naujasoop

Generated by Doxygen 1.13.2

1 NAUDOJIMOSI INSTRUKCIJA: 1

1.1 Naujienos . . . 2
1.2 Ką jie daro? . . . 2
1.3 V1.5 . . . 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 Stud Struct Reference . . . 9
5.1.1 Constructor & Destructor Documentation . . . 10
5.1.1.1 Stud() [1/5] . . . 10
5.1.1.2 ~Stud() . . . 10
5.1.1.3 Stud() [2/5] . . . 11
5.1.1.4 Stud() [3/5] . . . 11
5.1.1.5 Stud() [4/5] . . . 11
5.1.1.6 Stud() [5/5] . . . 11
5.1.2 Member Function Documentation . . . 11
5.1.2.1 addPaz() . . . 11
5.1.2.2 clearPaz() . . . 11
5.1.2.3 galutinis_mediana() . . . 11
5.1.2.4 galutinis_vidurkis() . . . 11
5.1.2.5 getEgzaminas() . . . 11
5.1.2.6 getGalutinis() . . . 11
5.1.2.7 getMediana() . . . 11
5.1.2.8 getNdvid() . . . 11
5.1.2.9 getPazymiai() . . . 12
5.1.2.10 operator=() [1/2] . . . 12
5.1.2.11 operator=() [2/2] . . . 12
5.1.2.12 paskaiciuoti_gal() . . . 12
5.1.2.13 paskaiciuoti_vid_ir_med() . . . 12
5.1.2.14 print() . . . 12
5.1.2.15 read() . . . 12
5.1.2.16 readStudent() . . . 12
5.1.2.17 setEgrez() . . . 12
5.1.2.18 setGal() . . . 12
5.1.2.19 setMed() . . . 12
5.1.2.20 setNdvid() . . . 13

5.1.2.21 setPav() . . . 13
5.1.2.22 setVar() . . . 13
5.1.3 Friends And Related Symbol Documentation . . . 13
5.1.3.1 operator<< . . . 13
5.1.3.2 operator>> . . . 13
5.1.4 Member Data Documentation . . . 13
5.1.4.1 egrez . . . 13
5.1.4.2 egrez_ . . . 13
5.1.4.3 gal . . . 13
5.1.4.4 gal_ . . . 13
5.1.4.5 med . . . 13
5.1.4.6 med_ . . . 13
5.1.4.7 ndvid . . . 13
5.1.4.8 ndvid_ . . . 14
5.1.4.9 P . . . 14
5.1.4.10 pav . . . 14
5.1.4.11 paz . . . 14
5.1.4.12 paz_ . . . 14
5.1.4.13 pazkiek . . . 14
5.1.4.14 var . . . 14
5.2 Zmogus Class Reference . . . 14
5.2.1 Constructor & Destructor Documentation . . . 15
5.2.1.1 Zmogus() [1/2] . . . 15
5.2.1.2 Zmogus() [2/2] . . . 15
5.2.1.3 ~Zmogus() . . . 15
5.2.2 Member Function Documentation . . . 15
5.2.2.1 getPavarde() . . . 15
5.2.2.2 getVardas() . . . 15
5.2.2.3 print() . . . 15
5.2.2.4 read() . . . 15
5.2.3 Member Data Documentation . . . 15
5.2.3.1 pav_ . . . 15
5.2.3.2 var_ . . . 15

6 File Documentation **17**

6.1 deque/dequefunk.cpp File Reference . . . 17
6.1.1 Function Documentation . . . 18
6.1.1.1 atrinkimas1() . . . 18
6.1.1.2 atrinkimas2() . . . 18
6.1.1.3 atrinkimas3() . . . 18
6.1.1.4 failasegzistuoja() . . . 18
6.1.1.5 failogen() . . . 18

6.1.1.6 gautteisinga() . . . 18
6.1.1.7 isvedimas() . . . 18
6.1.1.8 operator<<() . . . 18
6.1.1.9 operator>>() . . . 19
6.1.1.10 processBatch() . . . 19
6.1.1.11 rng() [1/3] . . . 19
6.1.1.12 rng() [2/3] . . . 19
6.1.1.13 rng() [3/3] . . . 19
6.1.1.14 skaitymas() . . . 19
6.1.1.15 sortGal() . . . 19
6.1.1.16 sortMed() . . . 19
6.1.1.17 sortPav() . . . 19
6.1.1.18 sortVardu() . . . 20
6.1.2 Variable Documentation . . . 20
6.1.2.1 g_printMode . . . 20
6.2 deque/dequeh.h File Reference . . . 20
6.2.1 Function Documentation . . . 21
6.2.1.1 atrinkimas1() . . . 21
6.2.1.2 atrinkimas2() . . . 21
6.2.1.3 atrinkimas3() . . . 21
6.2.1.4 failogen() . . . 21
6.2.1.5 galvid() . . . 21
6.2.1.6 gautteisinga() . . . 22
6.2.1.7 isvedimas() . . . 22
6.2.1.8 mediana() . . . 22
6.2.1.9 ndvid() . . . 22
6.2.1.10 operator<<() . . . 22
6.2.1.11 operator>>() . . . 22
6.2.1.12 processBatch() . . . 22
6.2.1.13 rng() [1/3] . . . 22
6.2.1.14 rng() [2/3] . . . 22
6.2.1.15 rng() [3/3] . . . 23
6.2.1.16 skaitymas() . . . 23
6.2.1.17 sortGal() . . . 23
6.2.1.18 sortMed() . . . 23
6.2.1.19 sortPav() . . . 23
6.2.1.20 sortVardu() . . . 23
6.2.2 Variable Documentation . . . 23
6.2.2.1 pavardes . . . 23
6.2.2.2 vardai . . . 23
6.3 dequeh.h . . . 23
6.4 deque/dequemain.cpp File Reference . . . 25

6.4.1 Function Documentation . . . 26
6.4.1.1 main() . . . 26
6.5 deque/test.cpp File Reference . . . 26
6.5.1 Function Documentation . . . 26
6.5.1.1 main() . . . 26
6.5.1.2 test_io() . . . 26
6.5.1.3 test_rule_of_five() . . . 26
6.6 list/listfunk.cpp File Reference . . . 26
6.6.1 Function Documentation . . . 27
6.6.1.1 atrinkimas1() . . . 27
6.6.1.2 atrinkimas2() . . . 27
6.6.1.3 atrinkimas3() . . . 27
6.6.1.4 failasegzistuoja() . . . 27
6.6.1.5 failogen() . . . 28
6.6.1.6 galvid() . . . 28
6.6.1.7 gautteisinga() . . . 28
6.6.1.8 isvedimas() . . . 28
6.6.1.9 mediana() . . . 28
6.6.1.10 ndvid() . . . 28
6.6.1.11 processBatch() . . . 28
6.6.1.12 rng() [1/3] . . . 28
6.6.1.13 rng() [2/3] . . . 28
6.6.1.14 rng() [3/3] . . . 29
6.6.1.15 skaitymas() . . . 29
6.6.1.16 sortGal() . . . 29
6.6.1.17 sortMed() . . . 29
6.6.1.18 sortPav() . . . 29
6.6.1.19 sortVardu() . . . 29
6.7 list/listh.h File Reference . . . 29
6.7.1 Function Documentation . . . 30
6.7.1.1 atrinkimas1() . . . 30
6.7.1.2 atrinkimas2() . . . 30
6.7.1.3 atrinkimas3() . . . 31
6.7.1.4 failogen() . . . 31
6.7.1.5 galvid() . . . 31
6.7.1.6 gautteisinga() . . . 31
6.7.1.7 isvedimas() . . . 31
6.7.1.8 mediana() . . . 31
6.7.1.9 ndvid() . . . 31
6.7.1.10 processBatch() . . . 31
6.7.1.11 rng() [1/3] . . . 31
6.7.1.12 rng() [2/3] . . . 32

