

Studentu Pazymiu Skaiciavimo Programa

Generated by Doxygen 1.12.0

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 Detailed Description	8
4.1.2 Constructor & Destructor Documentation	8
4.1.2.1 Stud() [1/3]	8
4.1.2.2 Stud() [2/3]	8
4.1.2.3 ~Stud()	9
4.1.2.4 Stud() [3/3]	9
4.1.3 Member Function Documentation	9
4.1.3.1 getEgz()	9
4.1.3.2 getNd()	9
4.1.3.3 getOnePaz()	9
4.1.3.4 getPavarde()	9
4.1.3.5 getVardas()	9
4.1.3.6 operator=()	9
4.1.3.7 setEgz()	10
4.1.3.8 setNd()	10
4.1.3.9 setOnePaz()	10
4.1.3.10 setPavarde()	10
4.1.3.11 setVardas()	10
4.1.4 Friends And Related Symbol Documentation	10
4.1.4.1 operator<<	10
4.1.4.2 operator>>	11
4.2 Zmogus Class Reference	11
4.2.1 Detailed Description	12
4.2.2 Constructor & Destructor Documentation	12
4.2.2.1 Zmogus() [1/2]	12
4.2.2.2 Zmogus() [2/2]	12
4.2.2.3 ~Zmogus()	12
4.2.3 Member Function Documentation	12
4.2.3.1 getPavarde()	12
4.2.3.2 getVardas()	12
4.2.3.3 setPavarde()	12
4.2.3.4 setVardas()	13

4.2.4 Member Data Documentation	13
4.2.4.1 pavarde	13
4.2.4.2 vardas	13
5 File Documentation	15
5.1 C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/header.h File Reference	15
5.2 header.h	15
5.3 C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/main.cpp File Reference	15
5.3.1 Function Documentation	16
5.3.1.1 main()	16
5.4 C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/Stud.cpp File Reference	16
5.4.1 Function Documentation	17
5.4.1.1 atsitiktiniaiPazymiai()	17
5.4.1.2 dabLaikas()	17
5.4.1.3 galutinis()	17
5.4.1.4 generuotiFailus()	18
5.4.1.5 iverstiFaila()	18
5.4.1.6 iverstiDuomenisRanka()	18
5.4.1.7 ndMediana()	18
5.4.1.8 ndVidurkis()	19
5.4.1.9 nuskaitytiFaila()	19
5.4.1.10 paskirtytiStud()	19
5.4.1.11 patikrinti()	20
5.4.1.12 spausdinti()	20
5.4.1.13 test()	20
5.5 C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/Stud.h File Reference	20
5.5.1 Function Documentation	21
5.5.1.1 atsitiktiniaiPazymiai()	21
5.5.1.2 dabLaikas()	22
5.5.1.3 galutinis()	22
5.5.1.4 generuotiFailus()	22
5.5.1.5 iverstiFaila()	22
5.5.1.6 iverstiDuomenisRanka()	23
5.5.1.7 ndMediana()	23
5.5.1.8 ndVidurkis()	23
5.5.1.9 nuskaitytiFaila()	23
5.5.1.10 paskirtytiStud()	24
5.5.1.11 patikrinti()	24
5.5.1.12 spausdinti()	24
5.5.1.13 test()	24
5.6 Stud.h	25
Index	27

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Zmogus	11
Stud	7

Chapter 2

Class Index

2.1 Class List

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

Stud	7
Zmogus	11

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/ header.h	15
C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/ main.cpp	15
C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/ Stud.cpp	16
C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/ Stud.h	20

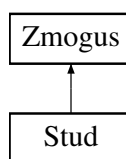
Chapter 4

Class Documentation

4.1 Stud Class Reference

```
#include <Stud.h>
```

Inheritance diagram for Stud:



Public Member Functions

- `Stud ()`
Konstruktorius, kuris vardui pavardei priskiria tuscias string, namu darbai yra tuscias vektrius ir egzamino pazymys lygus 0.
- `Stud (const string &v, const string &p, const vector< int > n, int e)`
- `~Stud ()`
Destruktorius, kuris isvalo visus duomenis.
- `Stud (const Stud &kitas)`
Kopijavimo konstruktorius, kuris sukuria kopija is kito studento objekto.
- `Stud & operator= (const Stud &kitas)`
Kopijavimo priskyrimo operatorius, kuris priskiria vieno studento duomenis kitam.
- `string getVardas () const`
- `void setVardas (const string &v)`
- `string getPavarde () const`
- `void setPavarde (const string &p)`
- `vector< int > getNd () const`
- `void setNd (const vector< int > &n)`
- `int getOnePaz (int i) const`
- `void setOnePaz (int paz)`
- `int getEgz () const`
- `void setEgz (const int &e)`

