Studentų pažymių skaičiuoklė

Generated by Doxygen 1.12.0

Chapter 1

Struktūra:

- pradineVersija.cpp pagrindinis vykdomasis taškas;
- Mylib.h bibliotekų failas;
- Stud.h studentų struktūros bei funkcijų deklaracijų failas;
- Stud.cpp funkcijų failas.

Parametras	Virtual Machine (Windows 11)	Host Machine (MacBook)
Operacinė sistema	Windows 11	macOS Sequoia 15.1.1
CPU	4 branduoliai (priskirti)	8 branduoliai
RAM	8 GB	16 GB
Saugykla	Virtualus diskas SSD	512 GB SSD
VMware Version	VMware Fusion 13	-

1.1 Diegimas ir paleidimas

1. Atsisiųsti: Include, src, CMakeLists.txt, run.bat aplankus ir failus.
2. Atsisiųstus aplankus ir failus sudėti į vieną aplanką.
3. Paleisti run.bat failą.
4. Aplanke atsiradusiame Debug aplanke randamas .exe failas Studentu_skaiciuokle.exe paruoštas naudoti.

1.2 v0.1 release

1.2.1 Sukurtos funkcijos:

- ivestis() - duomenims įvesti ranka arba generuoti atsitiktinai;
- isvestisIsFailo() - duomenims nuskaityti iš .txt failo;
- rusiavimas() - studentų pavardėms rūšiuoti abėcėlės tvarka;
- rezultatai() - galutiniam pažymiui skaičiuoti;
- isvestis() - studentų pavardei, vardui bei galutiniam pažymiui išvesti.

1.2.2 Funkcionalumas:

- galima studento duomenų įvestis terminale arba .txt failo nuskaitymas;
- pasirinkimas tarp keleto galimų nuskaityti failų;
- galimybė įvesti pažymius ranka arba generuoti atsitiktinai;
- neapribotas namų darbų pažymių įvedimo skaičius;
- pranešimai apie įvesties klaidas ir galimybė įvesti paskutinį vestą rodmenį naujai (exeption handling).

1.3 v0.2 release

1.3.1 Naujos funkcijos:

- generuotiFaila() - generuoti atsitiktinių studentų duomenų .txt failą;
- studentoKategorija() - išskiria studentus į dvi grupes pagal galutinį pažymį;
- isvestisIFaila() - į dvi grupes padalinti studentai išvedami į .txt failą.

1.3.2 Naujas funkcionalumas:

- galimybė generuoti 5 skirtingo dydžio .txt failus su atsitiktiniais studentų duomenimis;
- studentų padalijimas į dvi grupes pagal galutinį pažymį, jų išvedimas į atskirus .txt failus;
- programos veikimo spartos analizė.

1.3.3 Spartos analizė:

Pateiktas testavimų vidurkis:

Failo eilučių	Nuskaitymas	Dalijimas į dvi grupes	Įrašymas galvociai.txt	Įrašymas nuskriaustuka txt	Bendras veikimo laikas	Įrašymas į failą
1000	0.02131653 s	0.00059168 s	0.00992062 s	0.00620507 s	0.03803327 s	0.0183877 s
10000	0.144208 s	0.00439836 s	0.0588816 s	0.04473393 s	0.25155567 s	0.1003 s
100000	1.36194 s	0.03765913 s	0.5736883 s	0.4084937 s	2.38045 s	0.8448273 s
1000000	13.5593 s	0.4072613 s	5.66804 s	4.02833 s	23.6636 s	8.14784 s
10000000	137.6157 s	6.3477 s	55.94437 s	39.38567 s	239.293 s	80.41363 s

1.4 v0.3 release

1.4.1 Naujas funkcionalumas:

- galimybė pasirinkti konteinerį duomenims saugoti (list/vector).

1.4.2 Spartos analizė:

Prie v1.0 release S1 Vector ir S1 List stulpeliai.

Išvados:

- vector konteineriuose talpinami duomenys buvo apdorojami greičiau nei list;
- list konteineriai greičiau dalijami į dvi grupes kai dirbama su mažesniais failais;
- bendru atveju spartos skirtumas tarp vector ir list konteinerių nėra didelis.

1.5 v1.0 release

Studentų galutinio pažymio skaičiavimo programa.

