

Student Data Application

Generated by Doxygen 1.12.0

1 Programos naudojimo instrukcija	1
1.0.1 Testavimo įranga:	1
1.0.2 Programos versijų aprašas ir spartos analizė	1
1.0.2.1 Versija 0.1	1
1.0.2.2 Versija 0.2	1
1.0.2.3 Versija 0.3	2
1.0.2.4 Versija 1.0	2
1.0.2.5 Versija 1.1	2
1.0.3 Struct ir Class veikimo palyginimas	2
1.0.4 O1, O2 ir O3 optimizavimo flag'ų palyginimas	2
1.0.4.1 Versija 1.2	2
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 Human Class Reference	9
5.1.1 Detailed Description	9
5.1.2 Constructor & Destructor Documentation	10
5.1.2.1 ~Human()	10
5.1.3 Member Function Documentation	10
5.1.3.1 getPavarde()	10
5.1.3.2 getVardas()	10
5.1.3.3 setPavarde()	10
5.1.3.4 setVardas()	10
5.1.4 Member Data Documentation	11
5.1.4.1 pavarde_	11
5.1.4.2 vardas_	11
5.2 StudentasClass Class Reference	11
5.2.1 Detailed Description	13
5.2.2 Constructor & Destructor Documentation	13
5.2.2.1 StudentasClass() [1/3]	13
5.2.2.2 StudentasClass() [2/3]	13
5.2.2.3 ~StudentasClass()	13
5.2.2.4 StudentasClass() [3/3]	13
5.2.3 Member Function Documentation	14
5.2.3.1 arIslaike()	14
5.2.3.2 clear()	14

5.2.3.3 compare()	14
5.2.3.4 generuotiBalus()	14
5.2.3.5 getEgzamRez()	15
5.2.3.6 getGalutinis()	15
5.2.3.7 getMediana()	15
5.2.3.8 getPavarde()	15
5.2.3.9 getTarpRez()	16
5.2.3.10 getVardas()	16
5.2.3.11 getVidurkis()	16
5.2.3.12 isimtiGalutini()	16
5.2.3.13 operator=()	16
5.2.3.14 pridetiTarpRez()	17
5.2.3.15 printInfo()	17
5.2.3.16 rastiGalutini()	17
5.2.3.17 rastislaike()	17
5.2.3.18 setEgzamRez()	17
5.2.3.19 setPavarde()	18
5.2.3.20 setTarpRez()	18
5.2.3.21 setVardas()	18
5.2.4 Friends And Related Symbol Documentation	18
5.2.4.1 operator<<	18
5.2.4.2 operator>>	19
5.3 Timer Class Reference	19
5.3.1 Detailed Description	19
5.3.2 Constructor & Destructor Documentation	19
5.3.2.1 Timer()	19
5.3.3 Member Function Documentation	19
5.3.3.1 elapsed()	19
5.3.3.2 reset()	19
6 File Documentation	21
6.1 include/lib.h File Reference	21
6.2 lib.h	21
6.3 include/studentas.h File Reference	21
6.3.1 Function Documentation	22
6.3.1.1 generateEntries()	22
6.3.1.2 inputManual()	22
6.3.1.3 inputManualList()	23
6.3.1.4 inputScan()	23
6.3.1.5 inputSplitSort()	23
6.3.1.6 inputSplitSortImpl()	24
6.3.1.7 outputManual()	24

6.3.1.8 outputScan()	24
6.4 studentas.h	25
6.5 README.md File Reference	26
6.6 src/failuGeneratorius.cpp File Reference	26
6.6.1 Function Documentation	27
6.6.1.1 generateEntries()	27
6.6.1.2 inputSplitSort()	27
6.6.1.3 inputSplitSortImpl()	27
6.7 src/main.cpp File Reference	28
6.7.1 Function Documentation	28
6.7.1.1 main()	28
6.8 src/studentas.cpp File Reference	28
6.8.1 Function Documentation	28
6.8.1.1 inputManual()	28
6.8.1.2 inputManualList()	29
6.8.1.3 inputScan()	29
6.8.1.4 operator<<()	29
6.8.1.5 operator>>()	29
6.8.1.6 outputManual()	29
6.8.1.7 outputScan()	30
Index	31

Chapter 1

Programos naudojimo instrukcija

1. Atsisiųsti v1.0 release paketą ir CMAKE 3.25 versiją;
2. Atsidaryti terminalą ir jame pakeisti dabartinę direktoriją į atsisiųsto (un-zip'into) kodo aplanką ("cd ./objektinis-pirma-obj-4" ar panašiai);
3. Terminale paleisti run.sh script'ą su komanda "./run.sh". *PASTABA*: jeigu nepavyksta, panaudoti komandą "chmod +x run.sh";
4. Programai pasileidus, skaityti ekrane pasirodžiusias instrukcijas ir jomis vadovautis.