Public Member Functions inherited from [Zmogus](#)

- [Zmogus](#) ()
- [Zmogus](#) (const string &v, const string &p)
- virtual [~Zmogus](#) ()=0
- string [getVardas](#) () const
- string [getPavarde](#) () const
- string [setVardas](#) (const string &v)
- string [setPavarde](#) (const string &p)

Friends

- istream & [operator>>](#) (istream &is, [Stud](#) &stud)
Ivesties operatorius, kuriuo ivedami studento duomenys.
- ostream & [operator<<](#) (ostream &out, const [Stud](#) &stud)
Išvesties operatorius, kuris išveda studento duomenis.

Additional Inherited Members

Protected Attributes inherited from [Zmogus](#)

- string [vardas](#)
- string [pavarde](#)

4.1.1 Detailed Description

Derived klase [Stud](#), kuri apibrezia studento objekto savybes Paveldi viska is klases [Zmogus](#) ir turi savo du parametus - namu darbus ir egzamina. Turi savo default ir parametrizuotus konstruktorius, destruktorius, getterius ir setterius. Apibrezti kopijavimo ir kopijavimo priskyrimo konstruktoriai

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Stud() [1/3]

```
Stud::Stud () [inline]
```

Konstruktorius, kuris vardui pavardei priskiria tuscia string, namu darbai yra tuscias vektrius ir egzamino pazymys lygus 0.

4.1.2.2 Stud() [2/3]

```
Stud::Stud (
    const string & v,
    const string & p,
    const vector< int > n,
    int e) [inline]
```

Konstruktorius, kuris suteikia visiems klases parametrams reiksmes

Parameters

<i>v</i>	Studento vardas
<i>p</i>	Studento pavarde
<i>n</i>	Studento namu darbu pazymiai
<i>e</i>	Studento egzamino pazymys

4.1.2.3 ~Stud()

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

Destruktorius, kuris isvalo visus duomenis.

4.1.2.4 Stud() [3/3]

```
Stud::Stud (  
    const Stud & kitas) [inline]
```

Kopijavimo konstruktorius, kuris sukuria kopija is kito studento objekto.

Parameters

<i>kitas</i>	Studentas, kurio duomenys bus nukopijuoti
--------------	---

4.1.3 Member Function Documentation**4.1.3.1 getEgz()**

```
int Stud::getEgz () const [inline]
```

4.1.3.2 getNd()

```
vector< int > Stud::getNd () const [inline]
```

4.1.3.3 getOnePaz()

```
int Stud::getOnePaz (  
    int i) const [inline]
```

4.1.3.4 getPavarde()

```
string Stud::getPavarde () const [inline]
```

4.1.3.5 getVardas()

```
string Stud::getVardas () const [inline]
```

4.1.3.6 operator=()

```
Stud & Stud::operator= (  
    const Stud & kitas) [inline]
```

Kopijavimo priskyrimo operatorius, kuris priskiria vieno studento duomenis kitam.

Parameters

<i>kitas</i>	Studentas, kurio duomenys bus priskirti kitam
--------------	---

Returns

Grazina nuoroda i dabartini objekta

4.1.3.7 setEgz()

```
void Stud::setEgz (
    const int & e) [inline]
```

4.1.3.8 setNd()

```
void Stud::setNd (
    const vector< int > & n) [inline]
```

4.1.3.9 setOnePaz()

```
void Stud::setOnePaz (
    int paz) [inline]
```

4.1.3.10 setPavarde()

```
void Stud::setPavarde (
    const string & p) [inline]
```

4.1.3.11 setVardas()

```
void Stud::setVardas (
    const string & v) [inline]
```

4.1.4 Friends And Related Symbol Documentation**4.1.4.1 operator<<**

```
ostream & operator<< (
    ostream & out,
    const Stud & stud) [friend]
```

Isvesties operatorius, kuris isveda studento duomenis.

Parameters

<i>out</i>	Išvesties srautas
<i>stud</i>	Studentas, kurio duomenys bus išvedami

Returns

Išvesties srautas

4.1.4.2 operator>>

```
istream & operator>> (
    istream & is,
    Stud & stud) [friend]
```

Išvesties operatorius, kuriuo įvedami studento duomenys.

