022     float egz;
00023     vector<int> nd;
00024     float FinalMark;
00025 public:
00026     stud() noexcept : zmogus()//constructors start
00027     { vard = "";
00028       pav = "";
00029       egz = 0;
00030       nd.clear();
00031       FinalMark=0.0;
00032     }
00033     stud(std::istream &is) noexcept;
00034     stud(string vard_, string pav_, int egz_, vector<int> nd_) noexcept : zmogus()
00035     {
00036         vard = vard_;
00037         pav = pav_;
00038         egz = egz_;
00039         nd = nd_;
00040         int sum = 0;
00041         for(int i : nd){sum+=i;};
00042         CalculateFinalMark(sum);
00043     } //constructors end
00044
00045     float getEgzamRez() const { return egz; } // setters and getters start
00046     void setEgzamRez(int i) { egz = i; }
00047
00048     string getName() const {return vard;} // setters and getters start
00049     void setName(string str) { vard = str; }
00050
00051     string getSurname() const {return pav;}
00052     void setSurname(string str) { pav = str; } // setters and getters end
00053
00054     float getFinalMark() const { return FinalMark; }
00055
00056     vector<int>& getHomeWorkRez() { return nd; }
00057     int getHomeWorkSize() { return nd.size(); }
00058     void setHomeWorkRez(vector<int> vec) { nd = vec; } // setters and getters end
00059
00060     ~stud() {nd.clear();} // destructor
```

```

00061
00062     float CalculateFinalMark(int sum); // Methods start
00063
00064     std::istream& ReadStudent(std::istream& is);
00065
00066     void AddMark(int grade){nd.push_back(grade);}
00067
00068     void GenerateRandomGrades(int Quantity, int counter);
00069     void GenerateRandomGrades(int Quantity);
00070
00071     void GenerateRandomName(int counter);
00072     void GenerateRandomName();
00073
00074     bool static compareName(const stud& a, const stud& b){return a.vard < b.vard;}
00075     bool static compareSurname(const stud& a, const stud& b){return a.pav < b.pav;}
00076     bool static compareMark(const stud& a, const stud& b){return a.FinalMark < b.FinalMark;} // Methods
end
00077
00078     stud& operator=(const stud& obj) { // Rule of five
00079         if(this != &obj){
00080             vard = obj.vard;
00081             pav = obj.pav;
00082             egz = obj.egz;
00083             nd = obj.nd;
00084             FinalMark = obj.FinalMark;
00085         }
00086
00087         cout << "\nCopy operator called\n";
00088         return *this;
00089     }
00090
00091     stud& operator=(stud&& obj) noexcept {
00092         if(this != &obj){
00093             vard = move(obj.vard);
00094             pav = move(obj.pav);
00095             egz = move(obj.egz);
00096             nd = move(obj.nd);
00097             FinalMark = move(obj.FinalMark);
00098             obj.vard.clear();
00099             obj.pav.clear();
00100             obj.egz = 0;
00101             obj.nd.clear();
00102             obj.FinalMark = 0.0;
00103         }
00104         cout << "\nMove operator called\n";
00105         return *this;
00106     }
00107
00108     stud(const stud& obj):zmogus(){
00109         vard = obj.vard;
00110         pav = obj.pav;
00111         egz = obj.egz;
00112         nd = obj.nd;
00113         FinalMark = obj.FinalMark;
00114         cout << "\nCopy constructor called\n"«endl;
00115     }
00116
00117     stud(stud&& obj) noexcept:zmogus()
00118     {
00119         vard = (move(obj.vard));
00120         pav = (move(obj.pav));
00121         egz = (move(obj.egz));
00122         nd = (move(obj.nd));
00123         FinalMark = (move(obj.FinalMark));
00124         obj.vard.clear();
00125         obj.pav.clear();
00126         obj.egz = 0;
00127         obj.nd.clear();
00128         obj.FinalMark = 0.0;
00129         cout << "\nMove constructor called\n"«endl;
00130     }
00131
00132     friend std::ostream &operator<<(std::ostream &out, const stud &obj){
00133         out << setw(26) << left << obj.vard << setw(26) << left << obj.pav << setw(15) << setprecision(2) << fixed
<< left << obj.FinalMark << endl;
00134         return out;
00135     }
00136     friend std::istream &operator>>(std::istream &in, stud &obj){
00137
00138         bool pass = false;
00139         char optionForGeneration;
00140         auto Cinadress = &std::cin;
00141         if(&in == Cinadress){
00142             cout<<"Ar norite generuoti random varda ir pavarde? Y/N"«endl;
00143             in >> optionForGeneration;
00144             optionForGeneration == 'Y' ? pass = true : pass = false;
00145         }

```