6.7.1.13 rng() [3/3] . . . 32
6.7.1.14 skaitymas() . . . 32
6.7.1.15 sortGal() . . . 32
6.7.1.16 sortMed() . . . 32
6.7.1.17 sortPav() . . . 32
6.7.1.18 sortVardu() . . . 32
6.7.2 Variable Documentation . . . 32
6.7.2.1 pavardes . . . 32
6.7.2.2 vardai . . . 32
6.8 listh.h . . . 32
6.9 list/listmain.cpp File Reference . . . 34
6.9.1 Function Documentation . . . 34
6.9.1.1 main() . . . 34
6.10 README.md File Reference . . . 34
6.11 vector/vectorfunk.cpp File Reference . . . 34
6.11.1 Function Documentation . . . 35
6.11.1.1 atrinkimas1() . . . 35
6.11.1.2 atrinkimas2() . . . 35
6.11.1.3 atrinkimas3() . . . 35
6.11.1.4 failasegzistuoja() . . . 35
6.11.1.5 failogen() . . . 36
6.11.1.6 galvid() . . . 36
6.11.1.7 gautteisinga() . . . 36
6.11.1.8 isvedimas() . . . 36
6.11.1.9 mediana() . . . 36
6.11.1.10 ndvid() . . . 36
6.11.1.11 processBatch() . . . 36
6.11.1.12 rng() [1/3] . . . 36
6.11.1.13 rng() [2/3] . . . 36
6.11.1.14 rng() [3/3] . . . 37
6.11.1.15 skaitymas() . . . 37
6.11.1.16 sortGal() . . . 37
6.11.1.17 sortMed() . . . 37
6.11.1.18 sortPav() . . . 37
6.11.1.19 sortVardu() . . . 37
6.12 vector/vectorh.h File Reference . . . 37
6.12.1 Function Documentation . . . 38
6.12.1.1 atrinkimas1() . . . 38
6.12.1.2 atrinkimas2() . . . 38
6.12.1.3 atrinkimas3() . . . 39
6.12.1.4 failogen() . . . 39
6.12.1.5 galvid() . . . 39

6.12.1.6 gautteisinga() . . . 39
6.12.1.7 isvedimas() . . . 39
6.12.1.8 mediana() . . . 39
6.12.1.9 ndvid() . . . 39
6.12.1.10 processBatch() . . . 39
6.12.1.11 rng() [1/3] . . . 39
6.12.1.12 rng() [2/3] . . . 40
6.12.1.13 rng() [3/3] . . . 40
6.12.1.14 skaitymas() . . . 40
6.12.1.15 sortGal() . . . 40
6.12.1.16 sortMed() . . . 40
6.12.1.17 sortPav() . . . 40
6.12.1.18 sortVardu() . . . 40
6.12.2 Variable Documentation . . . 40
6.12.2.1 pavardes . . . 40
6.12.2.2 vardai . . . 40
6.13 vectorh.h . . . 40
6.14 vector/vectormain.cpp File Reference . . . 42
6.14.1 Function Documentation . . . 42
6.14.1.1 main() . . . 42

Chapter 1

NAUDOJIMOSI INSTRUKCIJA:

Ačiū, kad parsisiuntėte šią programą! Deja, nesu baisiai protingas, todėl ir rašau naudojimosi instrukciją, kadangi paleidimas nėra vien paprasto .exe failo paspaudimas.

Pradedant nuo pradžių, turite turėti veikiantį GCC, kurio versija palaiko C++17 ar naujesnes versijas. Antra, turite turėti veikiantį MinGW, kuris reikalingas programos .exe failo sukūrimui, kuriam reikia mingw32-make komandos. Tam pridedu šią nuorodą: https://nerdyelectronics.com/install-mingw-on-windows-for-make/

Reikalingos komandos:
mingw32-make vector - sukuria .exe versijai su vektoriaus tipo konteineriu
mingw32-make class - sukuria .exe versijai su deque tipo konteineriu, bet naudojant klasę.
mingw32-make list - sukuria .exe versijai su list tipo konteineriu
mingw32-make clean - ištrina bin failą, kuriame laikomi sukurti jūsų .exe failai bei sugeneruoti duomenų failai
mingw32-make -B - rebuildina visų konteinerių tipų .exe failus

Parsisiuntę repo, turite du variantus:
Visual Studio

- Atsidaryti .sln failą, kuris atidarys visą projektą per Visual Studio, per kurį galėsite atsidaryti terminalą (CTRL+~) ir suvesti atitinkamas komandas, surašytas viršuje.

CMD

- Atsidaryti command line, (ieškoti cmd paspaudus Windows ikoną), nuvesti path iki šitos repo direktorijos naudojant cd komandą, pvz: cd C:\Users\andri\OneDrive\Desktop\oop naujas\bin ir tada vesti atitinkamas komandas pagal Jūsų poreikį.

Tai atlikus, atsiras bin folder'is, kuriame rasite .exe failus. Paleidus programą, visi pasirinkimai bus aiškiai jums duoti. Gero naudojimo!

TESTAVIMAS

- Kompiuterio, naudoto testavimui specifikacijos: CPU: Ryzen 7 5800U RAM: 32GB DDR4 3200MHz SSD: 480GB, ne M.2

v1.1 benchmark'as (O3 optimizacija)

-

Studentų kiekis	1000	10000	100000	1000000	10000000
Skaitymo laikas	0.003	0.02	0.20	1.62	14.45
Rūšiavimo laikas	0.0004	0.005	0.06	0.80	14.45
Atskyrimo i vektorius laikas	0.0003	0.002	0.02	0.21	2.78
Irašymo i nerdus laikas	0.007	0.02	0.17	1.72	18.21
Irašymo i galiorką laikas	0.005	0.01	0.12	1.19	12.29
Visas veikimo laikas	0.02	0.06	0.56	5.54	60.00

1.1 Naujienos

v1.2 buvo pritaikyta klasei penkių metodų taisyklė - t.y kopijavimo ir kėlimo konstruktoriai bei assign'eriai ir destruktorius. Taip pat pridėti operatorių išvesties ir įvesties būdai

1.2 Ką jie daro?

Šios taisyklės pagerina optimizaciją ir tuo pat programos veikimą, tiksliai parodantys klasei, kaip ką apdoroti, kelti, kopijuoti, bei naikinti.
Su išvesties ir įvesties metodais supaprastintas įvedimas ir išvedimas ranka arba atsitiktinai generuojant, t.y - viskas, kas nesusiję su darbu su failu. Tuose operatoriuose dabar perkelta faktiškai viskas, kas buvo kitose dalyse kodo ir susiję su rankiniu įvedimu bei išvedimu. (Šios eilutės užėmė daug vietos main.cpp faile, o dabar yra atskirai ir tvarkingai laikomos operatoriuje funkcijų .cpp faile, kur jas daug lengviau iššaukti.)
Su išvedimu tas pats - paprasčiau suprast kodą, lengviau iššaukt.
Prieš -
Po -
Dabar cout$<<$n ne tik paprastai išveda į ekraną n, o aktyvuoja viską, kas yra išvedimo operatoriuje -

1.3 V1.5

Versijai 1.5 pritaikyta abstrakti klasė Zmogus, kuri bus naudinga ateityje. Kol kas, tai pakeitė tik studento klasės realizaciją - ji dabar yra išvestinė klasė Zmogus abstrakčiąjai (visi studentai žmonės, lygtais..)
Visas veikimas bus identiškas į V1.2.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Zmogus 14
Stud 9

Chapter 3

Class Index

3.1 Class List

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

Stud 9
Zmogus 14

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

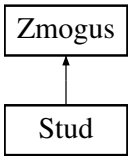
deque/dequefunk.cpp . . . 17
deque/dequeh.h . . . 20
deque/dequemain.cpp . . . 25
deque/test.cpp . . . 26
list/listfunk.cpp . . . 26
list/listh.h . . . 29
list/listmain.cpp . . . 34
vector/vectorfunk.cpp . . . 34
vector/vectorh.h . . . 37
vector/vectormain.cpp . . . 42

Chapter 5

Class Documentation

5.1 Stud Struct Reference

`#include <dequeh.h>`
Inheritance diagram for Stud:



Public Member Functions

- Stud ()
- ~Stud () override=default
- Stud (const Stud &other)
- Stud & operator= (const Stud &)

  kopijavimo konstruktorius
- Stud (Stud &&) noexcept

  kopijavimo assignment
- Stud & operator= (Stud &&) noexcept

  kelimo konstruktorius
- Stud (const string &vardas, const string &pavarde, const vector< int > &pazymiai, const int egzaminas)

  kelimo assignment
- Stud (istream &is)
- void setVar (const string &v)

  setteriai
- void setPav (const string &p)
- void addPaz (int p)
- void setEgrez (int e)
- void setNdvid (double v)
- void setGal (double g)
- void setMed (double m)
- void clearPaz ()
- vector< int > & getPazymiai ()

  getteriai
- int getEgzaminas () const
- double getNdvid () const
- double getGalutinis () const
- double getMediana () const

