My Project

Generated by Doxygen 1.10.0

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

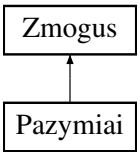
Class Documentation

4.1 Pazymiai Class Reference

Klase Pazymiai saugo informacija apie studentu pazymius.

`#include <pagalbinesvector.h>`

Inheritance diagram for Pazymiai:



Public Member Functions

- **Pazymiai** ()

  Numatytasis konstruktorius, inicializuoja narius nuliais.
- Pazymiai (const std::string &var, const std::string &pav, double vid, int egz, const std::vector< int > &paz, double galutinis, double med)

  Parametrizuotas konstruktorius, inicializuoja narius su duotomis reiksmemis.
- ~**Pazymiai** ()

  Desktruktorius, isvalo vektoriu.
- **Pazymiai** (const Pazymiai &other)

  Kopijavimo konstruktorius.
- **Pazymiai** (Pazymiai &&other) noexcept

  Perkelimo konstruktorius.
- Pazymiai & **operator=** (const Pazymiai &other)

  Kopijavimo priskyrimo operatorius.
- Pazymiai & **operator=** (Pazymiai &&other) noexcept

  Perkelimo priskyrimo operatorius.
- int **getPazN** (const std::vector< int > &newPaz, int pos) const

  Grazina nurodyta pazymi is saraso pagal pozicija.
- void **clearPaz** ()

  Isvalo vektoriu.

- void **setVid** (double newVid)

  Nustato studento vidurki.
- void **setEgz** (int newEgz)

  Nustato egzamino rezultata.
- void **setOnePaz** (int newPaz)

  Iterpia viena pazymi i sarasa..
- void **setPazymiai** (const std::vector< int > &pazymiai)

  Nustato pazymiu sarasa.
- void **setGalutinis** (double newGalutinis)

  Nustato galutini pazymi.
- void **setMed** (double newMed)

  Nustato mediana.
- void **sortPaz** (Pazymiai &C)

  Rikiuoja pazymiu sarasa.
- double **getVid** () const

  Grazina vidurki.
- int **getEgz** () const

  Grazina egzamino rezultata.
- std::vector< int > **getPaz** () const

  Grazina pazymiu sarasa.
- double **getGalutinis** () const

  Grazina galutini pazymi.
- double **getMed** () const

  Grazina mediana.
- std::string getVar () const override

  Grazina zmogaus varda.
- std::string getPav () override

  Grazina zmogaus pavarde.

Public Member Functions inherited from Zmogus

- **Zmogus** ()=default

  Numatytasis konstruktorius.
- Zmogus (const std::string &var, const std::string &pav)

  Parametrizuotas konstruktorius, inicializuoja varda ir pavarde.
- virtual ~**Zmogus** ()

  Virtualus destruktorius.
- virtual void **setVar** (const std::string &newVar)

  Nustato studento varda.
- virtual void **setPav** (const std::string &newPav)

  Nustato studento pavarde.

Private Attributes

- double vid_
- int egz_
- std::vector< int > paz_
- double galutinis_
- double med_

Friends

- double mediana (int u, const Pazymiai h)

  Skaiciuoja mediana.
- std::istream & operator>> (std::istream &is, Pazymiai &obj)

  Ivedimo operatorius, skirtas nuskaityti objekto duomenis is ivesties srauto.
- std::ostream & operator<< (std::ostream &os, const Pazymiai &obj)

  Isvedimo operatorius, skirtas isvesti objekto duomenis i isvesties srauta.

Additional Inherited Members

Protected Attributes inherited from Zmogus

- std::string var_
- std::string pav_

4.1.1 Detailed Description

Klase Pazymiai saugo informacija apie studentu pazymius.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Pazymiai()

```
Pazymiai::Pazymiai (
            const std::string & var,
            const std::string & pav,
            double vid,
            int egz,
            const std::vector< int > & paz,
            double galutinis,
            double med )  [inline]
```

Parametrizuotas konstruktorius, inicializuoja narius su duotomis reiksmemis.

Parameters

var	Studento vardas
pav	Studento pavarde
vid	Vidurkis
egz	Egzamino rezultatas
paz	Pazymiu sarasas
galutinis	Galutinis pazymys
med	Mediana

4.1.3 Member Function Documentation

4.1.3.1 getPav()

```
std::string Pazymiai::getPav ( )  [inline], [override], [virtual]
```

Grazina zmogaus pavarde.