Parameters

<i>is</i>	Išvesties srautas
<i>stud</i>	Studentas, kurio duomenys įvedami

Returns

Išvesties srautas

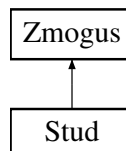
The documentation for this class was generated from the following file:

- C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/Stud.h

4.2 Zmogus Class Reference

```
#include <Stud.h>
```

Inheritance diagram for Zmogus:



Public Member Functions

- [Zmogus](#) ()
- [Zmogus](#) (const string &v, const string &p)
- virtual [~Zmogus](#) ()=0
- string [getVardas](#) () const
- string [getPavarde](#) () const
- string [setVardas](#) (const string &v)
- string [setPavarde](#) (const string &p)

Protected Attributes

- string [vardas](#)
- string [pavarde](#)

4.2.1 Detailed Description

Abstrakti klase [Zmogus](#), kuri apibrezia zmogaus objekto savybes Turi du parametrus - vardas ir pavarde, default ir parametrizuotas konstruktoriu, destruktoriu, getterius ir setterius.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Zmogus() [1/2]

```
Zmogus::Zmogus () [inline]
```

4.2.2.2 Zmogus() [2/2]

```
Zmogus::Zmogus (  
    const string & v,  
    const string & p) [inline]
```

4.2.2.3 ~Zmogus()

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

4.2.3 Member Function Documentation

4.2.3.1 getPavarde()

```
string Zmogus::getPavarde () const [inline]
```

4.2.3.2 getVardas()

```
string Zmogus::getVardas () const [inline]
```

4.2.3.3 setPavarde()

```
string Zmogus::setPavarde (  
    const string & p) [inline]
```


4.2.3.4 setVardas()

```
string Zmogus::setVardas (  
    const string & v) [inline]
```

4.2.4 Member Data Documentation

4.2.4.1 pavarde

```
string Zmogus::pavarde [protected]
```

4.2.4.2 vardas

```
string Zmogus::vardas [protected]
```

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

- C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/[Stud.h](#)

Chapter 5

File Documentation

5.1 C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/header.h File Reference

```
#include <iostream>
#include <iomanip>
#include <vector>
#include <algorithm>
#include <fstream>
#include <string>
#include <sstream>
#include <limits>
#include <chrono>
#include <cassert>
```

5.2 header.h

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

```
00001 #ifndef STUDENT_H
00002 #define STUDENT_H
00003
00004 #include <iostream>
00005 #include <iomanip>
00006 #include <vector>
00007 #include <algorithm>
00008 #include <fstream>
00009 #include <string>
00010 #include <sstream>
00011 #include <limits>
00012 #include <chrono>
00013 #include <cassert>
00014
00015 using namespace std;
00016
00017 #endif
```

5.3 C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/main.cpp File Reference

```
#include "header.h"
#include "Stud.h"
```

Functions

- int `main` ()

Funkcija, kuria yra vykdoma programa.

5.3.1 Function Documentation

5.3.1.1 `main()`

```
int main ()
```

Funkcija, kuria yra vykdoma programa.

1. Paklausia, ar norima testuoti konstruktorius ir operatorius
2. Paklausia, ar norima generuoti failus su atsitiktiniais duomenimis 3. Paklausia, ar norima iversti duomenis ranka, generuoti atsitiktinai ar skaityti is failo
3. Paklausia, ar norima galutini pazymi skaiciuoti pagal vidurki arba mediana
4. Paklausia, ar norima skirstyti studentus i saunuolius ir nevykelius
5. Paklausia, ar norima surusiuoti studentus pagal varda, pavarde arba galutini pazymi
6. Paklausia, ar norima duomenis isvesti i terminala arba i faila

Visos programos metu yra skaiciuojama sparta, kuri isspausdinama gale

5.4 C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/Stud.cpp File Reference

```
#include "Stud.h"
```

Functions

- double `ndVidurkis` (const `Stud` &student)
Funkcija, kuri suskaiciuoja studentu namu darbu vidurkis.
- double `ndMediana` (const `Stud` &student)
Funkcija, kuri suskaiciuoja studentu namu darbu mediana.
- double `galutinis` (const `Stud` &student, char pasirinkimas)
*Funkcija, kuri suskaiciuoja galutini rezultata, naudojantis formule $0.4 * ndPasirinkimas + 0.6 * egz.$*
- void `iverstiDuomenisRanka` (vector< `Stud` > &student, int ndSkaicius)
Funkcija, kuria ivedami studentu duomenys ranka - vardas, pavarde, namu darbu ir egzamino pazymiai.
- void `atsitiktiniaiPazymiai` (vector< `Stud` > &student, double ndSkaicius)
Funkcija, kuri atsitiktinai generuoja studento pazymius (varda pavarde reikia iversti ranka)
- void `nuskaitytiFaila` (vector< `Stud` > &student, string failoPav)
Funkcija, kuri nuskaityti faile esancius studentu duomenis.
- void `spausdinti` (vector< `Stud` > &student, char pasirinkimas)
Funkcija, kuri isspausdina studento varda, pavarde ir galutini vidurki (pagal vidurki arba mediana)