- void paskaiciuoti_vid_ir_med ()
- void paskaiciuoti_gal ()
- double galutinis_vidurkis () const
- double galutinis_mediana () const
- istream & readStudent (istream &is)
- std::istream & read (std::istream &in) override
- std::ostream & print (std::ostream &out) const override

Public Member Functions inherited from Zmogus

- Zmogus ()=default
- Zmogus (std::string v, std::string p)
- virtual ~Zmogus ()=0
- const std::string & getVardas () const
- const std::string & getPavarde () const

Public Attributes

- string pav
- string var
- int egrez
- int pazkiek = 0
- int * P
- vector< int > paz
- double ndvid
- double gal
- double med

Private Attributes

- vector< int > paz_
- int egrez_
- double gal_
- double ndvid_
- double med_

Friends

- std::istream & operator>> (std::istream &in, Stud &s)
- std::ostream & operator<< (std::ostream &os, const Stud &s)

Additional Inherited Members

Protected Attributes inherited from Zmogus

- string var_
- string pav_

5.1.1 Constructor & Destructor Documentation

5.1.1.1 Stud() [1/5]

```
Stud::Stud ()
```

5.1.1.2 ~Stud()

```
Stud::~Stud ()  [override], [default]
```

5.1.1.3 Stud() [2/5]

```
Stud::Stud (
            const Stud & other)
```

5.1.1.4 Stud() [3/5]

```
Stud::Stud (
            Stud && other)  [noexcept]
```

kopijavimo assignment

5.1.1.5 Stud() [4/5]

```
Stud::Stud (
            const string & vardas,
            const string & pavarde,
            const vector< int > & pazymiai,
            const int egzaminas)
```

kelimo assignment

5.1.1.6 Stud() [5/5]

```
Stud::Stud (
            istream & is)
```

5.1.2 Member Function Documentation

5.1.2.1 addPaz()

```
void Stud::addPaz (
            int p)  [inline]
```

5.1.2.2 clearPaz()

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

5.1.2.3 galutinis_mediana()

```
double Stud::galutinis_mediana () const  [inline]
```

5.1.2.4 galutinis_vidurkis()

```
double Stud::galutinis_vidurkis () const  [inline]
```

5.1.2.5 getEgzaminas()

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

5.1.2.6 getGalutinis()

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

5.1.2.7 getMediana()

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

5.1.2.8 getNdvid()

```
double Stud::getNdvid () const  [inline]
```

5.1.2.9 getPazymiai()

```
vector< int > & Stud::getPazymiai ()  [inline]
```
getteriai

5.1.2.10 operator=() [1/2]

```
Stud & Stud::operator= (
            const Stud & other)
```
kopijavimo konstruktorius

5.1.2.11 operator=() [2/2]

```
Stud & Stud::operator= (
            Stud && other)  [noexcept]
```
kelimo konstruktorius

5.1.2.12 paskaiciuoti_gal()

```
void Stud::paskaiciuoti_gal ()
```

5.1.2.13 paskaiciuoti_vid_ir_med()

```
void Stud::paskaiciuoti_vid_ir_med ()
```

5.1.2.14 print()

```
std::ostream & Stud::print (
            std::ostream & out) const  [override], [virtual]
```
Implements Zmogus.

5.1.2.15 read()

```
std::istream & Stud::read (
            std::istream & in)  [override], [virtual]
```
Implements Zmogus.

5.1.2.16 readStudent()

```
std::istream & Stud::readStudent (
            istream & is)
```
rankinis/automatinis irasymas (ne is failo)

5.1.2.17 setEgrez()

```
void Stud::setEgrez (
            int e)  [inline]
```

5.1.2.18 setGal()

```
void Stud::setGal (
            double g)  [inline]
```

5.1.2.19 setMed()

```
void Stud::setMed (
            double m)  [inline]
```

5.1.2.20 setNdvid()

```
void Stud::setNdvid (
            double v)  [inline]
```

5.1.2.21 setPav()

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

5.1.2.22 setVar()

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

setteriai

5.1.3 Friends And Related Symbol Documentation

5.1.3.1 operator<<

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

5.1.3.2 operator>>

```
std::istream & operator>> (
            std::istream & in,
            Stud & s)  [friend]
```

5.1.4 Member Data Documentation

5.1.4.1 egrez

```
int Stud::egrez
```

5.1.4.2 egrez_

```
int Stud::egrez_  [private]
```

5.1.4.3 gal

```
double Stud::gal
```

5.1.4.4 gal_

```
double Stud::gal_  [private]
```

5.1.4.5 med

```
double Stud::med
```

5.1.4.6 med_

```
double Stud::med_  [private]
```

5.1.4.7 ndvid

```
double Stud::ndvid
```

5.1.4.8 ndvid_

```
double Stud::ndvid_  [private]
```

5.1.4.9 P

```
int * Stud::P
```

5.1.4.10 pav

```
string Stud::pav
```

5.1.4.11 paz

```
vector< int > Stud::paz
```

5.1.4.12 paz_

```
vector<int> Stud::paz_  [private]
```

5.1.4.13 pazkiek

```
int Stud::pazkiek = 0
```

5.1.4.14 var

```
string Stud::var
```

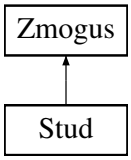
The documentation for this struct was generated from the following files:

- deque/dequeh.h
- list/listh.h
- vector/vectorh.h
- deque/dequefunk.cpp

5.2 Zmogus Class Reference

```
#include <dequeh.h>
```

Inheritance diagram for Zmogus:



Public Member Functions

- Zmogus ()=default
- Zmogus (std::string v, std::string p)
- virtual ~Zmogus ()=0
- virtual std::istream & read (std::istream &in)=0
- virtual std::ostream & print (std::ostream &out) const =0
- const std::string & getVardas () const
- const std::string & getPavarde () const

Protected Attributes

- string var_
- string pav_

5.2.1 Constructor & Destructor Documentation

5.2.1.1 Zmogus() [1/2]

```
Zmogus::Zmogus ()  [default]
```

5.2.1.2 Zmogus() [2/2]

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

5.2.1.3 ~Zmogus()

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

5.2.2 Member Function Documentation

5.2.2.1 getPavarde()

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

5.2.2.2 getVardas()

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

5.2.2.3 print()

```
virtual std::ostream & Zmogus::print (
            std::ostream & out) const  [pure virtual]
```

Implemented in Stud.

5.2.2.4 read()

```
virtual std::istream & Zmogus::read (
            std::istream & in)  [pure virtual]
```

Implemented in Stud.

5.2.3 Member Data Documentation

5.2.3.1 pav_

```
string Zmogus::pav_  [protected]
```

5.2.3.2 var_

```
string Zmogus::var_  [protected]
```

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

- deque/dequeh.h

Chapter 6

File Documentation

6.1 deque/dequefunk.cpp File Reference

```
#include "dequeh.h"
```

Functions

- bool failasegzistuoja (const string &failopav)

  1 - mediana, 2 - vidurkis (isvedimui)
- bool sortVardu (const Stud &a, const Stud &b)
- bool sortPav (const Stud &a, const Stud &b)
- bool sortMed (const Stud &a, const Stud &b)
- bool sortGal (const Stud &a, const Stud &b)
- void rng (vector< int > &paz)
- void rng (int &egrez)

  gauname adresa egrez, sugeneruojame ir grazinam
- void rng (string &vardas, string &pavarde)

  egz pazymiu gen
- std::istream & operator>> (std::istream &in, Zmogus &z)
- std::ostream & operator<< (std::ostream &os, const Zmogus &z)
- vector< Stud > processBatch (const vector< string > &lines)
- void skaitymas (deque< Stud > &grupe, duration< double > &veiklaik)

  vardu generavimas
- void isvedimas (int pas, int pasmv, const deque< Stud > &grupe)
- bool gautteisinga (int &input, const string &prompt, int min, int max)
- void failogen (const string &failopav, int irasuk)

  error handlingas neteisingos ivesties atveju
- void atrinkimas1 (deque< Stud > &grupe, vector< Stud > &nerdai, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas2 (deque< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas3 (deque< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)

Variables

- int g_printMode = 1

6.1.1 Function Documentation

6.1.1.1 atrinkimas1()

```
void atrinkimas1 (
            deque< Stud > & grupe,
            vector< Stud > & nerdai,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.1.1.2 atrinkimas2()