1.0.1 Testavimo įranga:

- CPU: Apple M1
- RAM: 8 GB LPDDR4X-4266
- SSD: 256 GB ~2700 MB/s

1.0.2 Programos versijų aprašas ir spartos analizė

1.0.2.1 Versija 0.1

- Generuoja studentų informaciją: namų darbai ir egzamino rezultatai
- Leidžia duomenis įvesti rankomis arba nuskaityti iš failo
- Informacija išvedama į failą sulygiuota ir išrikiuota pagal pavardę

Panaudotos bibliotekos: iostream, vector, sstream, random, algorithm, fstream, iomanip, stdexception.

1.0.2.2 Versija 0.2

- Generuoja studentų informaciją: namų darbai ir egzamino rezultatai
- Leidžia duomenis įvesti rankomis arba nuskaityti iš failo
- Informacija išvedama į failą sulygiuota ir išrikiuota pagal pavardę
- *Studentus suskirsto į "protingus" ir "kvailus" ir juos išveda į atskirus failus*

Panaudotos bibliotekos: iostream, vector, sstream, random, algorithm, fstream, iomanip, stdexception, chrono.

1.0.2.3 Versija 0.3

- Generuoja studentų informaciją: namų darbai ir egzamino rezultatai
- Leidžia duomenis įvesti rankomis arba nuskaityti iš failo
- Informacija išvedama į failą sulygiuota ir išrikiuota pagal pavardę
- Studentus suskirsto į "protingus" ir "kvailus" ir juos išveda į atskirus failus
- *Fiksuoja programos spartos laiką vector ir list konteineriams: įrašų generavimas, nuskaitymas, rikiavimas, studentų įrašų išskirtymas į "protingus" ir "kvailus"*

Panaudotos bibliotekos: iostream, vector, sstream, random, algorithm, fstream, iomanip, stdexception, chrono, list.

1.0.2.4 Versija 1.0

- Generuoja studentų informaciją: namų darbai ir egzamino rezultatai
- Leidžia duomenis įvesti rankomis arba nuskaityti iš failo
- Informacija išvedama į failą sulygiuota ir išrikiuota pagal pavardę
- Studentus suskirsto į "protingus" ir "kvailus" ir juos išveda į atskirus failus
- Fiksuoja programos spartos laiką vector ir list konteineriams: įrašų generavimas, nuskaitymas, rikiavimas, studentų įrašų išskirtymas į "protingus" ir "kvailus"
- *Studentų įrašų duomenis galima apdoroti pagal 3 skirtingas strategijas: (1) padalinti į naujus 2 konteinerius, (2) padalinti sukuriant tik 1 naują konteinerį, (3) taikant greičiausią būdą ir pritaikant efektyvius STL algoritmus*

Panaudotos bibliotekos: iostream, vector, sstream, random, algorithm, fstream, iomanip, stdexception, chrono, list, numeric.

1.0.2.5 Versija 1.1

1.0.3 Struct ir Class veikimo palyginimas

1.0.4 O1, O2 ir O3 optimizavimo flag'ų palyginimas

1.0.4.1 Versija 1.2

Įvedimo ir išvedimo member metodai class'ei Studentas buvo perdengti. Įvedimo metodas pilnai nuskaityto duomenų eilutę sugeneruotame faile (vardas, pavardė ir visi gauti įvertinimai). Kai kuriais atvejais įvedimo metodas rankinio įvedimo metu nepanaudotas dėl skirtingų duomenų reikalavimų. Išvedimo metodas taip pat pritaikytas, nebent prašoma išvesti skirtingus duomenis (pvz. prašo vidurkio ir medianos, o metodas išveda tik vieną).

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Human	9
StudentasClass	11
Timer	19

Chapter 3

Class Index

3.1 Class List

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

Human	
Abstract father class for StudentasClass (p. 11)	9
StudentasClass	
Student data processing	11
Timer	
Tracks time for run-time measurements	19

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

include/ lib.h	21
include/ studentas.h	21
src/ failuGeneratorius.cpp	26
src/ main.cpp	28
src/ studentas.cpp	28

Chapter 5

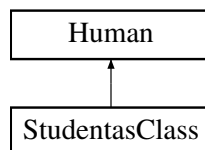
Class Documentation

5.1 Human Class Reference

Abstract father class for **StudentasClass** (p. 11).

```
#include <studentas.h>
```

Inheritance diagram for Human:



Public Member Functions

- virtual **~Human** ()=0
- virtual void **setVardas** (const std::string &vardas)=0
Set the First Name object.
- virtual void **setPavarde** (const std::string &pavarde)=0
Set the Last Name object.
- virtual const std::string & **getVardas** () const =0
Get the First Name object.
- virtual const std::string & **getPavarde** () const =0
Get the Last Name object.