- void `isvestiFaila` (vector< `Stud` > student, char pasirinkimas, string failoPav)
Funkcija, kuri isveda duomenis i faila.
- void `generuotiFailus` (string failoPav, int studSk)
Funkcija, kuri generuoja failus su atsitiktiniais duomenimis.
- void `paskirtytiStud` (std::vector< `Stud` > &studentai, std::vector< `Stud` > &saunuoliai, std::vector< `Stud` > &nevykeliai, char pasirinkimas)
Funkcija, kuri paskirsto studentus i saunuolius ir nevykelius pagal ju pazymius.
- std::chrono::steady_clock::time_point `dabLaikas` ()
Funkcija, kuri parodo dabartini laika, naudoja chrono biblioteka.
- void `patikrinti` (bool a, const string &pav)
Funkcija, kuri patikrina ar testas pavyko ar nepavyko.
- void `test` ()
Funkcija, kuri demonstruoja Rule Of Three - kopijavimo konstruktoriu, kopijavimo priskyrimo konstruktoriu ir destruktoriu.

5.4.1 Function Documentation

5.4.1.1 atsitiktiniaiPazymiai()

```
void atsitiktiniaiPazymiai (
    vector< Stud > & student,
    double ndSkaicius)
```

Funkcija, kuri atsitiktinai generuoja studento pazymius (varda pavarde reikia ivesti ranka)

Parameters

<i>student</i>	Studentu vektorius, kuriu duomenys generuojami
<i>ndSkaicius</i>	Namu darbu skaicius, kuri reikia ivesti

5.4.1.2 dabLaikas()

```
std::chrono::steady_clock::time_point dabLaikas ()
```

Funkcija, kuri parodo dabartini laika, naudoja chrono biblioteka.

5.4.1.3 galutinis()

```
double galutinis (
    const Stud & student,
    char pasirinkimas)
```

Funkcija, kuri suskaiciuoja galutini rezultata, naudojantis formule $0.4 * ndPasirinkimas + 0.6 + egz$.

Parameters

<i>student</i>	Studentu vektorius, kuriu galutinis pazymys bus skaiciuojamas
<i>pasirinkimas</i>	Pasirinkimas (vidurkis arba mediana), pagal kuri bus skaiciuojamas galutinis pazymis

Returns

Grazinamas studento galutinis pazymys

5.4.1.4 generuotiFailus()

```
void generuotiFailus (
    string failoPav,
    int studSk)
```

Funkcija, kuri generuoja failus su atsitiktiniais duomenimis.

Parameters

<i>failoPav</i>	Kaip norime pavadinti generuojama faila
<i>studSk</i>	Skaicius studentu, kuriu duomenis norime atsitiktinai sugeneruoti

5.4.1.5 isvestiFaila()

```
void isvestiFaila (
    vector< Stud > student,
    char pasirinkimas,
    string failoPav)
```

Funkcija, kuri isveda duomenis i faila.

Parameters

<i>student</i>	Studentu vektorius, kurio duomenys bus isvedami i faila
<i>pasirinkimas</i>	Galutinio pazymio skaiciavimo pasirinkimas (vidurkis arba mediana)
<i>failoPav</i>	Kaip norime pavadinti isvedama faila

5.4.1.6 ivestiDuomenisRanka()

```
void ivestiDuomenisRanka (
    vector< Stud > & student,
    int ndSkaicius)
```

Funkcija, kuria ivedami studentu duomenys ranka - vardas, pavarde, namu darbu ir egzamino pazymiai.

Parameters

<i>student</i>	Studentu vektorius, kuriu duomenys ivedami
<i>ndSkaicius</i>	Namu darbu skaicius, kuri reikia ivesti

5.4.1.7 ndMediana()

```
double ndMediana (
    const Stud & student)
```

Funkcija, kuri suskaiciuoja studentu namu darbu mediana.

Parameters

<i>student</i>	Studentu vektorius, kuriu namu darbu mediana skaiciuojama
----------------	---

Returns

Grazinama namu darbu mediana

5.4.1.8 ndVidurkis()

```
double ndVidurkis (  
    const Stud & student)
```

Funkcija, kuri suskaiciuoja studentu namu darbu vidurkis.

Parameters

<i>student</i>	Studentu vektorius, kuriu namu darbu vidurkis skaiciuojamas
----------------	---

Returns

Grazinamas namu darbu vidurkis

5.4.1.9 nuskaitytiFaila()

```
void nuskaitytiFaila (  
    vector< Stud > & student,  
    string failoPav)
```

Funkcija, kuri nuskaito faile esancius studentu duomenis.