```
void atrinkimas2 (
            deque< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.1.1.3 atrinkimas3()

```
void atrinkimas3 (
            deque< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.1.1.4 failasegzistuoja()

```
bool failasegzistuoja (
            const string & failopav)
```

1 - mediana, 2 - vidurkis (isvedimui)

6.1.1.5 failogen()

```
void failogen (
            const string & failopav,
            int irasuk)
```

error handlingas neteisingos ivesties atveju
int kiekpaz = pazymk(gen);

6.1.1.6 gautteisinga()

```
bool gautteisinga (
            int & input,
            const string & prompt,
            int min,
            int max)
```

6.1.1.7 isvedimas()

```
void isvedimas (
            int pas,
            int pasmv,
            const deque< Stud > & grupe)
```

6.1.1.8 operator<<()

```
std::ostream & operator<< (
            std::ostream & os,
            const Zmogus & z)
```

6.1.1.9 operator>>()

```
std::istream & operator>> (
            std::istream & in,
            Zmogus & z)
```

6.1.1.10 processBatch()

```
vector< Stud > processBatch (
            const vector< string > & lines)
```

6.1.1.11 rng() [1/3]

```
void rng (
            int & egrez)
```

gauname adresa egrez, sugeneruojame ir grazinam
nd pazymiu gen

6.1.1.12 rng() [2/3]

```
void rng (
            string & vardas,
            string & pavarde)
```

egz pazymiu gen

6.1.1.13 rng() [3/3]

```
void rng (
            vector< int > & paz)
```

6.1.1.14 skaitymas()

```
void skaitymas (
            deque< Stud > & grupe,
            duration< double > & veiklaik)
```

vardu generavimas
return;
return;

6.1.1.15 sortGal()

```
bool sortGal (
            const Stud & a,
            const Stud & b)
```

6.1.1.16 sortMed()

```
bool sortMed (
            const Stud & a,
            const Stud & b)
```

6.1.1.17 sortPav()

```
bool sortPav (
            const Stud & a,
            const Stud & b)
```

6.1.1.18 sortVardu()

```
bool sortVardu (
            const Stud & a,
            const Stud & b)
```

6.1.2 Variable Documentation

6.1.2.1 g_printMode

```
int g_printMode = 1
```

6.2 deque/dequeh.h File Reference

```
#include <iostream>
#include <vector>
#include <iomanip>
#include <string>
#include <limits>
#include <random>
#include <fstream>
#include <sstream>
#include <algorithm>
#include <thread>
#include <execution>
#include <future>
#include <functional>
#include <deque>
#include <chrono>
#include <list>
#include <numeric>
```

Classes

- class Zmogus
- struct Stud

Functions

- std::istream & operator>> (std::istream &in, Zmogus &z)
- std::ostream & operator<< (std::ostream &os, Zmogus const &z)
- bool sortVardu (const Stud &a, const Stud &b)
- bool sortPav (const Stud &a, const Stud &b)
- bool sortMed (const Stud &a, const Stud &b)
- bool sortGal (const Stud &a, const Stud &b)
- double mediana (const vector< int > &paz, double egrez, string var)
- double galvid (double egrez, double ndvd)
- double ndvid (const vector< int > &paz)
- void rng (vector< int > &paz)
- void rng (int &egrez)

  nd pazymiu gen
- void rng (string &vardas, string &pavarde)

  egz pazymiu gen
- void skaitymas (deque< Stud > &grupe, duration< double > &veiklaik)

  vardu generavimas
- void isvedimas (int pas, int pasmv, const deque< Stud > &grupe)

- bool gautteisinga (int &input, const string &prompt, int min, int max)
- void failogen (const string &failopav, int irasuk)

  error handlingas neteisingos ivesties atveju
- void atrinkimas1 (deque< Stud > &grupe, vector< Stud > &nerdai, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas2 (deque< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas3 (deque< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- vector< Stud > processBatch (const vector< string > &lines)

Variables

- const string vardai [ ] = { "Jonas", "Petras", "Marius", "Lukas", "Tomas", "Simas", "Andrius", "Darius" }
- const string pavardes [ ] = { "Kazlauskas", "Petraitis", "Jonaitis", "Mikalauskas", "Bagdonas", "Vaitkus", "Urbonas", "Grigas" }

6.2.1 Function Documentation

6.2.1.1 atrinkimas1()

```
void atrinkimas1 (
            deque< Stud > & grupe,
            vector< Stud > & nerdai,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.2.1.2 atrinkimas2()

```
void atrinkimas2 (
            deque< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.2.1.3 atrinkimas3()

```
void atrinkimas3 (
            deque< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.2.1.4 failogen()

```
void failogen (
            const string & failopav,
            int irasuk)
```

error handlingas neteisingos ivesties atveju
int kiekpaz = pazymk(gen);

6.2.1.5 galvid()

```
double galvid (
            double egrez,
            double ndvd)
```

6.2.1.6 gautteisinga()

```
bool gautteisinga (
            int & input,
            const string & prompt,
            int min,
            int max)
```

6.2.1.7 isvedimas()

```
void isvedimas (
            int pas,
            int pasmv,
            const deque< Stud > & grupe)
```

6.2.1.8 mediana()

```
double mediana (
            const vector< int > & paz,
            double egrez,
            string var)
```

6.2.1.9 ndvid()

```
double ndvid (
            const vector< int > & paz)
```

6.2.1.10 operator<<()

```
std::ostream & operator<< (
            std::ostream & os,
            Zmogus const & z)
```

6.2.1.11 operator>>()

```
std::istream & operator>> (
            std::istream & in,
            Zmogus & z)
```

6.2.1.12 processBatch()

```
vector< Stud > processBatch (
            const vector< string > & lines)
```

6.2.1.13 rng() [1/3]

```
void rng (
            int & egrez)
```

nd pazymiu gen
nd pazymiu gen

6.2.1.14 rng() [2/3]

```
void rng (
            string & vardas,
            string & pavarde)
```

egz pazymiu gen

6.2.1.15 rng() [3/3]

```
void rng (
            vector< int > & paz)
```

6.2.1.16 skaitymas()

```
void skaitymas (
            deque< Stud > & grupe,
            duration< double > & veiklaik)
```

vardu generavimas
return;
return;

6.2.1.17 sortGal()

```
bool sortGal (
            const Stud & a,
            const Stud & b)
```

6.2.1.18 sortMed()

```
bool sortMed (
            const Stud & a,
            const Stud & b)
```

6.2.1.19 sortPav()

```
bool sortPav (
            const Stud & a,
            const Stud & b)
```

6.2.1.20 sortVardu()

```
bool sortVardu (
            const Stud & a,
            const Stud & b)
```

6.2.2 Variable Documentation

6.2.2.1 pavardes

```
const string pavardes[] = { "Kazlauskas", "Petraitis", "Jonaitis", "Mikalauskas", "Bagdonas",
"Vaitkus", "Urbonas", "Grigas" }
```

6.2.2.2 vardai

```
const string vardai[] = { "Jonas", "Petras", "Marius", "Lukas", "Tomas", "Simas", "Andrius",
"Darius" }
```