```

00146     if(pass){
00147         obj.GenerateRandomName();
00148         pass = false;
00149     }
00150     else{
00151         if(&in == Cinadress){cout<<"Iveskite varda ir pavarde atskirta tarpu"«endl;}
00152         in » obj.vard;
00153         in » obj.pav;
00154     }
00155
00156     int number = 0;
00157     if(&in == Cinadress){
00158         cout<<"Ar norite generuoti random pazymius? Y/N"«endl;
00159         in » optionForGeneration;
00160         optionForGeneration == 'Y' ? pass = true : pass = false;
00161     }
00162     if(pass){
00163         cout<<"Kiek namu darbu pazymių norite?"«endl;
00164         in » number;
00165         obj.GenerateRandomGrades(number);
00166     }
00167     else{
00168         if(&in == Cinadress){cout<<"Iveskite visus pazymius (iskaitant egzamino kaip paskutini), norint
00169         baigti parasykite raide"«endl;}
00170         while(in » number){
00171             obj.nd.push_back(number);
00172         }
00173         obj.egz = obj.nd.back();
00174         obj.nd.pop_back();
00175     }
00176     if(obj.nd.empty()){
00177         cerr << "Klaida! Studento pažymių skaitymo nuo failo klaida "« endl;
00178         throw runtime_error("Klaida! Studento pažymių skaitymo nuo failo klaida ");
00179     }
00180     int ndSum = 0;
00181     for(int i : obj.nd){ndSum+=i;};
00182
00183     obj.CalculateFinalMark(ndSum);
00184
00185     return in;
00186 }
00187
00188 };

```

5.17 Tests.cpp File Reference

```
#include "Tests.h"
```

Functions

- [stud IstreamInputTest](#) ([stud](#) &Tester, string filename)
- [stud CinInputTest](#) ([stud](#) &Tester)
- void [OutputTest](#) (const [stud](#) &Tester)
- void [OutputTestFile](#) (const [stud](#) &Tester, string filename)
- void [CopyOpConTest](#) ([stud](#) &Tester1, [stud](#) &Tester2)
- void [MoveOpConTest](#) ([stud](#) &Tester1, [stud](#) &Tester2)

5.17.1 Function Documentation

5.17.1.1 CinInputTest()

```

stud CinInputTest (
    stud & Tester)

```

5.17.1.2 CopyOpConTest()

```
void CopyOpConTest (
    stud & Tester1,
    stud & Tester2)
```

5.17.1.3 IstreamInputTest()

```
stud IstreamInputTest (
    stud & Tester,
    string filename)
```

5.17.1.4 MoveOpConTest()

```
void MoveOpConTest (
    stud & Tester1,
    stud & Tester2)
```

5.17.1.5 OutputTest()

```
void OutputTest (
    const stud & Tester)
```

5.17.1.6 OutputTestFile()

```
void OutputTestFile (
    const stud & Tester,
    string filename)
```

5.18 Tests.h File Reference

```
#include "lib.h"
#include "student.hpp"
```

Functions

- void [CopyOpConTest](#) (stud &Tester1, stud &Tester2)
- void [MoveOpConTest](#) (stud &Tester1, stud &Tester2)
- void [OutputTest](#) (const stud &Tester)
- void [OutputTestFile](#) (const stud &Tester, string filename)
- [stud IstreamInputTest](#) (stud &Tester, string filename)
- [stud CinInputTest](#) (stud &Tester)

5.18.1 Function Documentation

5.18.1.1 CinInputTest()