Implements Zmogus.

4.1.3.2 getVar()

```
std::string Pazymiai::getVar ( ) const  [inline], [override], [virtual]
```

Grazina zmogaus varda.

Implements Zmogus.

4.1.4 Friends And Related Symbol Documentation

4.1.4.1 mediana

```
double mediana (
            int u,
            const Pazymiai h )  [friend]
```

Skaiciuoja mediana.

Parameters

u	Pazymiu skaicius
h	Pazymiai objektas

Returns

Mediana

4.1.4.2 operator<<

```
std::ostream & operator<< (
            std::ostream & os,
            const Pazymiai & obj )  [friend]
```

Isvedimo operatorius, skirtas isvesti objekto duomenis i isvesties srauta.

Parameters

os	Isvesties srautas
obj	Pazymiai objektas

Returns

Isvesties srautas

4.1.4.3 operator>>

```
std::istream & operator>> (
            std::istream & is,
            Pazymiai & obj )  [friend]
```

Ivedimo operatorius, skirtas nuskaityti objekto duomenis is ivesties srauto.

Parameters

is	Ivesties srautas
obj	Pazymiai objektas, i kuri nuskaitomi duomenys

Returns

Ivesties srautas

4.1.5 Member Data Documentation

4.1.5.1 egz_

```
int Pazymiai::egz_  [private]
```

Egzamino rezultatas

4.1.5.2 galutinis_

```
double Pazymiai::galutinis_  [private]
```

Galutinis pazymys

4.1.5.3 med_

```
double Pazymiai::med_  [private]
```

Mediana

4.1.5.4 paz_

```
std::vector<int> Pazymiai::paz_  [private]
```

Pazymiu sarasas

4.1.5.5 vid_

```
double Pazymiai::vid_  [private]
```

Vidurkis

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

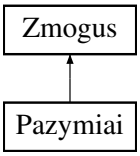
- C:/Users/Paulina/Documents/Objektinis programavimas 1 kursas/vectorcontainer/pagalbinesvector.h

4.2 Zmogus Class Reference

Klase Zmogus saugo informacija apie studento varda ir pavarde.

```
#include <pagalbinesvector.h>
```

Inheritance diagram for Zmogus:



Public Member Functions

- **Zmogus** ()=default

  Numatytasis konstruktorius.
- Zmogus (const std::string &var, const std::string &pav)

  Parametrizuotas konstruktorius, inicializuoja varda ir pavarde.
- virtual ~**Zmogus** ()

  Virtualus destruktorius.
- virtual void **setVar** (const std::string &newVar)

  Nustato studento varda.
- virtual void **setPav** (const std::string &newPav)

  Nustato studento pavarde.
- virtual std::string getVar () const =0

  Grazina zmogaus varda.
- virtual std::string getPav ()=0

  Grazina zmogaus pavarde.

Protected Attributes

- std::string var_
- std::string pav_

4.2.1 Detailed Description

Klase Zmogus saugo informacija apie studento varda ir pavarde.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Zmogus()

```
Zmogus::Zmogus (
            const std::string & var,
            const std::string & pav )  [inline]
```

Parametrizuotas konstruktorius, inicializuoja varda ir pavarde.

Parameters

var	Zmogaus vardas
pav	Zmogaus pavarde

4.2.3 Member Function Documentation

4.2.3.1 getPav()

```
virtual std::string Zmogus::getPav ( )  [pure virtual]
```

Grazina zmogaus pavarde.

Implemented in Pazymiai.

4.2.3.2 getVar()

```
virtual std::string Zmogus::getVar ( ) const  [pure virtual]
```

Grazina zmogaus varda.

Implemented in Pazymiai.

4.2.4 Member Data Documentation

4.2.4.1 pav_

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

Zmogaus pavarde

4.2.4.2 var_

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

Zmogaus vardas

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

- C:/Users/Paulina/Documents/Objektinis programavimas 1 kursas/vectorcontainer/pagalbinesvector.h

Chapter 5

File Documentation

5.1 C:/Users/Paulina/Documents/Objektinis programavimas 1 kursas/vectorcontainer/mainvector.cpp File Reference

Pagrindinio failo vykdymas.

```
#include "pagalbinesvector.h"
```

Functions

- int main ()

  Vartotojas is meniu pasirenka norima veiksma ir pagal pasirinkima, veda arba generuoja studentu duomenis.