Protected Attributes

- std::string **vardas_**
- std::string **pavarde_**

5.1.1 Detailed Description

Abstract father class for **StudentasClass** (p. 11).

5.1.2 Constructor & Destructor Documentation

5.1.2.1 ~Human()

```
Human::~~Human () [inline], [pure virtual]
```

5.1.3 Member Function Documentation

5.1.3.1 getPavarde()

```
virtual const std::string & Human::getPavarde () const [pure virtual]
```

Get the Last Name object.

Returns

const std::string&

Implemented in **StudentasClass** (p. 15).

5.1.3.2 getVardas()

```
virtual const std::string & Human::getVardas () const [pure virtual]
```

Get the First Name object.

Returns

const std::string&

Implemented in **StudentasClass** (p. 16).

5.1.3.3 setPavarde()

```
virtual void Human::setPavarde (  
    const std::string & pavarde) [pure virtual]
```

Set the Last Name object.

Parameters

<i>pavarde</i>	
----------------	--

Implemented in **StudentasClass** (p. 18).

5.1.3.4 setVardas()

```
virtual void Human::setVardas (  
    const std::string & vardas) [pure virtual]
```

Set the First Name object.

Parameters

<code>vardas</code>	
---------------------	--

Implemented in **StudentasClass** (p. 18).

5.1.4 Member Data Documentation

5.1.4.1 pavarde_

```
std::string Human::pavarde_ [protected]
```

5.1.4.2 vardas_

```
std::string Human::vardas_ [protected]
```

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

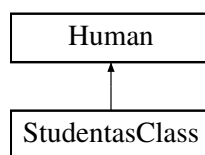
- include/ **studentas.h**

5.2 StudentasClass Class Reference

Student data processing.

```
#include <studentas.h>
```

Inheritance diagram for StudentasClass:



Public Member Functions

- **StudentasClass** ()
Construct a new Studentas Class object.
- **StudentasClass** (const std::string &vardas, const std::string &pavarde)
Construct a new Studentas Class object.
- **~StudentasClass** ()
Destroy the Studentas Class object.
- **StudentasClass** (const **StudentasClass** &other)
Construct a new Studentas Class object by copying data.
- **StudentasClass** & **operator=** (const **StudentasClass** &other)
Construct a new Studentas Class object by assigning data.

- const std::string & **getVardas** () const override
Get the First Name object.
- const std::string & **getPavarde** () const override
Get the Last Name object.
- const std::vector< int > & **getTarpRez** () const
Get the Intermediate Grades object.
- int **getEgzamRez** () const
Get the Exam Grade object.
- double **getVidurkis** () const
Get the Grade Average object.
- double **getMediana** () const
Get the Grade Median object.
- double **getGalutinis** () const
Get the Final Grade object.
- void **setVardas** (const std::string &vardas) override
Set the First Name object.
- void **setPavarde** (const std::string &pavarde) override
Set the Last Name object.
- void **setEgzamRez** (int rez)
Set the Exam Grade object.
- void **setTarpRez** (const std::vector< int > &naujiTarpRez)
Set the Intermediate Grades object.
- void **pridetiTarpRez** (int rez)
Add a Single Intermediate Grade.
- void **generuotiBalus** (int kiekBalu=15)
Generate random Intermediate Grades and Exam Grade.
- void **rastiGalutini** (bool naudotiVidurki=true)
Calculate the Final Grade.
- void **isimtiGalutini** ()
Remove the last Intermediate Grade and replace it with Exam Grade.
- void **clear** ()
Remove all saved data.
- void **rastislaike** ()
Update Passing Grade boolean islaike.
- bool **arIslaike** () const
Recheck and return Passing Grade boolean.
- bool **compare** (const **StudentasClass** &b, int criteria=2)
Compares Students either by First Name, Last Name or Final Grade.
- std::ostream & **printInfo** (std::ostream &os) const

Public Member Functions inherited from **Human**

- virtual ~**Human** ()=0

Friends

- std::istream & **operator>>** (std::istream &is, **StudentasClass** &s)
- std::ostream & **operator<<** (std::ostream &os, const **StudentasClass** &s)

Additional Inherited Members

Protected Attributes inherited from Human

- std::string **vardas_**
- std::string **pavarde_**

5.2.1 Detailed Description

Student data processing.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 StudentasClass() [1/3]

```
StudentasClass::StudentasClass () [inline]
```

Construct a new Studentas Class object.

5.2.2.2 StudentasClass() [2/3]

```
StudentasClass::StudentasClass (
    const std::string & vardas,
    const std::string & pavarde) [inline]
```

Construct a new Studentas Class object.