```
stud CinInputTest (  
    stud & Tester)
```

5.18.1.2 CopyOpConTest()

```
void CopyOpConTest (  
    stud & Tester1,  
    stud & Tester2)
```

5.18.1.3 IstreamInputTest()

```
stud IstreamInputTest (  
    stud & Tester,  
    string filename)
```

5.18.1.4 MoveOpConTest()

```
void MoveOpConTest (  
    stud & Tester1,  
    stud & Tester2)
```

5.18.1.5 OutputTest()

```
void OutputTest (  
    const stud & Tester)
```

5.18.1.6 OutputTestFile()

```
void OutputTestFile (  
    const stud & Tester,  
    string filename)
```

5.19 Tests.h

[Go to the documentation of this file.](#)

```
00001 #pragma once  
00002  
00003 #include "lib.h"  
00004 #include "student.hpp"  
00005  
00006 void CopyOpConTest(stud &Tester1, stud &Tester2);  
00007 void MoveOpConTest(stud &Tester1, stud &Tester2);  
00008 void OutputTest(const stud& Tester);  
00009 void OutputTestFile(const stud& Tester, string filename);  
00010 stud IstreamInputTest(stud& Tester, string filename);  
00011 stud CinInputTest(stud& Tester);
```


5.20 vector.cpp File Reference

```
#include "lib.h"
#include "IOmanip.h"
#include "ErrorHandling.h"
#include "DataManip.h"
#include "student.hpp"
#include "Tests.h"
```

Functions

- int [main](#) ()

Variables

- vector< double > [Marktime](#)
- vector< double > [AverageTime](#)
- bool [CountByAvg](#) = true

5.20.1 Function Documentation

5.20.1.1 main()