Variables

- int **pasirinkimas**
- int **c**
- int **x**
- int **s**
- int **i** =0
- std::vector< Pazymiai > **S**
- std::vector< Pazymiai > **P**
- std::vector< Pazymiai > **Z**

5.1.1 Detailed Description

Pagrindinio failo vykdymas.

5.1.2 Function Documentation

5.1.2.1 main()

```
int main ( )
```

Vartotojas is meniu pasirenka norima veiksma ir pagal pasirinkima, veda arba generuoja studentu duomenis.

Meniu:

Atvejis 1: Ivesti duomenis ranka Si dalis leidzia vartotojui ivesti studentu duomenis rankiniu budu. Vartotojas gali vesti studentu duomenis, kol ives "4" i studento vardo arba pavardes lauka. Kai duomenu ivedimas baigtas, atvaizduojami ivesti studentu duomenys ekrane

Parameters

P	Vektorius, i kuri pridedami ivesti duomenys.
Pazymiai	C; Objektas, kuri laikinai saugomi ivesti duomenys.

Atvejis 2: Generuoja tik pazymius. Si dalis leidzia vartotojui generuoti studentu duomenis su atsitiktiniais pazymiais. Vartotojui leidziama ivesti varda ir pavarde Vartotojas pasirenka, ar nori generuoti atsitiktinius pazymius arba pasirinkti kieki Atvaizduojami ivesti studentu duomenys ekrane

Parameters

P	Vektorius, i kuri pridedami studentu duomenys.
int	i; Skaiciuoja kiek studentu iraso vartotojas.
Pazymiai	C; Studento objektas, i kuri laikinai saugomi studento duomenys.
string	xx, yy; Laikini kintamieji, kuriuose saugomi studento vardas ir pavarde.
int	egg; Kintamasis, kuriame laikinai saugomas studento egzamino rezultatas.
double	suma = 0.0; Kintamasis, kuriame laikinai saugoma pazymiu suma.
x	Laikinas kintamasis, kuriame saugomas vartotojo atsakymas apie generavima.
string	y; Laikinas kintamasis, kuriame saugomas vartotojo atsakymas apie generavimo tesima.
int	h = 0; Skaiciuoja pazymiu kieki.
int	k; Kintamasis, kuriame saugomas pasirinktu pazymiu skaicius.
int	w; Kintamasis, kuriame saugomas vartotojo pasirinkimas apie egzamino rezultata.

Atvejis 3: Generuoja studentu vardus, pavardes ir pazymius. Si dalis leidzia automatiskai generuoti studentu duomenis su atsitiktiniais pazymiais. Vartotojui atsitiktinai bus generuojamas studeno vardas ir pavarde kol nebus pasirinkta stabdyti generavima. Vartotojui leidziama pasirinkti, ar generuoti atsitiktinius ar pasirinktus pazymius. Vartotojui leidziama pasirinkti, ar generuoti daugiau studentu duomenu. Atvaizduojami studentu rezultatai ekrane.

Parameters

P	Vektorius, i kuri pridedami studentu duomenys.
Pazymiai	C; Studento objektas, i kuri laikinai saugomi studento duomenys.
string	xx, yy; Laikini kintamieji, kuriuose saugomi studento vardas ir pavarde.
x	Laikinas kintamasis, kuriame saugomas vartotojo atsakymas apie generavimo buda.
string	y; Laikinas kintamasis, kuriame saugomas vartotojo atsakymas apie generavimo tesima.
int	h = 0; Skaiciuoja pazymiu kieki.
int	k; Kintamasis, kuriame saugomas pasirinktu pazymiu skaicius.
int	w; Kintamasis, kuriame saugomas vartotojo pasirinkimas apie egzamino rezultata.
int	qq=0; Indikuoja, ar norima testi studentu generavima. Jei qq == 1, baigiama studentu generavimo procedura.
int	egg; Saugomas gzamino rezultatas, jei pasirinkta ivesti ranka.
string	vardai[] = {"Paulina", "Adriana", "Gitanas", "Donald", "Ugne", "Kamile", "Rugile", "Roberta", "Valdemaras", "Jurgis"}; Masyvas, kuriame saugomi studentu vardai, is kuriu bus atsitiktinai generuojami duomenys.
string	pavardes[] = {"Podgaiska", "Obama", "Trump", "Nauseda", "Sirokyte", "Mockute", "Zobelaite", "Macaite", "Jurpalyte", "Jankauskas"}; Masyvas, kuriame saugomos studentu pavardes, is kuriu bus atsitiktinai generuojami duomenys.