Parameters

<i>student</i>	Studentu vektorius, kuriu duomenys bus nuskaitomi
<i>failoPav</i>	Failo, kuri nuskaitome pavadinimas

5.4.1.10 paskirtytiStud()

```
void paskirtytiStud (  
    std::vector< Stud > & studentai,  
    std::vector< Stud > & saunuoliai,  
    std::vector< Stud > & nevykeliai,  
    char pasirinkimas)
```

Funkcija, kuri paskirsto studentus i saunuolius ir nevykelius pagal ju pazymius.

Parameters

<i>studentai</i>	Studentu vektorius, kur laikomi visi studentai
<i>saunuoliai</i>	Saunuoliu vektorius, kuriame bus laikomi studentai su galutiniu pazymiu ≥ 5
<i>nevykeliai</i>	Nevykeliu vektorius, kuriame bus laikomi studentai su galutiniu pazymiu ≤ 5
<i>pasirinkimas</i>	Galutinio pazymio skaiciavimo pasirinkimas (vidurkis arba mediana)

5.4.1.11 patikrinti()

```
void patikrinti (
    bool a,
    const string & pav)
```

Funkcija, kuri patikrina ar testas pavyko ar nepavyko.

Parameters

<i>a</i>	Logine reiksme, kuri parodo ar testas sekmingas (true) ar ne (false)
<i>pav</i>	Tekstas, kuris isvedamas su testo rezultatu

5.4.1.12 spausdinti()

```
void spausdinti (
    vector< Stud > & student,
    char pasirinkimas)
```

Funkcija, kuri isspausdina studento varda, pavarde ir galutini vidurki (pagal vidurki arba mediana)

Parameters

<i>student</i>	Studentu vektorius, kuriu duomenys bus isspausdinami
<i>pasirinkimas</i>	Galutinio pazymio skaiciavimo pasirinkimas (vidurkis arba mediana)

5.4.1.13 test()

```
void test ()
```

Funkcija, kuri demonstruoja Rule Of Three - kopijavimo konstruktoriu, kopijavimo priskyrimo konstruktoriu ir destruktoriu.

5.5 C:/Users/abrom/OneDrive/Stalinis kompiuteris/obj-2.0/Stud.h File Reference

```
#include "header.h"
```


Classes

- class [Zmogus](#)
- class [Stud](#)

Functions

- double [ndVidurkis](#) (const [Stud](#) &student)
Funkcija, kuri suskaiciuoja studentu namu darbu vidurkis.
- double [ndMediana](#) (const [Stud](#) &student)
Funkcija, kuri suskaiciuoja studentu namu darbu mediana.
- double [galutinis](#) (const [Stud](#) &student, char pasirinkimas)
*Funkcija, kuri suskaiciuoja galutini rezultata, naudojantis formule $0.4 * ndPasirinkimas + 0.6 * egz.$*
- void [investiDuomenisRanka](#) (vector< [Stud](#) > &student, int ndSkaicius)
Funkcija, kuria ivedami studentu duomenys ranka - vardas, pavarde, namu darbu ir egzamino pazymiai.
- void [atsitiktiniaiPazymiai](#) (vector< [Stud](#) > &student, double ndSkaicius)
Funkcija, kuri atsitiktinai generuoja studento pazymius (varda pavarde reikia ivesti ranka)
- void [nuskaitytiFaila](#) (vector< [Stud](#) > &student, string failoPav)
Funkcija, kuri nuskaity faila esancius studentu duomenis.
- void [spausdinti](#) (vector< [Stud](#) > &studentai, char pasirinkimas)
Funkcija, kuri isspausdina studento varda, pavarde ir galutini vidurki (pagal vidurki arba mediana)
- void [investiFaila](#) (vector< [Stud](#) > student, char pasirinkimas, string failoPav)
Funkcija, kuri isveda duomenis i faila.
- void [generuotiFailus](#) (string failoPav, int studSk)
Funkcija, kuri generuoja failus su atsitiktiniais duomenimis.
- void [paskirtytiStud](#) (vector< [Stud](#) > &studentai, vector< [Stud](#) > &saunuoliai, vector< [Stud](#) > &nevykeliai, char pasirinkimas)
- std::chrono::steady_clock::time_point [dabLaikas](#) ()
Funkcija, kuri parodo dabartini laika, naudoja chrono biblioteka.
- void [patikrinti](#) (bool a, const string &pav)
Funkcija, kuri patikrina ar testas pavyko ar nepavyko.
- void [test](#) ()
Funkcija, kuri demonstruoja Rule Of Three - kopijavimo konstruktoriu, kopijavimo priskyrimo konstruktoriu ir destruktoriu.