```
int main ()
```

5.20.2 Variable Documentation

5.20.2.1 AverageTime

```
vector<double> AverageTime
```

5.20.2.2 CountByAvg

```
bool CountByAvg = true
```

5.20.2.3 Marktime

```
vector<double> Marktime
```


Index

- ~stud
 - stud, [8](#)
- ~zmogus
 - zmogus, [13](#)
- AddMark
 - stud, [9](#)
- ARCHITECTURE_ID
 - CMakeCCompilerId.c, [15](#)
 - CMakeCXXCompilerId.cpp, [18](#)
- Assign
 - IManip.cpp, [23](#)
 - IManip.h, [24](#)
- AverageTime
 - lib.h, [26](#)
 - vector.cpp, [33](#)
- build/CMakeFiles/3.16.3/CompilerIdC/CMakeCCompilerId.c,
 - [15](#)
- build/CMakeFiles/3.16.3/CompilerIdCXX/CMakeCXXCompilerId.cpp,
 - [17](#)
- C_DIALECT
 - CMakeCCompilerId.c, [15](#)
- CalculateFinalMark
 - stud, [9](#)
- CinInputTest
 - Tests.cpp, [30](#)
 - Tests.h, [32](#)
- ClearCin
 - ErrorHandling.cpp, [21](#)
 - ErrorHandling.h, [22](#)
- CMakeCCompilerId.c
 - ARCHITECTURE_ID, [15](#)
 - C_DIALECT, [15](#)
 - COMPILER_ID, [16](#)
 - DEC, [16](#)
 - HEX, [16](#)
 - info_arch, [17](#)
 - info_compiler, [17](#)
 - info_language_dialect_default, [17](#)
 - info_platform, [17](#)
 - main, [17](#)
 - PLATFORM_ID, [16](#)
 - STRINGIFY, [16](#)
 - STRINGIFY_HELPER, [16](#)
- CMakeCXXCompilerId.cpp
 - ARCHITECTURE_ID, [18](#)
 - COMPILER_ID, [18](#)
 - CXX_STD, [18](#)
- DEC, [18](#)
- HEX, [18](#)
- info_arch, [19](#)
- info_compiler, [19](#)
- info_language_dialect_default, [19](#)
- info_platform, [19](#)
- main, [19](#)
- PLATFORM_ID, [18](#)
- STRINGIFY, [19](#)
- STRINGIFY_HELPER, [19](#)
- compareMark
 - stud, [9](#)
- compareName
 - stud, [9](#)
- compareSurname
 - stud, [9](#)
- COMPILER_ID
 - CMakeCCompilerId.c, [16](#)
 - CMakeCXXCompilerId.cpp, [18](#)
- CopyOnTest
 - Tests.cpp, [30](#)
 - Tests.h, [32](#)
- CountByAvg
 - student.hpp, [28](#)
 - vector.cpp, [33](#)
- CreateFile
 - IManip.cpp, [23](#)
 - IManip.h, [24](#)
- CXX_STD
 - CMakeCXXCompilerId.cpp, [18](#)
- DataManip.cpp, [20](#)
 - sorting, [20](#)
 - SplitVector, [20](#)
- DataManip.h, [20](#)
 - sorting, [20](#)
 - SplitVector, [20](#)
- DEC
 - CMakeCCompilerId.c, [16](#)
 - CMakeCXXCompilerId.cpp, [18](#)
- egz
 - stud, [12](#)
- ErrorHandling.cpp, [21](#)
 - ClearCin, [21](#)
 - ExamParameters, [21](#)
 - GradingParameters, [21](#)
 - StartParameters, [21](#)
 - StringParameters, [22](#)
- ErrorHandling.h, [22](#)

- ClearCin, [22](#)
- ExamParameters, [22](#)
- GradingParameters, [22](#)
- StartParameters, [22](#)
- StringParameters, [22](#)
- ExamParameters
 - ErrorHandling.cpp, [21](#)
 - ErrorHandling.h, [22](#)
- FinalMark
 - stud, [12](#)
- GenerateRandomGrades
 - stud, [9](#), [10](#)
- GenerateRandomName
 - stud, [10](#)
- getEgzamRez
 - stud, [10](#)
- getFinalMark
 - stud, [10](#)
- getHomeWorkRez
 - stud, [10](#)
- getHomeWorkSize
 - stud, [10](#)
- getName
 - stud, [10](#)
 - zmogus, [13](#)
- getSurname
 - stud, [10](#)
 - zmogus, [13](#)
- GradingParameters
 - ErrorHandling.cpp, [21](#)
 - ErrorHandling.h, [22](#)
- HEX
 - CMakeCCompilerId.c, [16](#)
 - CMakeCXXCompilerId.cpp, [18](#)
- info_arch
 - CMakeCCompilerId.c, [17](#)
 - CMakeCXXCompilerId.cpp, [19](#)
- info_compiler
 - CMakeCCompilerId.c, [17](#)