Atvejis 4: Nuskaito duomenis is pasirinktu failu. Vartotojui leidziama pasirinkti, is kurio failo nuskaityti duomenis. Nustatomas pasirinktas failo pavadinimas pagal vartotojo pasirinkima. Vartotojui leidziama pasirinkti, kiek studentu

duomenu nuskaityti is failo. Vartotojui leidziama pasirinkti, kaip surikiuoti studentus. Vartotojui leidziama pasirinkti, kur isvesti surikiuotus duomenis.

Parameters

P	Vektorius, i kuri pridedami studentu duomenys.
Pazymiai	C; Studento objektas, i kuri laikinai saugomi generuojami duomenys.
int	z; Laikinas kintamasis pazymiui saugoti.
int	o; Vartotojo pasirinkimas, kur isvesti rezultatus (ekranas ar failas).
int	stud; Vartotojo pasirinkimas, kiek studentu duomenu nuskaityti is failo.
string	zodziai; Laikinas kintamasis pirmos eilutes su pavadinimais nuskaitymui.
int	xyz; Vartotojo pasirinkimas, kaip rusiuoti studentus.
int	pv = 0; Kintamasis saugantis nuskaitytu pazymiu kieki.
int	numeris; Kintamasis saugantis vartotojo pasirinkta failo numeri.
string	wp; Kintamasis saugantis failo pavadinima.
string	xx, yy; Laikini kintamieji studento vardo ir pavardes saugojimui.
int	egg; Laikinas kintamasis egzamino rezultatui saugoti.

Atvejis 5: Generuoja failus su atsitiktiniais duomenimis duomenimis. Generuojami failai yra skirtingu dydziu(1000, 10 000, 100 000, 1 000 000, 10 000 000).

Parameters

Pazymiai	C; Studento objektas, i kuri laikinai saugomi generuojami duomenys.

Atvejis 6: Sugeneruotu failu skaitymas ir apdorojimas. Vartotojas pasirenka strategijas, pagal kurias failai bus skaiciuojami. Rezultatai isspausdinami i failus.

Parameters

Pazymiai	C; Studento objektas, i kuri laikinai saugomi generuojami duomenys.
S	Vektorius, i kuri pridedami visu studentu duomenys.
P	Vektorius, i kuri pridedami studentu duomenys, kuriu vidurkis didesnis nei 5.
Z	Vektorius, i kuri pridedami studentu duomenys, kuriu vidurkis mazesnis uz 5.
int	strategy; Pasirinkta strategija, pagal kuria vykdomas failu apdorojimas.
const	vector<string> filenames = {"1k.txt", "10k.txt", "100k.txt", "1kk.txt", "10kk.txt"}; Vektorius, saugantis failu pavadinimus, kurie bus apdorojami.
const	vector<string> lopukaiFilenames = {"lopukai1.txt", "lopukai2.txt", "lopukai3.txt", "lopukai4.txt", "lopukai5.txt"}; Vektorius, saugantis failu pavadinimus, kuriuose bus isvedami rezultatai.
const	vector<string> saunuoliukaiFilenames = {"saunuoliukai1.txt", "saunuoliukai2.txt", "saunuoliukai3.txt", "saunuoliukai4.txt", "saunuoliukai5.txt"}; Vektorius, saugantis failu pavadinimus, kuriuose bus isvedami rezultatai.

Atvejis 7: Klasiu testavimas. Iskvieciamas funkcijos testai() ir ekrane matome testu rezultata.