6.3 dequeh.h

Go to the documentation of this file.

```
00001 #ifndef DEQU_LIB_H
00002 #define DEQU_LIB_H
00003 #include <iostream>
00004 #include <vector>
00005 #include <iomanip>
00006 #include <string>
00007 #include <limits>
00008 #include <random>
00009 #include <fstream>
```

```
00010 #include <sstream>
00011 #include <algorithm> //sortui
00012 #include <thread>
00013 #include <execution>
00014 #include <future>
00015 #include <functional> //del greater ir less funkciju
00016 #include <deque>
00017 #include <chrono>
00018 #include <list>
00019 #include <numeric>
00020
00021 using std::cout;
00022 using std::cin;
00023 using std::min;
00024 using std::endl;
00025 using std::vector;
00026 using std::stringstream;
00027 using std::launch;
00028 using std::fixed;
00029 using std::setw;
00030 using std::left;
00031 using std::streamsize;
00032 using std::numeric_limits;
00033 using std::random_device;
00034 using std::mt19937;
00035 using std::uniform_int_distribution;
00036 using std::string;
00037 using std::sort;
00038 using std::setprecision;
00039 using std::ifstream;
00040 using std::ofstream;
00041 using std::ws;
00042 using std::istringstream;
00043 using std::cerr;
00044 using std::nth_element;
00045 using std::future;
00046 using std::istream;
00047 using std::runtime_error;
00048 using std::ios;
00049 using std::exception;
00050 using std::to_string;
00051 using std::ostringstream;
00052 using std::deque;
00053 using std::list;
00054 using std::chrono::high_resolution_clock;
00055 using std::chrono::duration;
00056 using std::copy;
00057 using std::back_inserter;
00058 using std::accumulate;
00059
00060 const string vardai[] = { "Jonas", "Petras", "Marius", "Lukas", "Tomas", "Simas", "Andrius", "Darius"
      };
00061 const string pavardes[] = { "Kazlauskas", "Petraitis", "Jonaitis", "Mikalauskas", "Bagdonas",
      "Vaitkus", "Urbonas", "Grigas" };
00062
00063 class Zmogus {
00064 protected:
00065     string var_;
00066     string pav_;
00067 public:
00068     Zmogus() = default;
00069     Zmogus(std::string v, std::string p)
00070         : var_(std::move(v)), pav_(std::move(p)) {
00071     }
00072     virtual ~Zmogus() = 0;
00073     virtual std::istream& read(std::istream& in) = 0;
00074     virtual std::ostream& print(std::ostream& out) const = 0;
00075
00076     const std::string& getVardas()  const { return var_; }
00077     const std::string& getPavarde() const { return pav_; }
00078 };
00079 inline Zmogus::~Zmogus() = default;
00080
00081 class Stud : public Zmogus {
00082
00083 private:
00084
00085     vector<int>paz_;
00086     int egrez_;
00087     double gal_;
00088     double ndvid_;
00089     double med_;
00090
00091
00092 public:
00093     Stud();
00094     ~Stud() override = default;
```

```
00095
00096     Stud(const Stud& other);
00097     Stud& operator=(const Stud&);
00098     Stud(Stud&&) noexcept;
00099     Stud& operator=(Stud&&) noexcept;
00100
00101
00102
00103
00104     Stud(const string& vardas, const string& pavarde, const vector<int>& pazymiai, const int
      egzaminas);
00105     Stud(istream& is);
00106
00108     void setVar(const string& v) { var_ = v; }
00109     void setPav(const string& p) { pav_ = p; }
00110     void addPaz(int p) { paz_.push_back(p); }
00111     void setEgrez(int e) { egrez_ = e; }
00112     void setNdvid(double v) { ndvid_ = v; }
00113     void setGal(double g) { gal_ = g; }
00114     void setMed(double m) { med_ = m; }
00115     void clearPaz() { paz_.clear(); }
00116
00117
00119     inline vector<int>& getPazymiai() { return paz_; }
00120     inline int getEgzaminas() const { return egrez_; }
00121     inline double getNdvid() const { return ndvid_; }
00122     inline double getGalutinis() const { return gal_; }
00123     inline double getMediana() const { return med_; }
00124
00125     void paskaiciuoti_vid_ir_med();
00126     void paskaiciuoti_gal();
00127     double galutinis_vidurkis() const { return gal_; }
00128     double galutinis_mediana() const { return med_; }
00129
00130     istream& readStudent(istream& is);
00131
00132     std::istream& read(std::istream& in) override;
00133     std::ostream& print(std::ostream& out) const override;
00134
00135     friend std::istream& operator»(std::istream& in, Stud& s) {
00136         return s.read(in);
00137     }
00138     friend std::ostream& operator«(std::ostream& os, const Stud& s) {
00139         return s.print(os);
00140     }
00141 };
00142 std::istream& operator»(std::istream& in, Zmogus& z);
00143 std::ostream& operator«(std::ostream& os, Zmogus const& z);
00144
00145 bool sortVardu(const Stud& a, const Stud& b);
00146 bool sortPav(const Stud& a, const Stud& b);
00147 bool sortMed(const Stud& a, const Stud& b);
00148 bool sortGal(const Stud& a, const Stud& b);
00149 double mediana(const vector<int>& paz, double egrez, string var);
00150 double galvid(double egrez, double ndvd);
00151 double ndvid(const vector<int>& paz);
00152 void rng(vector<int>& paz);
00153 void rng(int& egrez);
00154 void rng(string& vardas, string& pavarde);
00155 void skaitymas(deque<Stud>& grupe, duration<double>& veiklaik);
00156 void isvedimas(int pas, int pasmv, const deque<Stud>& grupe);
00157 bool gautteisinga(int& input, const string& prompt, int min, int max);
00158 void failogen(const string& failopav, int irasuk);
00159 void atrinkimas1(deque<Stud>& grupe, vector<Stud>& nerdai, vector<Stud>& galiorka, int pasmv,
      duration<double>& veiklaik);
00160 void atrinkimas2(deque<Stud>& grupe, vector<Stud>& galiorka, int pasmv, duration<double>& veiklaik);
00161 void atrinkimas3(deque<Stud>& grupe, vector<Stud>& galiorka, int pasmv, duration<double>& veiklaik);
00162 vector<Stud> processBatch(const vector<string>& lines);
00163 #endif
```

6.4 deque/dequemain.cpp File Reference

```
#include "dequeh.h"
```

Functions

- int main ()

6.4.1 Function Documentation

6.4.1.1 main()

```
int main ()
```

```
grupe.sort(sortVardu);
grupe.sort(sortPav);
grupe.sort(sortMed);
grupe.sort(sortGal);
cout << "Rusiavimo pagal " << pasirn << " laikas:" << duration<double>(end - start).count() << endl;
return 1;
return 1;
cout << "Visos programos testo laikas: " << veiklaik.count() << endl;
```

6.5 deque/test.cpp File Reference

```
#include "dequeh.h"
#include <cassert>
#include <iostream>
#include <sstream>
```

Functions

- void test_rule_of_five ()
- void test_io ()
- int main ()

6.5.1 Function Documentation

6.5.1.1 main()

```
int main ()
```

6.5.1.2 test_io()

```
void test_io ()
```

6.5.1.3 test_rule_of_five()

```
void test_rule_of_five ()
```

6.6 list/listfunk.cpp File Reference

```
#include "listh.h"
```

Functions

- bool failasegzistuoja (const string &failopav)
- bool sortVardu (const Stud &a, const Stud &b)
- bool sortPav (const Stud &a, const Stud &b)
- bool sortMed (const Stud &a, const Stud &b)
- bool sortGal (const Stud &a, const Stud &b)
- double mediana (const vector< int > &paz, double egrez, string var)
- double galvid (double egrez, double ndvd)
- double ndvid (const vector< int > &paz)
- void rng (vector< int > &paz)

- void rng (int &egrez)

  gauname adresa egrez, sugeneruojame ir grazinam
- void rng (string &vardas, string &pavarde)

  egz pazymiu gen
- vector< Stud > processBatch (const vector< string > &lines)
- void skaitymas (list< Stud > &grupe, duration< double > &veiklaik)

  vardu generavimas
- void isvedimas (int pas, int pasmv, const list< Stud > &grupe)
- bool gautteisinga (int &input, const string &prompt, int min, int max)
- void failogen (const string &failopav, int irasuk)

  error handlingas neteisingos ivesties atveju
- void atrinkimas1 (list< Stud > &grupe, vector< Stud > &nerdai, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas2 (list< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas3 (list< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)

6.6.1 Function Documentation

6.6.1.1 atrinkimas1()

```
void atrinkimas1 (
            list< Stud > & grupe,
            vector< Stud > & nerdai,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

auto start2 = high_resolution_clock::now();
auto end2 = high_resolution_clock::now(); veiklaik += end2 - start2; cout << "Nerdu irasymo i faila veikimo laikas: " << duration<double>(end2 - start2).count() << endl; auto start3 = high_resolution_clock::now();
auto end3 = high_resolution_clock::now(); veiklaik += end3 - start3; cout << "Galiorkos irasymo i faila veikimo laikas: " << duration<double>(end3 - start3).count() << endl;

6.6.1.2 atrinkimas2()

```
void atrinkimas2 (
            list< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.6.1.3 atrinkimas3()

```
void atrinkimas3 (
            list< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.6.1.4 failasegzistuoja()

```
bool failasegzistuoja (
            const string & failopav)
```

6.6.1.5 failogen()

```
void failogen (
            const string & failopav,
            int irasuk)
```

error handlingas neteisingos ivesties atveju
int kiekpaz = pazymk(gen);

6.6.1.6 galvid()