Parameters

<i>vardas</i>	
<i>pavarde</i>	

5.2.2.3 ~StudentasClass()

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

Destroy the Studentas Class object.

5.2.2.4 StudentasClass() [3/3]

```
StudentasClass::StudentasClass (
    const StudentasClass & other) [inline]
```

Construct a new Studentas Class object by copying data.

Parameters

<i>other</i>	
--------------	--

5.2.3 Member Function Documentation

5.2.3.1 arIsIaike()

```
bool StudentasClass::arIsIaike () const [inline]
```

Recheck and return Passing Grade boolean.

Returns

true
false

5.2.3.2 clear()

```
void StudentasClass::clear ()
```

Remove all saved data.

5.2.3.3 compare()

```
bool StudentasClass::compare (  
    const StudentasClass & b,  
    int criteria = 2)
```

Compares Students either by First Name, Last Name or Final Grade.

Parameters

<i>b</i>	
<i>criteria</i>	

Returns

true
false

5.2.3.4 generuotiBalus()

```
void StudentasClass::generuotiBalus (  
    int kiekBalus = 15)
```

Generate random Intermediate Grades and Exam Grade.

Parameters

<i>kiekBalu</i>	
-----------------	--

5.2.3.5 getEgzamRez()

```
int StudentasClass::getEgzamRez () const [inline]
```

Get the Exam Grade object.

Returns

int

5.2.3.6 getGalutinis()

```
double StudentasClass::getGalutinis () const [inline]
```

Get the Final Grade object.

Returns

double

5.2.3.7 getMediana()

```
double StudentasClass::getMediana () const [inline]
```

Get the Grade Median object.

Returns

double

5.2.3.8 getPavarde()

```
const std::string & StudentasClass::getPavarde () const [inline], [override], [virtual]
```

Get the Last Name object.

Returns

const std::string&

Implements **Human** (p. 10).

5.2.3.9 getTarpRez()

```
const std::vector< int > & StudentasClass::getTarpRez () const [inline]
```

Get the Intermediate Grades object.

Returns

const std::vector<int>&

5.2.3.10 getVardas()

```
const std::string & StudentasClass::getVardas () const [inline], [override], [virtual]
```

Get the First Name object.

Returns

const std::string&

Implements **Human** (p. 10).

5.2.3.11 getVidurkis()

```
double StudentasClass::getVidurkis () const [inline]
```

Get the Grade Average object.

Returns

double

5.2.3.12 isimtiGalutini()

```
void StudentasClass::isimtiGalutini ()
```

Remove the last Intermediate Grade and replace it with Exam Grade.

5.2.3.13 operator=()

```
StudentasClass & StudentasClass::operator= (  
    const StudentasClass & other) [inline]
```

Construct a new Studentas Class object by assigning data.

Parameters

<i>other</i>	
--------------	--

Returns

StudentasClass (p.11)&

5.2.3.14 pridetiTarpRez()

```
void StudentasClass::pridetiTarpRez (  
    int rez)
```

Add a Single Intermediate Grade.

Parameters

<i>rez</i>	
------------	--

5.2.3.15 printInfo()

```
std::ostream & StudentasClass::printInfo (  
    std::ostream & os) const
```

5.2.3.16 rastiGalutini()

```
void StudentasClass::rastiGalutini (  
    bool naudotiVidurki = true)
```

Calculate the Final Grade.

Parameters

<i>naudotiVidurki</i>	
-----------------------	--

5.2.3.17 rastiIslaike()

```
void StudentasClass::rastiIslaike () [inline]
```

Update Passing Grade boolean islaike.

5.2.3.18 setEgzamRez()

```
void StudentasClass::setEgzamRez (  
    int rez) [inline]
```

Set the Exam Grade object.

Parameters

<i>rez</i>	
------------	--

5.2.3.19 setPavarde()

```
void StudentasClass::setPavarde (  
    const std::string & pavarde) [inline], [override], [virtual]
```

Set the Last Name object.

Parameters

<i>pavarde</i>	
----------------	--

Implements **Human** (p.10).

5.2.3.20 setTarpRez()

```
void StudentasClass::setTarpRez (  
    const std::vector< int > & naujiTarpRez)
```

Set the Intermediate Grades object.

Parameters

<i>naujiTarpRez</i>	
---------------------	--

5.2.3.21 setVardas()

```
void StudentasClass::setVardas (  
    const std::string & vardas) [inline], [override], [virtual]
```

Set the First Name object.

Parameters

<i>vardas</i>	
---------------	--

Implements **Human** (p.10).