5.5.1 Function Documentation

5.5.1.1 atsitiktiniaiPazymiai()

```
void atsitiktiniaiPazymiai (
    vector< Stud > & student,
    double ndSkaicius)
```

Funkcija, kuri atsitiktinai generuoja studento pazymius (varda pavarde reikia ivesti ranka)

Parameters

<i>student</i>	Studentu vektorius, kuriu duomenys generuojami
<i>ndSkaicius</i>	Namu darbu skaicius, kuri reikia ivesti

5.5.1.2 dabLaikas()

```
std::chrono::steady_clock::time_point dabLaikas ()
```

Funkcija, kuri parodo dabartinį laiką, naudoja chrono biblioteką.

5.5.1.3 galutinis()

```
double galutinis (
    const Stud & student,
    char pasirinkimas)
```

Funkcija, kuri suskaiciuoja galutinius rezultatus, naudojantis formule $0.4 * ndPasirinkimas + 0.6 * egz$.

Parameters

<i>student</i>	Studentų vektorius, kuriu galutinis pažymys bus skaičiuojamas
<i>pasirinkimas</i>	Pasirinkimas (vidurkis arba mediana), pagal kuri bus skaičiuojamas galutinis pažymys

Returns

Grazinamas studento galutinis pažymys

5.5.1.4 generuotiFailus()

```
void generuotiFailus (
    string failoPav,
    int studSk)
```

Funkcija, kuri generuoja failus su atsitiktiniais duomenimis.

Parameters

<i>failoPav</i>	Kaip norime pavadinti generuojama failą
<i>studSk</i>	Skaicius studentų, kurių duomenis norime atsitiktinai sugeneruoti

5.5.1.5 isvestiFaila()

```
void isvestiFaila (
    vector< Stud > student,
    char pasirinkimas,
    string failoPav)
```

Funkcija, kuri išveda duomenis į failą.

Parameters

<i>student</i>	Studentų vektorius, kurio duomenys bus išvedami į failą
<i>pasirinkimas</i>	Galutinio pažymio skaičiavimo pasirinkimas (vidurkis arba mediana)
<i>failoPav</i>	Kaip norime pavadinti išvedama failą

5.5.1.6 iverstiDuomenisRanka()

```
void iverstiDuomenisRanka (
    vector< Stud > & student,
    int ndSkaicius)
```

Funkcija, kuria ivedami studentu duomenys ranka - vardas, pavarde, namu darbu ir egzamino pazymiai.

Parameters

<i>student</i>	Studentu vektorius, kuriu duomenys ivedami
<i>ndSkaicius</i>	Namu darbu skaicius, kuri reikia iversti

5.5.1.7 ndMediana()

```
double ndMediana (
    const Stud & student)
```

Funkcija, kuri suskaiciuoja studentu namu darbu mediana.

Parameters

<i>student</i>	Studentu vektorius, kuriu namu darbu mediana skaiciuojama
----------------	---

Returns

Grazinama namu darbu mediana

5.5.1.8 ndVidurkis()

```
double ndVidurkis (
    const Stud & student)
```

Funkcija, kuri suskaiciuoja studentu namu darbu vidurkis.

Parameters

<i>student</i>	Studentu vektorius, kuriu namu darbu vidurkis skaiciuojamas
----------------	---

Returns

Grazinamas namu darbu vidurkis

5.5.1.9 nuskaitytiFaila()

```
void nuskaitytiFaila (
    vector< Stud > & student,
    string failoPav)
```

Funkcija, kuri nuskaity faile esancius studentu duomenis.

Parameters

<i>student</i>	Studentu vektorius, kuriu duomenys bus nuskaitomi
<i>failoPav</i>	Failo, kuri nuskaitome pavadinimas

5.5.1.10 paskirtytiStud()

```
void paskirtytiStud (
    vector< Stud > & studentai,
    vector< Stud > & saunuoliai,
    vector< Stud > & nevykeliai,
    char pasirinkimas)
```

5.5.1.11 patikrinti()

```
void patikrinti (
    bool a,
    const string & pav)
```

Funkcija, kuri patikrina ar testas pavyko ar nepavyko.

Parameters

<i>a</i>	Logine reiksme, kuri parodo ar testas sekmingas (true) ar ne (false)
<i>pav</i>	Tekstas, kuris isvedamas su testo rezultatu

5.5.1.12 spausdinti()

```
void spausdinti (
    vector< Stud > & student,
    char pasirinkimas)
```

Funkcija, kuri isspausdina studento varda, pavarde ir galutini vidurki (pagal vidurki arba mediana)

Parameters

<i>student</i>	Studentu vektorius, kuriu duomenys bus isspausdinami
<i>pasirinkimas</i>	Galutinio pazymio skaiciavimo pasirinkimas (vidurkis arba mediana)

5.5.1.13 test()

```
void test ()
```

Funkcija, kuri demonstruoja Rule Of Three - kopijavimo konstruktoriu, kopijavimo priskyrimo konstruktoriu ir destruktoriu.