```
double galvid (
            double egrez,
            double ndvd)
```

6.6.1.7 gautteisinga()

```
bool gautteisinga (
            int & input,
            const string & prompt,
            int min,
            int max)
```

6.6.1.8 isvedimas()

```
void isvedimas (
            int pas,
            int pasmv,
            const list< Stud > & grupe)
```

6.6.1.9 mediana()

```
double mediana (
            const vector< int > & paz,
            double egrez,
            string var)
```

6.6.1.10 ndvid()

```
double ndvid (
            const vector< int > & paz)
```

6.6.1.11 processBatch()

```
vector< Stud > processBatch (
            const vector< string > & lines)
```

6.6.1.12 rng() [1/3]

```
void rng (
            int & egrez)
```

gauname adresa egrez, sugeneruojame ir grazinam
nd pazymiu gen

6.6.1.13 rng() [2/3]

```
void rng (
            string & vardas,
            string & pavarde)
```

egz pazymiu gen

6.6.1.14 rng() [3/3]

```
void rng (
            vector< int > & paz)
```

6.6.1.15 skaitymas()

```
void skaitymas (
            list< Stud > & grupe,
            duration< double > & veiklaik)
```

vardu generavimas

6.6.1.16 sortGal()

```
bool sortGal (
            const Stud & a,
            const Stud & b)
```

6.6.1.17 sortMed()

```
bool sortMed (
            const Stud & a,
            const Stud & b)
```

6.6.1.18 sortPav()

```
bool sortPav (
            const Stud & a,
            const Stud & b)
```

6.6.1.19 sortVardu()

```
bool sortVardu (
            const Stud & a,
            const Stud & b)
```

6.7 list/listh.h File Reference

```
#include <iostream>
#include <vector>
#include <iomanip>
#include <string>
#include <limits>
#include <random>
#include <fstream>
#include <sstream>
#include <algorithm>
#include <thread>
#include <execution>
#include <future>
#include <functional>
#include <deque>
#include <chrono>
#include <list>
```

Classes

- struct Stud

Functions

- bool sortVardu (const Stud &a, const Stud &b)
- bool sortPav (const Stud &a, const Stud &b)
- bool sortMed (const Stud &a, const Stud &b)
- bool sortGal (const Stud &a, const Stud &b)
- double mediana (const vector< int > &paz, double egrez, string var)
- double galvid (double egrez, double ndvd)
- double ndvid (const vector< int > &paz)
- void rng (vector< int > &paz)
- void rng (int &egrez)

  nd pazymiu gen
- void rng (string &vardas, string &pavarde)

  egz pazymiu gen
- void skaitymas (list< Stud > &grupe, duration< double > &veiklaik)

  vardu generavimas
- void isvedimas (int pas, int pasmv, const list< Stud > &grupe)
- bool gautteisinga (int &input, const string &prompt, int min, int max)
- void failogen (const string &failopav, int irasuk)

  error handlingas neteisingos ivesties atveju
- void atrinkimas1 (list< Stud > &grupe, vector< Stud > &nerdai, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas2 (list< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas3 (list< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- vector< Stud > processBatch (const vector< string > &lines)

Variables

- const string vardai [ ] = { "Jonas", "Petras", "Marius", "Lukas", "Tomas", "Simas", "Andrius", "Darius" }
- const string pavardes [ ] = { "Kazlauskas", "Petraitis", "Jonaitis", "Mikalauskas", "Bagdonas", "Vaitkus", "Urbonas", "Grigas" }

6.7.1 Function Documentation

6.7.1.1 atrinkimas1()

```
void atrinkimas1 (
            list< Stud > & grupe,
            vector< Stud > & nerdai,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

auto start2 = high_resolution_clock::now();
auto end2 = high_resolution_clock::now(); veiklaik += end2 - start2; cout << "Nerdu irasymo i faila veikimo laikas: " << duration<double>(end2 - start2).count() << endl; auto start3 = high_resolution_clock::now();
auto end3 = high_resolution_clock::now(); veiklaik += end3 - start3; cout << "Galiorkos irasymo i faila veikimo laikas: " << duration<double>(end3 - start3).count() << endl;

6.7.1.2 atrinkimas2()

```
void atrinkimas2 (
            list< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.7.1.3 atrinkimas3()

```
void atrinkimas3 (
            list< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.7.1.4 failogen()

```
void failogen (
            const string & failopav,
            int irasuk)
```

error handlingas neteisingos ivesties atveju
int kiekpaz = pazymk(gen);

6.7.1.5 galvid()

```
double galvid (
            double egrez,
            double ndvd)
```

6.7.1.6 gautteisinga()

```
bool gautteisinga (
            int & input,
            const string & prompt,
            int min,
            int max)
```

6.7.1.7 isvedimas()

```
void isvedimas (
            int pas,
            int pasmv,
            const list< Stud > & grupe)
```

6.7.1.8 mediana()

```
double mediana (
            const vector< int > & paz,
            double egrez,
            string var)
```

6.7.1.9 ndvid()

```
double ndvid (
            const vector< int > & paz)
```

6.7.1.10 processBatch()

```
vector< Stud > processBatch (
            const vector< string > & lines)
```

6.7.1.11 rng() [1/3]

```
void rng (
            int & egrez)
```

nd pazymiu gen
nd pazymiu gen

6.7.1.12 rng() [2/3]

```
void rng (
            string & vardas,
            string & pavarde)
```

egz pazymiu gen

6.7.1.13 rng() [3/3]

```
void rng (
            vector< int > & paz)
```

6.7.1.14 skaitymas()

```
void skaitymas (
            list< Stud > & grupe,
            duration< double > & veiklaik)
```

vardu generavimas

6.7.1.15 sortGal()

```
bool sortGal (
            const Stud & a,
            const Stud & b)
```

6.7.1.16 sortMed()

```
bool sortMed (
            const Stud & a,
            const Stud & b)
```

6.7.1.17 sortPav()

```
bool sortPav (
            const Stud & a,
            const Stud & b)
```

6.7.1.18 sortVardu()

```
bool sortVardu (
            const Stud & a,
            const Stud & b)
```

6.7.2 Variable Documentation

6.7.2.1 pavardes

```
const string pavardes[] = { "Kazlauskas", "Petraitis", "Jonaitis", "Mikalauskas", "Bagdonas",
"Vaitkus", "Urbonas", "Grigas" }
```

6.7.2.2 vardai

```
const string vardai[] = { "Jonas", "Petras", "Marius", "Lukas", "Tomas", "Simas", "Andrius",
"Darius" }
```