5.2.4 Friends And Related Symbol Documentation**5.2.4.1 operator<<**

```
std::ostream & operator<< (  
    std::ostream & os,  
    const StudentasClass & s) [friend]
```


5.2.4.2 operator>>

```
std::istream & operator>> (  
    std::istream & is,  
    StudentasClass & s) [friend]
```

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

- include/ **studentas.h**
- src/ **studentas.cpp**

5.3 Timer Class Reference

Tracks time for run-time measurements.

Public Member Functions

- **Timer** ()
- void **reset** ()
- double **elapsed** () const

5.3.1 Detailed Description

Tracks time for run-time measurements.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Timer()

```
Timer::Timer () [inline]
```

5.3.3 Member Function Documentation

5.3.3.1 elapsed()

```
double Timer::elapsed () const [inline]
```

5.3.3.2 reset()

```
void Timer::reset () [inline]
```

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

- src/ **failuGeneratorius.cpp**

Chapter 6

File Documentation

6.1 include/lib.h File Reference

```
#include <iostream>
#include <vector>
#include <sstream>
#include <random>
#include <algorithm>
#include <fstream>
#include <iomanip>
#include <stdexcept>
#include <chrono>
#include <list>
#include <numeric>
```

6.2 lib.h

Go to the documentation of this file.

```
00001 #ifndef LIB_H_INCLUDED
00002 #define LIB_H_INCLUDED
00003
00004 #include <iostream>
00005 #include <vector>
00006 #include <sstream> // getline išskirstymui
00007 #include <random> // atsitiktiniai skaičiai
00008 #include <algorithm> // sort
00009 #include <fstream> // darbas su failais
00010 #include <iomanip> // setw, setprecision, std::left, std::right
00011 #include <stdexcept> // try catch
00012 #include <chrono> // funkcijos darbo laiko fiksavimas
00013 #include <list>
00014 #include <numeric>
00015
00016 #endif
```

6.3 include/studentas.h File Reference

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

Classes

- class **Human**
*Abstract father class for **StudentasClass** (p. 11).*
- class **StudentasClass**
Student data processing.

Functions

- void **inputManual** (std::vector< **StudentasClass** > &studentai, int studSk)
Lets the user manually input Student data into a Vector.
- void **inputManualList** (std::list< **StudentasClass** > &studentaiList, int studSk)
Lets the user manually input Student data into a List.
- void **inputScan** (std::vector< **StudentasClass** > &studentai, std::string failoPav)
Reads Student data from a File.
- void **outputManual** (**StudentasClass** Lok, int vidMed)
Outputs Student data from nanual user input and displays the Grade Average or Median.
- void **outputScan** (std::vector< **StudentasClass** > &studentai)
Outputs Student data into a summary File.
- void **generateEntries** (int studGenSk, int ndGenSk)
Randomly generates Student data.
- void **inputSplitSort** (std::string failoPav, int rusiavKateg, int useVector, int testStrat)
Intermediate function for choosing vector or list containers.
- void **inputSplitSortImpl** (std::string failoPav, int rusiavKateg, int testStrat)
Processes Student data.

6.3.1 Function Documentation

6.3.1.1 generateEntries()

```
void generateEntries (
    int studGenSk,
    int ndGenSk)
```

Randomly generates Student data.

Parameters

<i>studGenSk</i>	Number of students
<i>ndGenSk</i>	Number of homework grades

6.3.1.2 inputManual()

```
void inputManual (
    std::vector< StudentasClass > & studentai,
    int studSk)
```

Lets the user manually input Student data into a Vector.

Parameters

<i>studentai</i>	Vector of Students
<i>studSk</i>	Number of Students

6.3.1.3 inputManualList()

```
void inputManualList (
    std::list< StudentasClass > & studentaiList,
    int studSk)
```

Lets the user manually input Student data into a List.

Parameters

<i>studentaiList</i>	List of Students
<i>studSk</i>	Number of Students

6.3.1.4 inputScan()

```
void inputScan (
    std::vector< StudentasClass > & studentai,
    std::string failoPav)
```

Reads Student data from a File.

Parameters

<i>studentai</i>	
<i>failoPav</i>	

6.3.1.5 inputSplitSort()

```
void inputSplitSort (
    std::string failoPav,
    int rusiavKateg,
    int useVector,
    int testStrat)
```

Intermediate function for choosing vector or list containers.

Parameters

<i>failoPav</i>	Name of file containing Student data
<i>rusiavKateg</i>	Sorting by First name, Last name or Final grade
<i>useVector</i>	
<i>testStrat</i>	Sorting methodology option

6.3.1.6 inputSplitSortImpl()

```
void inputSplitSortImpl (
    std::string failoPav,
    int rusiavKateg,
    int testStrat)
```

Processes Student data.