Atvejis 8: Programos pabaiga

Returns

Grazinamas nulis, nurodantis, kad programa sekmingai baige darba

5.2 C:/Users/Paulina/Documents/Objektinis programavimas 1 kursas/vectorcontainer/pagalbinesvector.cpp File Reference

Funkciju deklaracija.

```
#include "pagalbinesvector.h"
```

Functions

- void rezultatai (std::vector< Pazymiai > hh)

  Spausdina studentu rezultatus i konsole.
- void rezultataifailas (std::vector< Pazymiai > hh, std::string failiukas)

  Spausdina studentu rezultatus i faila.
- double mediana (int u, Pazymiai h)

  Skaiciuoja mediana.
- void generuojam (int studentusk, std::string failopav)

  Generuoja atsitiktinius studentu duomenis ir iraso i faila.
- void failuskaick (std::string wp, Pazymiai hi, vector< Pazymiai > &P, vector< Pazymiai > &Z)

  Nuskaito duomenis is failo ir apdoroja juos.
- void failuskaickstrategija1 (string wp, Pazymiai hi, std::vector< Pazymiai > &S, std::vector< Pazymiai > &P, std::vector< Pazymiai > &Z)

  Nuskaito duomenis is failo ir apdoroja juos, taikant strategija 1.
- void failuskaickstrategija2 (string wp, Pazymiai hi, std::vector< Pazymiai > &P, std::vector< Pazymiai > &Z)

  Funkcija nuskaito duomenis is failo, apdoroja juos ir isskirsto i du konteinerius.
- void failuskaickstrategija3 (string wp, Pazymiai hi, std::vector< Pazymiai > &S, std::vector< Pazymiai > &P, std::vector< Pazymiai > &Z)

  Funkcija nuskaito duomenis is failo, apdoroja juos ir isskirsto i tris konteinerius.
- void spausdintuvas (std::string zekai, std::string malaciai, vector< Pazymiai > P, vector< Pazymiai > Z)

  Funkcija isveda studentu rezultatus i du atskirus failus.
- void **testai** ()

  Funkcija skirta programos testavimui.

5.2.1 Detailed Description

Funkciju deklaracija.

5.2.2 Function Documentation

5.2.2.1 failuskaick()

```
void failuskaick (
            std::string wp,
            Pazymiai hi,
            vector< Pazymiai > & P,
            vector< Pazymiai > & Z )
```

Nuskaito duomenis is failo ir apdoroja juos.

Parameters

wp	Failo pavadinimas
hi	Pazymiai objektas
P	Studentai, kuriu galutinis pazymys >= 5
Z	Studentai, kuriu galutinis pazymys < 5

5.2.2.2 failuskaickstrategija1()

```
void failuskaickstrategija1 (
            string wp,
            Pazymiai hi,
            std::vector< Pazymiai > & S,
            std::vector< Pazymiai > & P,
            std::vector< Pazymiai > & Z )
```

Nuskaito duomenis is failo ir apdoroja juos, taikant strategija 1.

Parameters

wp	Failo pavadinimas
hi	Pazymiai objektas
S	Visi nuskaityti studentai
P	Studentai, kuriu galutinis pazymys >= 5
Z	Studentai, kuriu galutinis pazymys < 5

5.2.2.3 failuskaickstrategija2()

```
void failuskaickstrategija2 (
            string wp,
            Pazymiai hi,
            std::vector< Pazymiai > & P,
            std::vector< Pazymiai > & Z )
```

Funkcija nuskaito duomenis is failo, apdoroja juos ir isskirsto i du konteinerius.

Parameters

wp	Failo pavadinimas, is kurio nuskaitomi duomenys.
hi	Objektas, kuris naudojamas saugoti viena studento informacijos irasa.
P	Konteineris, kuriame saugomi studentai, kuriu galutinis rezultatas yra didesnis nei 5.
Z	Konteineris, kuriame saugomi studentai, kuriu galutinis rezultatas yra mazesnis nei 5.

5.2.2.4 failuskaickstrategija3()

```
void failuskaickstrategija3 (
            string wp,
```

```
            Pazymiai hi,
            std::vector< Pazymiai > & S,
            std::vector< Pazymiai > & P,
            std::vector< Pazymiai > & Z )
```

Funkcija nuskaito duomenis is failo, apdoroja juos ir isskirsto i tris konteinerius.

Parameters

wp	Failo pavadinimas, is kurio nuskaitomi duomenys.
hi	Objektas, kuris naudojamas saugoti viena studento informacijos irasa.
S	Konteineris, kuriame saugomi visi studentai.
P	Konteineris, kuriame saugomi studentai, kuriu galutinis rezultatas yra didesnis nei 5.
Z	Konteineris, kuriame saugomi studentai, kuriu galutinis rezultatas yra mazesnis nei 5.

5.2.2.5 generuojam()

```
void generuojam (
            int studentusk,
            std::string failopav )
```

Generuoja atsitiktinius studentu duomenis ir iraso i faila.