6.8 listh.h

Go to the documentation of this file.

```
00001 #ifndef LIST_LIB_H
00002 #define LIST_LIB_H
```

```
00003 #include <iostream>
00004 #include <vector>
00005 #include <iomanip>
00006 #include <string>
00007 #include <limits>
00008 #include <random>
00009 #include <fstream>
00010 #include <sstream>
00011 #include <algorithm> //sortui
00012 #include <thread>
00013 #include <execution>
00014 #include <future>
00015 #include <functional> //del greater ir less funkciju
00016 #include <deque>
00017 #include <chrono>
00018 #include <list>
00019
00020 using std::cout;
00021 using std::cin;
00022 using std::min;
00023 using std::endl;
00024 using std::vector;
00025 using std::stringstream;
00026 using std::launch;
00027 using std::fixed;
00028 using std::setw;
00029 using std::left;
00030 using std::streamsize;
00031 using std::numeric_limits;
00032 using std::random_device;
00033 using std::mt19937;
00034 using std::uniform_int_distribution;
00035 using std::string;
00036 using std::sort;
00037 using std::setprecision;
00038 using std::ifstream;
00039 using std::ofstream;
00040 using std::ws;
00041 using std::istringstream;
00042 using std::cerr;
00043 using std::nth_element;
00044 using std::future;
00045 using std::runtime_error;
00046 using std::ios;
00047 using std::exception;
00048 using std::to_string;
00049 using std::ostringstream;
00050 using std::deque;
00051 using std::list;
00052 using std::chrono::high_resolution_clock;
00053 using std::chrono::duration;
00054 using std::copy;
00055 using std::back_inserter;
00056
00057 const string vardai[] = { "Jonas", "Petras", "Marius", "Lukas", "Tomas", "Simas", "Andrius", "Darius"
      };
00058 const string pavardes[] = { "Kazlauskas", "Petraitis", "Jonaitis", "Mikalauskas", "Bagdonas",
      "Vaitkus", "Urbonas", "Grigas" };
00059
00060 struct Stud {
00061     string pav;
00062     string var;
00063     int egrez;
00064     int pazkiek = 0;
00065     int* P;
00066     vector<int> paz;
00067     double ndvid;
00068     double gal;
00069     double med;
00070     /*Stud() : P(nullptr), pazkiek(0) {}  ///default konstruktorius
00071
00072     ~Stud() {
00073         delete[]P;
00074         P = nullptr; ///kad nebutu kabanti rodykle
00075     }
00076
00077     Stud(const Stud& other)  ///kopijavimo konstruktorius
00078     {
00079         var = other.var;
00080         pav = other.pav;
00081         pazkiek = other.pazkiek;
00082         egrez = other.egrez;
00083         ndvid = other.ndvid;
00084         gal = other.gal;
00085         med = other.med;
00086
00087         if (other.P != nullptr) {
```

```
00088             P = new int[pazkiek];
00089             for (int i = 0; i < pazkiek; i++) {
00090                 P[i] = other.P[i];
00091             }
00092         }
00093         else {
00094             P = nullptr;
00095         }
00096     }*/
00097 };
00098
00099
00100
00101 bool sortVardu(const Stud& a, const Stud& b);
00102 bool sortPav(const Stud& a, const Stud& b);
00103 bool sortMed(const Stud& a, const Stud& b);
00104 bool sortGal(const Stud& a, const Stud& b);
00105 double mediana(const vector<int>& paz, double egrez, string var);
00106 double galvid(double egrez, double ndvd);
00107 double ndvid(const vector<int>& paz);
00108 void rng(vector<int>& paz);
00109 void rng(int& egrez);
00110 void rng(string& vardas, string& pavarde);
00111 void skaitymas(list<Stud>& grupe, duration<double>& veiklaik);
00112 void isvedimas(int pas, int pasmv, const list<Stud>& grupe);
00113 bool gautteisinga(int& input, const string& prompt, int min, int max);
00114 void failogen(const string& failopav, int irasuk);
00115 void atrinkimas1(list<Stud>& grupe, vector<Stud>& nerdai, vector<Stud>& galiorka, int pasmv,
      duration<double>& veiklaik);
00116 void atrinkimas2(list<Stud>& grupe, vector<Stud>& galiorka, int pasmv, duration<double>& veiklaik);
00117 void atrinkimas3(list<Stud>& grupe, vector<Stud>& galiorka, int pasmv, duration<double>& veiklaik);
00118 vector<Stud> processBatch(const vector<string>& lines);
00119 #endif
```

6.9 list/listmain.cpp File Reference

```
#include "listh.h"
```

Functions

- int main ()

6.9.1 Function Documentation

6.9.1.1 main()

```
int main ()
```

6.10 README.md File Reference

6.11 vector/vectorfunk.cpp File Reference

```
#include "vectorh.h"
```

Functions

- bool failasegzistuoja (const string &failopav)
- bool sortVardu (const Stud &a, const Stud &b)
- bool sortPav (const Stud &a, const Stud &b)
- bool sortMed (const Stud &a, const Stud &b)
- bool sortGal (const Stud &a, const Stud &b)
- double mediana (const vector< int > &paz, double egrez, string var)
- double galvid (double egrez, double ndvd)
- double ndvid (const vector< int > &paz)
- void rng (vector< int > &paz)

- void rng (int &egrez)

  gauname adresa egrez, sugeneruojame ir grazinam
- void rng (string &vardas, string &pavarde)

  egz pazymiu gen
- vector< Stud > processBatch (const vector< string > &lines)
- void skaitymas (vector< Stud > &grupe, duration< double > &veiklaik)

  vardu generavimas
- void isvedimas (int pas, int pasmv, const vector< Stud > &grupe)
- bool gautteisinga (int &input, const string &prompt, int min, int max)
- void failogen (const string &failopav, int irasuk)

  error handlingas neteisingos ivesties atveju
- void atrinkimas1 (vector< Stud > &grupe, vector< Stud > &nerdai, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas2 (vector< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas3 (vector< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)

6.11.1 Function Documentation

6.11.1.1 atrinkimas1()

```
void atrinkimas1 (
            vector< Stud > & grupe,
            vector< Stud > & nerdai,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

auto start2 = high_resolution_clock::now();
auto end2 = high_resolution_clock::now(); veiklaik += end2 - start2; cout << "Nerdu irasymo i faila veikimo laikas: " << duration<double>(end2 - start2).count() << endl; auto start3 = high_resolution_clock::now();
auto end3 = high_resolution_clock::now(); veiklaik += end3 - start3; cout << "Galiorkos irasymo i faila veikimo laikas: " << duration<double>(end3 - start3).count() << endl;

6.11.1.2 atrinkimas2()

```
void atrinkimas2 (
            vector< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.11.1.3 atrinkimas3()

```
void atrinkimas3 (
            vector< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.11.1.4 failasegzistuoja()

```
bool failasegzistuoja (
            const string & failopav)
```

6.11.1.5 failogen()

```
void failogen (
            const string & failopav,
            int irasuk)
```

error handlingas neteisingos ivesties atveju
int kiekpaz = pazymk(gen);

6.11.1.6 galvid()

```
double galvid (
            double egrez,
            double ndvd)
```

6.11.1.7 gautteisinga()

```
bool gautteisinga (
            int & input,
            const string & prompt,
            int min,
            int max)
```

6.11.1.8 isvedimas()

```
void isvedimas (
            int pas,
            int pasmv,
            const vector< Stud > & grupe)
```

6.11.1.9 mediana()

```
double mediana (
            const vector< int > & paz,
            double egrez,
            string var)
```

6.11.1.10 ndvid()

```
double ndvid (
            const vector< int > & paz)
```

6.11.1.11 processBatch()

```
vector< Stud > processBatch (
            const vector< string > & lines)
```

6.11.1.12 rng() [1/3]

```
void rng (
            int & egrez)
```

gauname adresa egrez, sugeneruojame ir grazinam
nd pazymiu gen

6.11.1.13 rng() [2/3]

```
void rng (
            string & vardas,
            string & pavarde)
```

egz pazymiu gen

6.11.1.14 rng() [3/3]

```
void rng (
            vector< int > & paz)
```

6.11.1.15 skaitymas()

```
void skaitymas (
            vector< Stud > & grupe,
            duration< double > & veiklaik)
```

vardu generavimas

6.11.1.16 sortGal()

```
bool sortGal (
            const Stud & a,
            const Stud & b)
```

6.11.1.17 sortMed()

```
bool sortMed (
            const Stud & a,
            const Stud & b)
```

6.11.1.18 sortPav()

```
bool sortPav (
            const Stud & a,
            const Stud & b)
```

6.11.1.19 sortVardu()

```
bool sortVardu (
            const Stud & a,
            const Stud & b)
```

6.12 vector/vectorh.h File Reference

```
#include <iostream>
#include <vector>
#include <iomanip>
#include <string>
#include <limits>
#include <random>
#include <fstream>
#include <sstream>
#include <algorithm>
#include <thread>
#include <execution>
#include <future>
#include <functional>
#include <deque>
#include <chrono>
#include <list>
```

Classes

- struct Stud

Functions

- bool sortVardu (const Stud &a, const Stud &b)
- bool sortPav (const Stud &a, const Stud &b)
- bool sortMed (const Stud &a, const Stud &b)
- bool sortGal (const Stud &a, const Stud &b)
- double mediana (const vector< int > &paz, double egrez, string var)
- double galvid (double egrez, double ndvd)
- double ndvid (const vector< int > &paz)
- void rng (vector< int > &paz)
- void rng (int &egrez)

  nd pazymiu gen
- void rng (string &vardas, string &pavarde)

  egz pazymiu gen
- void skaitymas (vector< Stud > &grupe, duration< double > &veiklaik)

  vardu generavimas
- void isvedimas (int pas, int pasmv, const vector< Stud > &grupe)
- bool gautteisinga (int &input, const string &prompt, int min, int max)
- void failogen (const string &failopav, int irasuk)

  error handlingas neteisingos ivesties atveju
- void atrinkimas1 (vector< Stud > &grupe, vector< Stud > &nerdai, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas2 (vector< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- void atrinkimas3 (vector< Stud > &grupe, vector< Stud > &galiorka, int pasmv, duration< double > &veiklaik)
- vector< Stud > processBatch (const vector< string > &lines)

Variables

- const string vardai [ ] = { "Jonas", "Petras", "Marius", "Lukas", "Tomas", "Simas", "Andrius", "Darius" }
- const string pavardes [ ] = { "Kazlauskas", "Petraitis", "Jonaitis", "Mikalauskas", "Bagdonas", "Vaitkus", "Urbonas", "Grigas" }

6.12.1 Function Documentation

6.12.1.1 atrinkimas1()