Template Parameters

<i>Container</i>	Vector or list
------------------	----------------

Parameters

<i>failoPav</i>	Name of file containing Student data
<i>rusiavKateg</i>	Sorting by First name, Last name or Final grade
<i>testStrat</i>	Sorting methodology option

6.3.1.7 outputManual()

```
void outputManual (
    StudentasClass Lok,
    int vidMed)
```

Outputs Student data from manual user input and displays the Grade Average or Median.

Parameters

<i>Lok</i>	
<i>vidMed</i>	Grade Average or Median

6.3.1.8 outputScan()

```
void outputScan (
    std::vector< StudentasClass > & studentai)
```

Outputs Student data into a summary File.

Parameters

<i>studentai</i>	
------------------	--

6.4 studentas.h

Go to the documentation of this file.

```

00001 #ifndef STUDENTAS_H_INCLUDED
00002 #define STUDENTAS_H_INCLUDED
00003
00004 #include "lib.h"
00005
00011 class Human {
00012 protected:
00013     std::string vardas_;
00014     std::string pavarde_;
00015
00016 public:
00017     virtual ~Human() = 0;
00018
00019     // Visiškai virtualūs metodai
00025     virtual void setVardas(const std::string& vardas) = 0;
00031     virtual void setPavarde(const std::string& pavarde) = 0;
00032
00038     virtual const std::string& getVardas() const = 0;
00044     virtual const std::string& getPavarde() const = 0;
00045 };
00046
00047 inline Human::~Human() {}
00048
00054 class StudentasClass : public Human{
00055 private:
00056     std::vector<int> tarpRez_;
00057     int egzamRez_;
00058     double vidurkis_;
00059     double mediana_;
00060     double galutinis_;
00061     bool islaike_;
00062
00063     // Privatūs pagalbiniai metodai
00064     void rastiVid();
00065     void rastiMed();
00066     void rastiRez();
00067
00068 public:
00074     // Default konstruktorius
00075     StudentasClass() :
00076         Human(),
00077         egzamRez_(0),
00078         vidurkis_(0.0),
00079         mediana_(0.0),
00080         galutinis_(0.0),
00081         islaike_(false) {}
00082
00090     // Parametrizuotas konstruktorius
00091     StudentasClass(const std::string& vardas, const std::string& pavarde) :
00092         Human(),
00093         egzamRez_(0),
00094         vidurkis_(0.0),
00095         mediana_(0.0),
00096         galutinis_(0.0),
00097         islaike_(false) {
00098         this->setVardas(vardas);
00099         this->setPavarde(pavarde);
00100     }
00101
00107     ~StudentasClass() {
00108         tarpRez_.clear();
00109     };
00110
00117     // Copy konstruktorius
00118     StudentasClass(const StudentasClass& other) : Human() {
00119         this->setVardas(other.getVardas());
00120         this->setPavarde(other.getPavarde());
00121         tarpRez_ = other.tarpRez_;
00122         egzamRez_ = other.egzamRez_;
00123         vidurkis_ = other.vidurkis_;
00124         mediana_ = other.mediana_;
00125         galutinis_ = other.galutinis_;
00126         islaike_ = other.islaike_;
00127     }
00128
00136     // Copy assignment konstruktorius
00137     StudentasClass& operator=(const StudentasClass& other) {
00138         if (this != &other) { // Apsauga nuo savęs priskyrimo
00139             vardas_ = other.vardas_;
00140             pavarde_ = other.pavarde_;
00141             tarpRez_ = other.tarpRez_;
00142             egzamRez_ = other.egzamRez_;