Parameters

studentusk	Studentu skaicius
failopav	Failo pavadinimas

5.2.2.6 mediana()

```
double mediana (
            int u,
            Pazymiai h )
```

Skaiciuoja mediana.

Parameters

u	Pazymiu skaicius
h	Pazymiai objektas

Returns

Mediana

5.2.2.7 rezultatai()

```
void rezultatai (
            std::vector< Pazymiai > hh )
```

Spausdina studentu rezultatus i konsole.

Parameters

hh	Studentu sarasas

5.2.2.8 rezultataifailas()

```
void rezultataifailas (
            std::vector< Pazymiai > hh,
            std::string failiukas )
```

Spausdina studentu rezultatus i faila.

Parameters

hh	Studentu sarasas
failiukas	Failo pavadinimas

5.2.2.9 spausdintuvas()

```
void spausdintuvas (
            std::string zekai,
            std::string malaciai,
            vector< Pazymiai > P,
            vector< Pazymiai > Z )
```

Funkcija isveda studentu rezultatus i du atskirus failus.

Parameters

zekai	Failo pavadinimas, i kuri isvedami studentai, kuriu galutinis rezultatas yra mazesnis nei 5.
malaciai	Failo pavadinimas, i kuri isvedami studentai, kuriu galutinis rezultatas yra didesnis nei 5.
P	Konteineris, kuriame saugomi studentai, kuriu galutinis rezultatas yra didesnis nei 5.
Z	Konteineris, kuriame saugomi studentai, kuriu galutinis rezultatas yra mazesnis nei 5.

5.3 C:/Users/Paulina/Documents/Objektinis programavimas 1 kursas/vectorcontainer/pagalbinesvector.h File Reference

Pazymiai ir Zmogus klasiu deklaracija ir funkciju reiksmiu priskyrimas.

```
#include <iostream>
#include <iomanip>
#include <string>
#include <cstdlib>
#include <vector>
```

```
#include <algorithm>
#include <fstream>
#include <chrono>
#include <stdexcept>
#include <cassert>
```

Classes

- class Zmogus

  Klase Zmogus saugo informacija apie studento varda ir pavarde.
- class Pazymiai

  Klase Pazymiai saugo informacija apie studentu pazymius.

Functions

- void rezultatai (std::vector< Pazymiai > hh)

  Spausdina studentu rezultatus i konsole.
- void rezultataifailas (std::vector< Pazymiai > hh, std::string failiukas)

  Spausdina studentu rezultatus i faila.
- double mediana (int u, Pazymiai h)

  Skaiciuoja mediana.
- void generuojam (int studentusk, std::string failopav)

  Generuoja atsitiktinius studentu duomenis ir iraso i faila.
- void **failuskaick** (std::string wp, Pazymiai hi, std::vector< Pazymiai > &P, std::vector< Pazymiai > &Z)
- void **failuskaickstrategija1** (std::string wp, Pazymiai hi, std::vector< Pazymiai > &S, std::vector< Pazymiai > &P, std::vector< Pazymiai > &Z)
- void **failuskaickstrategija2** (std::string wp, Pazymiai hi, std::vector< Pazymiai > &P, std::vector< Pazymiai > &Z)
- void **failuskaickstrategija3** (std::string wp, Pazymiai hi, std::vector< Pazymiai > &S, std::vector< Pazymiai > &P, std::vector< Pazymiai > &Z)
- void **spausdintuvas** (std::string zekai, std::string malaciai, std::vector< Pazymiai > P, std::vector< Pazymiai > Z)
- void **testai** ()

  Funkcija skirta programos testavimui.

5.3.1 Detailed Description

Pazymiai ir Zmogus klasiu deklaracija ir funkciju reiksmiu priskyrimas.

5.3.2 Function Documentation

5.3.2.1 generuojam()

```
void generuojam (
            int studentusk,
            std::string failopav )
```

Generuoja atsitiktinius studentu duomenis ir iraso i faila.

Parameters

studentusk	Studentu skaicius
failopav	Failo pavadinimas

5.3.2.2 mediana()

```
double mediana (
            int u,
            Pazymiai h )
```

Skaiciuoja mediana.

Parameters

u	Pazymiu skaicius
h	Pazymiai objektas

Returns

Mediana

5.3.2.3 rezultatai()

```
void rezultatai (
            std::vector< Pazymiai > hh )
```

Spausdina studentu rezultatus i konsole.