```
void atrinkimas1 (
            vector< Stud > & grupe,
            vector< Stud > & nerdai,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

auto start2 = high_resolution_clock::now();
auto end2 = high_resolution_clock::now(); veiklaik += end2 - start2; cout << "Nerdu irasymo i faila veikimo laikas: " << duration<double>(end2 - start2).count() << endl; auto start3 = high_resolution_clock::now();
auto end3 = high_resolution_clock::now(); veiklaik += end3 - start3; cout << "Galiorkos irasymo i faila veikimo laikas: " << duration<double>(end3 - start3).count() << endl;

6.12.1.2 atrinkimas2()

```
void atrinkimas2 (
            vector< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.12.1.3 atrinkimas3()

```
void atrinkimas3 (
            vector< Stud > & grupe,
            vector< Stud > & galiorka,
            int pasmv,
            duration< double > & veiklaik)
```

6.12.1.4 failogen()

```
void failogen (
            const string & failopav,
            int irasuk)
```

error handlingas neteisingos ivesties atveju
int kiekpaz = pazymk(gen);

6.12.1.5 galvid()

```
double galvid (
            double egrez,
            double ndvd)
```

6.12.1.6 gautteisinga()

```
bool gautteisinga (
            int & input,
            const string & prompt,
            int min,
            int max)
```

6.12.1.7 isvedimas()

```
void isvedimas (
            int pas,
            int pasmv,
            const vector< Stud > & grupe)
```

6.12.1.8 mediana()

```
double mediana (
            const vector< int > & paz,
            double egrez,
            string var)
```

6.12.1.9 ndvid()

```
double ndvid (
            const vector< int > & paz)
```

6.12.1.10 processBatch()

```
vector< Stud > processBatch (
            const vector< string > & lines)
```

6.12.1.11 rng() [1/3]

```
void rng (
            int & egrez)
```

nd pazymiu gen
nd pazymiu gen

6.12.1.12 rng() [2/3]

```
void rng (
            string & vardas,
            string & pavarde)
```

egz pazymiu gen

6.12.1.13 rng() [3/3]

```
void rng (
            vector< int > & paz)
```

6.12.1.14 skaitymas()

```
void skaitymas (
            vector< Stud > & grupe,
            duration< double > & veiklaik)
```

vardu generavimas

6.12.1.15 sortGal()

```
bool sortGal (
            const Stud & a,
            const Stud & b)
```

6.12.1.16 sortMed()

```
bool sortMed (
            const Stud & a,
            const Stud & b)
```

6.12.1.17 sortPav()

```
bool sortPav (
            const Stud & a,
            const Stud & b)
```

6.12.1.18 sortVardu()

```
bool sortVardu (
            const Stud & a,
            const Stud & b)
```

6.12.2 Variable Documentation

6.12.2.1 pavardes

```
const string pavardes[] = { "Kazlauskas", "Petraitis", "Jonaitis", "Mikalauskas", "Bagdonas",
"Vaitkus", "Urbonas", "Grigas" }
```

6.12.2.2 vardai

```
const string vardai[] = { "Jonas", "Petras", "Marius", "Lukas", "Tomas", "Simas", "Andrius",
"Darius" }
```

6.13 vectorh.h

Go to the documentation of this file.

```
00001 #ifndef VEKT_LIB_H
00002 #define VEKT_LIB_H
```

```
00003 #include <iostream>
00004 #include <vector>
00005 #include <iomanip>
00006 #include <string>
00007 #include <limits>
00008 #include <random>
00009 #include <fstream>
00010 #include <sstream>
00011 #include <algorithm> //sortui
00012 #include <thread>
00013 #include <execution>
00014 #include <future>
00015 #include <functional> //del greater ir less funkciju
00016 #include <deque>
00017 #include <chrono>
00018 #include <list>
00019
00020 using std::cout;
00021 using std::cin;
00022 using std::min;
00023 using std::endl;
00024 using std::vector;
00025 using std::stringstream;
00026 using std::launch;
00027 using std::fixed;
00028 using std::setw;
00029 using std::left;
00030 using std::streamsize;
00031 using std::numeric_limits;
00032 using std::random_device;
00033 using std::mt19937;
00034 using std::uniform_int_distribution;
00035 using std::string;
00036 using std::sort;
00037 using std::setprecision;
00038 using std::ifstream;
00039 using std::ofstream;
00040 using std::ws;
00041 using std::istringstream;
00042 using std::cerr;
00043 using std::nth_element;
00044 using std::future;
00045 using std::runtime_error;
00046 using std::ios;
00047 using std::exception;
00048 using std::to_string;
00049 using std::ostringstream;
00050 using std::deque;
00051 using std::list;
00052 using std::chrono::high_resolution_clock;
00053 using std::chrono::duration;
00054 using std::copy;
00055 using std::back_inserter;
00056
00057 const string vardai[] = { "Jonas", "Petras", "Marius", "Lukas", "Tomas", "Simas", "Andrius", "Darius"
      };
00058 const string pavardes[] = { "Kazlauskas", "Petraitis", "Jonaitis", "Mikalauskas", "Bagdonas",
      "Vaitkus", "Urbonas", "Grigas" };
00059
00060 struct Stud {
00061     string pav;
00062     string var;
00063     int egrez;
00064     int pazkiek = 0;
00065     int* P;
00066     vector<int> paz;
00067     double ndvid;
00068     double gal;
00069     double med;
00070     /*Stud() : P(nullptr), pazkiek(0) {}  ///default konstruktorius
00071
00072     ~Stud() {
00073         delete[]P;
00074         P = nullptr; ///kad nebutu kabanti rodykle
00075     }
00076
00077     Stud(const Stud& other)  ///kopijavimo konstruktorius
00078     {
00079         var = other.var;
00080         pav = other.pav;
00081         pazkiek = other.pazkiek;
00082         egrez = other.egrez;
00083         ndvid = other.ndvid;
00084         gal = other.gal;
00085         med = other.med;
00086
00087         if (other.P != nullptr) {
```

```
00088             P = new int[pazkiek];
00089             for (int i = 0; i < pazkiek; i++) {
00090                 P[i] = other.P[i];
00091             }
00092         }
00093         else {
00094             P = nullptr;
00095         }
00096     }*/
00097 };
00098
00099
00100
00101 bool sortVardu(const Stud& a, const Stud& b);
00102 bool sortPav(const Stud& a, const Stud& b);
00103 bool sortMed(const Stud& a, const Stud& b);
00104 bool sortGal(const Stud& a, const Stud& b);
00105 double mediana(const vector<int>& paz, double egrez, string var);
00106 double galvid(double egrez, double ndvd);
00107 double ndvid(const vector<int>& paz);
00108 void rng(vector<int>& paz);
00109 void rng(int& egrez);
00110 void rng(string& vardas, string& pavarde);
00111 void skaitymas(vector<Stud>& grupe, duration<double>& veiklaik);
00112 void isvedimas(int pas, int pasmv, const vector<Stud>& grupe);
00113 bool gautteisinga(int& input, const string& prompt, int min, int max);
00114 void failogen(const string& failopav, int irasuk);
00115 void atrinkimas1(vector<Stud>& grupe, vector<Stud>& nerdai, vector<Stud>& galiorka, int pasmv,
      duration<double>& veiklaik);
00116 void atrinkimas2(vector<Stud>& grupe, vector<Stud>& galiorka, int pasmv, duration<double>& veiklaik);
00117 void atrinkimas3(vector<Stud>& grupe, vector<Stud>& galiorka, int pasmv, duration<double>& veiklaik);
00118 vector<Stud> processBatch(const vector<string>& lines);
00119 #endif
```

6.14 vector/vectormain.cpp File Reference

```
#include "vectorh.h"
```

Functions

- int main ()

6.14.1 Function Documentation

6.14.1.1 main()

```
int main ()
```

grupe.sort(sortVardu);
grupe.sort(sortPav);
grupe.sort(sortMed);
grupe.sort(sortGal);