 - CMakeCXXCompilerId.cpp, [19](#)
- info_language_dialect_default
 - CMakeCCompilerId.c, [17](#)
 - CMakeCXXCompilerId.cpp, [19](#)
- info_platform
 - CMakeCCompilerId.c, [17](#)
 - CMakeCXXCompilerId.cpp, [19](#)
- IOmanip.cpp, [23](#)
 - Assign, [23](#)
 - CreateFile, [23](#)
 - OutputTime, [23](#)
 - Print, [23](#)
 - PrintFile, [23](#)
 - readFile, [24](#)
- IOmanip.h, [24](#)
 - Assign, [24](#)
 - CreateFile, [24](#)
 - OutputTime, [24](#)
 - Print, [24](#)
 - PrintFile, [25](#)
 - readFile, [25](#)
- IstreamInputTest
 - Tests.cpp, [31](#)
 - Tests.h, [32](#)
- lib.h, [25](#)
 - AverageTime, [26](#)
 - Marktime, [26](#)
 - Names, [26](#)
 - Surnames, [26](#)
- main
 - CMakeCCompilerId.c, [17](#)
 - CMakeCXXCompilerId.cpp, [19](#)
 - vector.cpp, [33](#)
- Marktime
 - lib.h, [26](#)
 - vector.cpp, [33](#)
- MoveOpConTest
 - Tests.cpp, [31](#)
 - Tests.h, [32](#)
- Names
 - lib.h, [26](#)
 - student.cpp, [27](#)
- nd
 - stud, [12](#)
- operator<<
 - stud, [12](#)
- operator>>
 - stud, [12](#)
- operator=
 - stud, [11](#)
- OutputTest
 - Tests.cpp, [31](#)
 - Tests.h, [32](#)
- OutputTestFile
 - Tests.cpp, [31](#)
 - Tests.h, [32](#)
- OutputTime
 - IOmanip.cpp, [23](#)
 - IOmanip.h, [24](#)
- pav
 - zmogus, [14](#)
- PLATFORM_ID
 - CMakeCCompilerId.c, [16](#)
 - CMakeCXXCompilerId.cpp, [18](#)
- Print
 - IOmanip.cpp, [23](#)
 - IOmanip.h, [24](#)
- PrintFile
 - IOmanip.cpp, [23](#)
 - IOmanip.h, [25](#)

- readFile
 - IManip.cpp, 24
 - IManip.h, 25
- ReadStudent
 - stud, 11
- setEgzamRez
 - stud, 11
- setHomeWorkRez
 - stud, 11
- setName
 - stud, 11
 - zmogus, 13
- setSurname
 - stud, 11
 - zmogus, 13
- sorting
 - DataManip.cpp, 20
 - DataManip.h, 20
- SplitVector
 - DataManip.cpp, 20
 - DataManip.h, 20
- StartParameters
 - ErrorHandling.cpp, 21
 - ErrorHandling.h, 22
- STRINGIFY
 - CMakeCCompilerId.c, 16
 - CMakeCXXCompilerId.cpp, 19
- STRINGIFY_HELPER
 - CMakeCCompilerId.c, 16
 - CMakeCXXCompilerId.cpp, 19
- StringParameters
 - ErrorHandling.cpp, 22
 - ErrorHandling.h, 22
- stud, 7
 - ~stud, 8
 - AddMark, 9
 - CalculateFinalMark, 9
 - compareMark, 9
 - compareName, 9
 - compareSurname, 9
 - egz, 12
 - FinalMark, 12
 - GenerateRandomGrades, 9, 10
 - GenerateRandomName, 10
 - getEgzamRez, 10
 - getFinalMark, 10
 - getHomeWorkRez, 10
 - getHomeWorkSize, 10
 - getName, 10
 - getSurname, 10
 - nd, 12
 - operator<<, 12
 - operator>>, 12
 - operator=, 11
 - ReadStudent, 11
 - setEgzamRez, 11
 - setHomeWorkRez, 11
 - setName, 11
 - setSurname, 11
 - stud, 8, 9
- student.cpp, 27
 - Names, 27
 - Surnames, 27
- student.hpp, 27
 - CountByAvg, 28
- Surnames
 - lib.h, 26
 - student.cpp, 27
- Tests.cpp, 30
 - CinInputTest, 30
 - CopyOpConTest, 30
 - IstreamInputTest, 31
 - MoveOpConTest, 31
 - OutputTest, 31
 - OutputTestFile, 31
- Tests.h, 31
 - CinInputTest, 32
 - CopyOpConTest, 32
 - IstreamInputTest, 32
 - MoveOpConTest, 32
 - OutputTest, 32
 - OutputTestFile, 32
- vard
 - zmogus, 14
- vector.cpp, 33
 - AverageTime, 33
 - CountByAvg, 33
 - main, 33
 - Marktime, 33
- zmogus, 12
 - ~zmogus, 13
 - getName, 13
 - getSurname, 13
 - pav, 14
 - setName, 13
 - setSurname, 13
 - vard, 14