Parameters

hh	Studentu sarasas

5.3.2.4 rezultataifailas()

```
void rezultataifailas (
            std::vector< Pazymiai > hh,
            std::string failiukas )
```

Spausdina studentu rezultatus i faila.

Parameters

hh	Studentu sarasas
failiukas	Failo pavadinimas

5.4 pagalbinesvector.h

Go to the documentation of this file.

```
00001
00005 #ifndef PAGALBINESVECTOR_H_INCLUDED
00006 #define PAGALBINESVECTOR_H_INCLUDED
00007 #include <iostream>
00008 #include <iomanip>
00009 #include <string>
00010 #include <cstdlib>
00011 #include <vector>
00012 #include <algorithm>
00013 #include <fstream>
00014 #include <chrono>
00015 #include <string>
00016 #include <stdexcept>
00017 #include <cassert>
00022 class Zmogus {
00023 protected:
00024     std::string var_;
00025     std::string pav_;
00027 public:
00031     Zmogus() = default;
00032
00038     Zmogus(const std::string& var, const std::string& pav) : var_(var), pav_(pav) {}
00039
00043     virtual ~Zmogus() {}
00044
00048     virtual void setVar(const std::string& newVar) { var_ = newVar; }
00049
00053     virtual void setPav(const std::string& newPav) { pav_ = newPav; }
00054
00058     virtual std::string getVar() const = 0;
00059
00063     virtual std::string getPav() = 0;
00064 };
00065
00070 class Pazymiai : public Zmogus {
00071 private:
00072     double vid_;
00073     int egz_;
00074     std::vector<int> paz_;
00075     double galutinis_;
00076     double med_;
00078 public:
00082     Pazymiai() : vid_(0), egz_(0), galutinis_(0), med_(0) {}
00083
00094     Pazymiai(const std::string& var, const std::string& pav, double vid, int egz, const
      std::vector<int>& paz,
00095              double galutinis, double med)
00096         : Zmogus(var, pav), vid_(vid), egz_(egz), paz_(paz), galutinis_(galutinis), med_(med) {}
00097
00101     ~Pazymiai() { paz_.clear(); }
00105     Pazymiai(const Pazymiai& other)
00106         : Zmogus(other.getVar(), other.getPav()), vid_(other.vid_), egz_(other.egz_),
00107           paz_(other.paz_), galutinis_(other.galutinis_), med_(other.med_) {}
00111     Pazymiai(Pazymiai&& other) noexcept
00112         : Zmogus(std::move(other.var_), std::move(other.pav_)),
00113           vid_(other.vid_), egz_(other.egz_), paz_(std::move(other.paz_)),
00114           galutinis_(other.galutinis_), med_(other.med_) {}
00118     Pazymiai& operator=(const Pazymiai& other) {
00119         if (this != &other) {
00120             Zmogus::setVar(other.getVar());
00121             Zmogus::setPav(other.getPav());
00122             vid_ = other.vid_;
00123             egz_ = other.egz_;
00124             paz_ = other.paz_;
00125             galutinis_ = other.galutinis_;
00126             med_ = other.med_;
00127         }
00128         return *this;
00129     }
00133     Pazymiai& operator=(Pazymiai&& other) noexcept {
00134         if (this != &other) {
00135             Zmogus::setVar(std::move(other.var_));
00136             Zmogus::setPav(std::move(other.pav_));
00137             vid_ = other.vid_;
00138             egz_ = other.egz_;
00139             paz_ = std::move(other.paz_);
00140             galutinis_ = other.galutinis_;
00141             med_ = other.med_;
00142         }
00143         return *this;
00144     }
00145
```

```
00148     int getPazN(const std::vector<int>& newPaz, int pos) const { return newPaz[pos]; }
00149
00153     void clearPaz() { paz_.clear(); }
00156     void setVid(double newVid) { vid_ = newVid; }
00157
00160     void setEgz(int newEgz) { egz_ = newEgz; }
00161
00164     void setOnePaz(int newPaz) { paz_.push_back(newPaz); }
00165
00168     void setPazymiai(const std::vector<int>& pazymiai) { paz_ = pazymiai; }
00169
00172     void setGalutinis(double newGalutinis) { galutinis_ = newGalutinis; }
00173
00176     void setMed(double newMed) { med_ = newMed; }
00177
00180     void sortPaz(Pazymiai& C) { std::sort(C.paz_.begin(), C.paz_.end()); }