```

```

00143         vidurkis_ = other.vidurkis_;
00144         mediana_ = other.mediana_;
00145         galutinis_ = other.galutinis_;
00146         islaike_ = other.islaike_;
00147     }
00148     return *this;
00149 }
00150
00151 // Get'eriai
00152 const std::string& getVardas() const override { return vardas_; }
00153 const std::string& getPavarde() const override { return pavarde_; }
00154 const std::vector<int>& getTarpRez() const { return tarpRez_; }
00155 int getEgzamRez() const { return egzamRez_; }
00156 double getVidurkis() const { return vidurkis_; }
00157 double getMediana() const { return mediana_; }
00158 double getGalutinis() const { return galutinis_; }
00159
00160 // Set'eriai
00161 void setVardas(const std::string& vardas) override { vardas_ = vardas; }
00162 void setPavarde(const std::string& pavarde) override { pavarde_ = pavarde; }
00163 void setEgzamRez(int rez) { egzamRez_ = rez; rastiGalutini(); }
00164 void setTarpRez(const std::vector<int>& naujiTarpRez);
00165
00166 // Metodai
00167 void pridetiTarpRez(int rez);
00168 void generuotiBalus(int kiekBalu = 15);
00169 void rastiGalutini(bool naudotiVidurki = true);
00170 void isimtiGalutini();
00171 void clear();
00172 void rastiIslaike(){ islaike_ = (galutinis_ >= 5.0); }
00173 bool arIslaike() const { return islaike_; }
00174 bool compare(const StudentasClass& b, int criteria = 2);
00175 std::ostream& printInfo(std::ostream& os) const;
00176
00177 //Friend'ai įvedimo ir išvedimo operacijoms
00178 friend std::istream& operator>(std::istream& is, StudentasClass& s);
00179 friend std::ostream& operator<(std::ostream& os, const StudentasClass& s);
00180 };
00181
00182 // Pre-v1.0, ne member metodai darbui su klase Studentas
00183 void inputManual(std::vector<StudentasClass> &studentai, int studSk);
00184 void inputManualList(std::list<StudentasClass> &studentaiList, int studSk);
00185 void inputScan(std::vector<StudentasClass> &studentai, std::string failoPav);
00186 void outputManual(StudentasClass Lok, int vidMed);
00187 void outputScan(std::vector<StudentasClass> &studentai);
00188 void generateEntries(int studGenSk, int ndGenSk);
00189 void inputSplitSort(std::string failoPav, int rusiavKateg, int useVector, int testStrat);
00190 void inputSplitSortImpl(std::string failoPav, int rusiavKateg, int testStrat);
00191
00192 #endif

```

6.5 README.md File Reference

6.6 src/failuGeneratorius.cpp File Reference

```

#include "lib.h"
#include "studentas.h"

```

Classes

- class **Timer**

Tracks time for run-time measurements.

Functions

- void **generateEntries** (int studGenSk, int ndGenSk)
Randomly generates Student data.
- template<typename Container >
void **inputSplitSortImpl** (std::string failoPav, int rusiavKateg, int testStrat)
- void **inputSplitSort** (std::string failoPav, int rusiavKateg, int useVector, int testStrat)
Intermediate function for choosing vector or list containers.

6.6.1 Function Documentation

6.6.1.1 generateEntries()

```
void generateEntries (
    int studGenSk,
    int ndGenSk)
```

Randomly generates Student data.

Parameters

<i>studGenSk</i>	Number of students
<i>ndGenSk</i>	Number of homework grades

6.6.1.2 inputSplitSort()

```
void inputSplitSort (
    std::string failoPav,
    int rusiavKateg,
    int useVector,
    int testStrat)
```

Intermediate function for choosing vector or list containers.

Parameters

<i>failoPav</i>	Name of file containing Student data
<i>rusiavKateg</i>	Sorting by First name, Last name or Final grade
<i>useVector</i>	
<i>testStrat</i>	Sorting methodology option

6.6.1.3 inputSplitSortImpl()

```
template<typename Container >
void inputSplitSortImpl (
    std::string failoPav,
    int rusiavKateg,
    int testStrat)
```

6.7 src/main.cpp File Reference

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

Functions

- int **main** ()

6.7.1 Function Documentation

6.7.1.1 main()

```
int main ()
```

6.8 src/studentas.cpp File Reference

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

Functions

- std::istream & **operator**>> (std::istream &is, **StudentasClass** &s)
- std::ostream & **operator**<< (std::ostream &os, const **StudentasClass** &s)
- void **inputManual** (std::vector< **StudentasClass** > &studentai, int studSk)

Lets the user manually input Student data into a Vector.
- void **inputManualList** (std::list< **StudentasClass** > &studentaiList, int studSk)

Lets the user manually input Student data into a List.
- void **inputScan** (std::vector< **StudentasClass** > &studentai, std::string failoPav)

Reads Student data from a File.
- void **outputManual** (**StudentasClass** stud, int vidMed)

Outputs Student data from nanual user input and displays the Grade Average or Median.
- void **outputScan** (std::vector< **StudentasClass** > &studentai)

Outputs Student data into a summary File.

6.8.1 Function Documentation

6.8.1.1 inputManual()

```
void inputManual (
    std::vector< StudentasClass > & studentai,
    int studSk)
```

Lets the user manually input Student data into a Vector.

Parameters

<i>studentai</i>	Vector of Students
<i>studSk</i>	Number of Students

6.8.1.2 inputManualList()

```
void inputManualList (
    std::list< StudentasClass > & studentaiList,
    int studSk)
```

Lets the user manually input Student data into a List.

Parameters

<i>studentaiList</i>	List of Students
<i>studSk</i>	Number of Students

6.8.1.3 inputScan()

```
void inputScan (
    std::vector< StudentasClass > & studentai,
    std::string failoPav)
```

Reads Student data from a File.