5.6 Stud.h

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

```

00001 #include "header.h"
00002
00008 class Zmogus {
00009     protected:
00010         string vardas;
00011         string pavarde;
00012
00013     public:
00014         //konstruktoriai
00015         Zmogus() : vardas(""), pavarde("") {}
00016         Zmogus(const string& v, const string&p) : vardas(v), pavarde(p) {}
00017
00018         //destruktorius
00019         virtual ~Zmogus() = 0;
00020
00021         //getteriai, setteriai
00022         string getVardas() const { return vardas; }
00023         string getPavarde() const { return pavarde; }
00024
00025         string setVardas(const string& v) { return vardas = v; }
00026         string setPavarde(const string& p) { return pavarde = p; }
00027 };
00028
00029 inline Zmogus::~Zmogus() {}
00030
00037 class Stud : public Zmogus {
00038     private:
00039         vector<int> nd;
00040         int egz;
00041
00042     public:
00043
00048         Stud() : Zmogus("", ""), nd{}, egz(0) {}
00049
00058         Stud(const string& v, const string& p, const vector<int> n, int e) : Zmogus(v, p), nd(n), egz(e)
00059     {}
00064
00065     ~Stud() {
00066         vardas.clear();
00067         pavarde.clear();
00068         nd.clear();
00069         egz = 0;
00070
00071         // cout << "destruktorius ivykdytas" << endl;
00072     }
00073
00078     Stud(const Stud &kitas) : Zmogus(kitas.vardas, kitas.pavarde), nd(kitas.nd), egz(kitas.egz) {}
00079
00086     Stud& operator = (const Stud &kitas) {
00087         if (this != &kitas) {
00088             vardas = kitas.vardas;
00089             pavarde = kitas.pavarde;
00090             nd = kitas.nd;
00091             egz = kitas.egz;
00092         }
00093         return *this;
00094     }
00095
00103     friend istream& operator >> (istream &is, Stud &stud) {
00104         cout << "Iveskite studento vardą pavarde: ";
00105         is >> stud.vardas >> stud.pavarde;
00106
00107         cout << "Iveskite nd skaičių: ";
00108         int ndSkaicius;
00109         is >> ndSkaicius;
00110         stud.nd.clear();
00111
00112         cout << "Iveskite " << ndSkaicius << " namų darbų pažymius: ";
00113         for (int i = 0; i < ndSkaicius; ++i) {
00114             int paz;
00115             is >> paz;
00116             stud.nd.push_back(paz);
00117         }
00118
00119         cout << "Iveskite egzaminų pažymius: ";
00120         is >> stud.egz;
00121
00122         return is;
00123     }
00124
00125     friend ostream& operator << (ostream &out, const Stud &stud) {
00133

```

```

00134         out << "Vardas: " << stud.getVardas() << endl;
00135         out << "Pavarde: " << stud.getPavarde() << endl;
00136         out << "Egzamino rezultatas: " << stud.getEgz() << endl;
00137         out << "Pazymiai: ";
00138         vector<int> paz = stud.getNd();
00139         for (int i = 0; i < paz.size(); i++){
00140             out << paz.at(i) << " ";
00141         }
00142         out << endl;
00143         return out;
00144     }
00145
00146     //getteriaim, setteriai
00147
00148     string getVardas() const {
00149         return vardas;
00150     }
00151
00152     void setVardas (const string &v) {
00153         vardas = v;
00154     }
00155
00156     string getPavarde() const {
00157         return pavarde;
00158     }
00159
00160     void setPavarde(const string &p) {
00161         pavarde = p;
00162     }
00163
00164     vector<int> getNd() const {
00165         return nd;
00166     }
00167
00168     void setNd(const vector<int> &n) {
00169         nd = n;
00170     }
00171
00172     int getOnePaz(int i) const {
00173         return nd.at(i);
00174     }
00175
00176     void setOnePaz(int paz) {
00177         nd.push_back(paz);
00178     }
00179
00180     int getEgz() const {
00181         return egz;
00182     }
00183
00184     void setEgz(const int &e) {
00185         egz = e;
00186     }
00187 };
00188
00189 double ndVidurkis(const Stud &student);
00190 double ndMediana(const Stud &student);
00191 double galutinis(const Stud &student, char pasirinkimas);
00192 void iverstiDuomenisRanka(vector<Stud> &student, int ndSkaicius);
00193 void atsitiktiniaiPazymiai(vector<Stud> &student, double ndSkaicius);
00194 void nuskaitytiFaila(vector<Stud> &student, string failoPav);
00195 void spausdinti(vector<Stud> &studentai, char pasirinkimas);
00196 void isvestiFaila(vector<Stud> student, char pasirinkimas, string failoPav);
00197 void generuotiFailus(string failoPav, int studSk);
00198 void paskirtytiStud(vector<Stud> &studentai, vector<Stud> &saunuoliai, vector<Stud> &nevykeliai, char
    pasirinkimas);
00199 std::chrono::steady_clock::time_point dabLaikas();
00200 void patikrinti(bool a, const string& pav);
00201 void test();