00181
00184     double getVid() const { return vid_; }
00185
00188     int getEgz() const { return egz_; }
00189
00192     std::vector<int> getPaz() const { return paz_; }
00193
00196     double getGalutinis() const { return galutinis_; }
00197
00200     double getMed() const { return med_; }
00201
00205     std::string getVar() const override { return var_; }
00206
00210     std::string getPav() override { return pav_; }
00211
00217     friend double mediana(int u, const Pazymiai h);
00218
00224     friend std::istream& operator»(std::istream& is, Pazymiai& obj) {
00225         std::cout « "Iveskite studento varda (noredami baigti spauskite 4):" « std::endl;
00226         is » obj.var_;
00227         if (obj.var_ == "4" || obj.pav_ == "4")
00228             return is;
00229         std::cout « "Iveskite studento pavarde (noredami baigti spauskite 4):" « std::endl;
00230         is » obj.pav_;
00231         if (obj.var_ == "4" || obj.pav_ == "4")
00232             return is;
00233
00234         double suma = 0.0;
00235         int pazymys;
00236         int j = 0;
00237
00238         do {
00239             std::cout « "Iveskite " « j + 1 « " pazymi (norint baigti spauskite 11): ";
00240             is » pazymys;
00241
00242             if (pazymys == 11)
00243                 break;
00244
00245             while (pazymys < 1 || pazymys > 10 || is.fail()) {
00246                 std::cout « "Klaida. Iveskite skaiciu nuo 1 iki 10: ";
00247                 is.clear();
00248                 is.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
00249                 is » pazymys;
00250             }
00251
00252             obj.paz_.push_back(pazymys);
00253             suma += pazymys;
00254             j++;
00255         } while (true);
00256
00257         obj.vid_ = suma / j;
00258
00259         std::cout « "Iveskite egzamino rezultata : ";
00260         is » obj.egz_;
00261
00262         while (obj.egz_ < 1 || obj.egz_ > 10 || is.fail()) {
00263             std::cout « "Klaida. Iveskite skaiciu nuo 1 iki 10: ";
00264             is.clear();
00265             is.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
00266             is » obj.egz_;
00267         }
00268
00269         obj.galutinis_ = (obj.vid_ * 0.4) + (obj.egz_ * 0.6);
00270
00271         std::sort(obj.paz_.begin(), obj.paz_.end());
00272
00273         obj.med_ = mediana(j, obj);
00274
00275         return is;
00276     }
00283     friend std::ostream& operator«(std::ostream& os, const Pazymiai& obj) {
```

```
00284         os « std::left « std::setw(15) « obj.var_ « std::setw(15) « obj.pav_ « std::setw(17)
00285            « std::fixed « std::setprecision(2) « obj.galutinis_ « std::setw(17) « std::fixed
00286            « std::setprecision(2) « obj.med_ « std::endl;
00287         return os;
00288     }
00289 };
00290
00291 void rezultatai(std::vector<Pazymiai> hh);
00292
00293 void rezultataifailas(std::vector<Pazymiai> hh, std::string failiukas);
00294
00295 double mediana(int u, Pazymiai h);
00296
00297 void generuojam(int studentusk, std::string failopav);
00298
00299 void failuskaick(std::string wp, Pazymiai hi, std::vector<Pazymiai>& P, std::vector<Pazymiai>& Z);
00300
00301 void failuskaickstrategija1(std::string wp, Pazymiai hi,std::vector<Pazymiai>& S,
      std::vector<Pazymiai>& P, std::vector<Pazymiai>& Z);
00302
00303 void failuskaickstrategija2(std::string wp, Pazymiai hi, std::vector<Pazymiai>& P,
      std::vector<Pazymiai>& Z);
00304
00305 void failuskaickstrategija3(std::string wp, Pazymiai hi,std::vector<Pazymiai>& S,
      std::vector<Pazymiai>& P, std::vector<Pazymiai>& Z);
00306
00307 void spausdintuvas(std::string zekai, std::string malaciai, std::vector<Pazymiai> P,
      std::vector<Pazymiai> Z );
00308
00309 void testai();
00310
00311 #endif // PAGALBINES_H_INCLUDED
```

Index