Parameters

<i>studentai</i>	
<i>failoPav</i>	

6.8.1.4 operator<<()

```
std::ostream & operator<< (
    std::ostream & os,
    const StudentasClass & s)
```

6.8.1.5 operator>>()

```
std::istream & operator>> (
    std::istream & is,
    StudentasClass & s)
```

6.8.1.6 outputManual()

```
void outputManual (
    StudentasClass Lok,
    int vidMed)
```

Outputs Student data from manual user input and displays the Grade Average or Median.

Parameters

<i>Lok</i>	
<i>vidMed</i>	Grade Average or Median

6.8.1.7 outputScan()

```
void outputScan (
    std::vector< StudentasClass > & studentai)
```

Outputs Student data into a summary File.

Parameters

<i>studentai</i>	
------------------	--

Index

- ~Human
 - Human, 10
- ~StudentasClass
 - StudentasClass, 13
- arlslaike
 - StudentasClass, 14
- clear
 - StudentasClass, 14
- compare
 - StudentasClass, 14
- elapsed
 - Timer, 19
- failuGeneratorius.cpp
 - generateEntries, 27
 - inputSplitSort, 27
 - inputSplitSortImpl, 27
- generateEntries
 - failuGeneratorius.cpp, 27
 - studentas.h, 22
- generuotiBalus
 - StudentasClass, 14
- getEgzamRez
 - StudentasClass, 15
- getGalutinis
 - StudentasClass, 15
- getMediana
 - StudentasClass, 15
- getPavarde
 - Human, 10
 - StudentasClass, 15
- getTarpRez
 - StudentasClass, 15
- getVardas
 - Human, 10
 - StudentasClass, 16
- getVidurkis
 - StudentasClass, 16
- Human, 9
 - ~Human, 10
 - getPavarde, 10
 - getVardas, 10
 - pavarde_, 11
 - setPavarde, 10
 - setVardas, 10
 - vardas_, 11
- include/lib.h, 21
- include/studentas.h, 21, 25
- inputManual
 - studentas.cpp, 28
 - studentas.h, 22
- inputManualList
 - studentas.cpp, 29
 - studentas.h, 23
- inputScan
 - studentas.cpp, 29
 - studentas.h, 23
- inputSplitSort
 - failuGeneratorius.cpp, 27
 - studentas.h, 23
- inputSplitSortImpl
 - failuGeneratorius.cpp, 27
 - studentas.h, 23
- isimtiGalutini
 - StudentasClass, 16
- main
 - main.cpp, 28
- main.cpp
 - main, 28
- operator<<
 - studentas.cpp, 29
 - StudentasClass, 18
- operator>>
 - studentas.cpp, 29
 - StudentasClass, 18
- operator=
 - StudentasClass, 16
- outputManual
 - studentas.cpp, 29
 - studentas.h, 24
- outputScan
 - studentas.cpp, 30
 - studentas.h, 24
- pavarde_
 - Human, 11
- pridetiTarpRez
 - StudentasClass, 17
- printInfo
 - StudentasClass, 17
- Programos naudojimo instrukcija, 1
- rastiGalutini
 - StudentasClass, 17

- rastilslaike
 - StudentasClass, 17
- README.md, 26
- reset
 - Timer, 19
- setEgzamRez
 - StudentasClass, 17
- setPavarde
 - Human, 10
 - StudentasClass, 18
- setTarpRez
 - StudentasClass, 18
- setVardas
 - Human, 10
 - StudentasClass, 18
- src/failuGeneratorius.cpp, 26
- src/main.cpp, 28
- src/studentas.cpp, 28
- studentas.cpp
 - inputManual, 28
 - inputManualList, 29
 - inputScan, 29
 - operator<<, 29
 - operator>>, 29
 - outputManual, 29
 - outputScan, 30
- studentas.h
 - generateEntries, 22
 - inputManual, 22
 - inputManualList, 23
 - inputScan, 23
 - inputSplitSort, 23
 - inputSplitSortImpl, 23
 - outputManual, 24
 - outputScan, 24
- StudentasClass, 11
 - ~StudentasClass, 13
 - arIslaike, 14
 - clear, 14
 - compare, 14
 - generuotiBalus, 14
 - getEgzamRez, 15
 - getGalutinis, 15
 - getMediana, 15
 - getPavarde, 15
 - getTarpRez, 15
 - getVardas, 16
 - getVidurkis, 16
 - isimtiGalutini, 16
 - operator<<, 18
 - operator>>, 18
 - operator=, 16
 - pridetiTarpRez, 17
 - printInfo, 17
 - rastiGalutini, 17
 - rastilslaike, 17
 - setEgzamRez, 17
 - setPavarde, 18
 - setTarpRez, 18
 - setVardas, 18
 - StudentasClass, 13
- Timer, 19
 - elapsed, 19
 - reset, 19
 - Timer, 19
- vardas_
 - Human, 11