```

Index

~Stud
 Stud, [9](#)

~Zmogus
 Zmogus, [12](#)

atsitiktiniaiPazymiai
 Stud.cpp, [17](#)
 Stud.h, [21](#)

C:/Users/abrom/OneDrive/Stalinis
 2.0/header.h, [15](#)

C:/Users/abrom/OneDrive/Stalinis
 2.0/main.cpp, [15](#)

C:/Users/abrom/OneDrive/Stalinis
 2.0/Stud.cpp, [16](#)

C:/Users/abrom/OneDrive/Stalinis
 2.0/Stud.h, [20](#), [25](#)

dabLaikas
 Stud.cpp, [17](#)
 Stud.h, [21](#)

galutinis
 Stud.cpp, [17](#)
 Stud.h, [22](#)

generuotiFailus
 Stud.cpp, [17](#)
 Stud.h, [22](#)

getEgz
 Stud, [9](#)

getNd
 Stud, [9](#)

getOnePaz
 Stud, [9](#)

getPavarde
 Stud, [9](#)
 Zmogus, [12](#)

getVardas
 Stud, [9](#)
 Zmogus, [12](#)

isvestiFaila
 Stud.cpp, [18](#)
 Stud.h, [22](#)

ivestiDuomenisRanka
 Stud.cpp, [18](#)
 Stud.h, [22](#)

main
 main.cpp, [16](#)

main.cpp

main, [16](#)

ndMediana
 Stud.cpp, [18](#)
 Stud.h, [23](#)

ndVidurkis
 Stud.cpp, [19](#)
 Stud.h, [23](#)

nuskaitytiFaila
 Stud.cpp, [19](#)
 Stud.h, [23](#)

operator<<
 Stud, [10](#)

operator>>
 Stud, [11](#)

operator=
 Stud, [9](#)

paskirtytiStud
 Stud.cpp, [19](#)
 Stud.h, [24](#)

patikrinti
 Stud.cpp, [20](#)
 Stud.h, [24](#)

pavarde
 Zmogus, [13](#)

setEgz
 Stud, [10](#)

setNd
 Stud, [10](#)

setOnePaz
 Stud, [10](#)

setPavarde
 Stud, [10](#)
 Zmogus, [12](#)

setVardas
 Stud, [10](#)
 Zmogus, [12](#)

spausdinti
 Stud.cpp, [20](#)
 Stud.h, [24](#)

Stud, [7](#)
 ~Stud, [9](#)
 getEgz, [9](#)
 getNd, [9](#)
 getOnePaz, [9](#)
 getPavarde, [9](#)
 getVardas, [9](#)

- operator<<, 10
- operator>>, 11
- operator=, 9
- setEgz, 10
- setNd, 10
- setOnePaz, 10
- setPavarde, 10
- setVardas, 10
- Stud, 8, 9
- Stud.cpp
 - atsitiktiniaiPazymiai, 17
 - dabLaikas, 17
 - galutinis, 17
 - generuotiFailus, 17
 - isvestiFaila, 18
 - investiDuomenisRanka, 18
 - ndMediana, 18
 - ndVidurkis, 19
 - nuskaitytiFaila, 19
 - paskirtytiStud, 19
 - patikrinti, 20
 - spausdinti, 20
 - test, 20
- Stud.h
 - atsitiktiniaiPazymiai, 21
 - dabLaikas, 21
 - galutinis, 22
 - generuotiFailus, 22
 - isvestiFaila, 22
 - investiDuomenisRanka, 22
 - ndMediana, 23
 - ndVidurkis, 23
 - nuskaitytiFaila, 23
 - paskirtytiStud, 24
 - patikrinti, 24
 - spausdinti, 24
 - test, 24
- test
 - Stud.cpp, 20
 - Stud.h, 24
- vardas
 - Zmogus, 13
- Zmogus, 11
 - ~Zmogus, 12
 - getPavarde, 12
 - getVardas, 12
 - pavarde, 13
 - setPavarde, 12
 - setVardas, 12
 - vardas, 13
 - Zmogus, 12