1.5.1 Naujos funkcijos:

- studentoKategorija1() - išskiria studentus į dvi grupes (du naujus konteinerius galvociai ir nuskriaustukai);
- studentoKategorija2() - išskiria studentus į dvi grupes (naują konteinerį nuskriaustukai ir studentų konteinerį be nuskriaustukų);
- studentoKategorija3() - išskiria studentus į dvi grupes (studentoKategorija2() pagreitinta std::partition algoritmu);
- studentoKategorijaVector() - išskiria studentus į dvi grupes (studentoKategorija2() pagreitinta std::partition algoritmu, skirta tik vector konteineriams).

1.5.2 Naujas funkcionalumas:

- galimybė pasirinkti duomenų rūšiavimo į dvi grupes strategiją.

1.5.3 Spartos analizė:

Pateiktas testavimų vidurkis:

S1 - 1 strategija, S2 - 2 strategija, S3 - 3 strategija.

Failo eilučių	Kriterijus	S1 Vector laikas	S1 List laikas	S2 Vector laikas	S2 List laikas	S3 Vector laikas	S3 List laikas	Fiksuoto Vector laikas
1000	Nuskaitymas	0.↩ 0113095 s	0.↩ 0138963 s	0.↩ 0167558 s	0.↩ 01712995 s	0.↩ 01120315 s	0.↩ 0107881 s	0.↩ 0149814 s
	Dalijimas į dvi grupes	0.↩ 00056075 s	0.↩ 000387625 s	0.↩ 000403395 s	0.↩ 000227021 s	0.↩ 000259104 s	0.↩ 000365229 s	0.↩ 000222041 s

Failo eilučių	Kriterijus	S1 Vector laikas	S1 List laikas	S2 Vector laikas	S2 List laikas	S3 Vector laikas	S3 List laikas	Fiksuoto Vector laikas
	Įrašymas galvociai.txt	0.00555825 s	0.00592483 s	0.007075 s	0.008406995 s	0.005477375 s	0.005456125 s	0.00677679 s
	Įrašymas nuskriaustukai.txt	0.0374054 s	0.00366213 s	0.004650525 s	0.00543275 s	0.003665085 s	0.00384277 s	0.00511758 s
	Bendras veikimo laikas	0.021169 s	0.0238708 s	0.02888475 s	0.0311967 s	0.0206047 s	0.02045225 s	0.0270978 s
10000	Nuskaitymas	0.10496 s	0.125007 s	0.140005 s	0.10597165 s	0.1018986 s	0.09736675 s	0.154275 s
	Dalijimas į dvi grupes	0.00449438 s	0.00265867 s	0.00214002 s	0.002315975 s	0.001532895 s	0.002043 s	0.00102333 s
	Įrašymas galvociai.txt	0.0474353 s	0.0482639 s	0.0602062 s	0.05024755 s	0.0498332 s	0.05652305 s	0.0581072 s
	Įrašymas nuskriaustukai.txt	0.0379377 s	0.0340268 s	0.04542225 s	0.04073535 s	0.0382545 s	0.03880095 s	0.0428352 s
	Bendras veikimo laikas	0.194827 s	0.209956 s	0.247773 s	0.1992705 s	0.1915195 s	0.1947335 s	0.25624 s
100000	Nuskaitymas	0.962287 s	0.960303 s	1.35193 s	0.977143 s	0.957694 s	0.94800 s	1.34642 s
	Dalijimas į dvi grupes	0.0254281 s	0.0367531 s	0.0208332 s	0.03807255 s	0.01054007 s	0.03119035 s	0.0108586 s
	Įrašymas galvociai.txt	0.496531 s	0.510438 s	0.5936945 s	0.5196565 s	0.489088 s	0.506586 s	0.57682 s
	Įrašymas nuskriaustukai.txt	0.344024 s	0.342789 s	0.4169465 s	0.360138 s	0.344018 s	0.3397615 s	0.405691 s
	Bendras veikimo laikas	1.82827 s	1.85028 s	2.383405 s	1.89501 s	1.80134 s	1.82554 s	2.33979 s
1000000	Nuskaitymas	9.57618 s	9.45731 s	14.28015 s	9.69733 s	9.75898 s	9.637725 s	14.0048 s
	Dalijimas į dvi grupes	0.498318 s	0.396237 s	0.4309075 s	1.09168 s	0.1412185 s	0.4371245 s	0.131964 s
	Įrašymas galvociai.txt	4.64819 s	4.82819 s	5.72546 s	5.000735 s	4.78187 s	4.85823 s	5.65232 s
	Įrašymas nuskriaustukai.txt	3.29679 s	3.35625 s	4.00987 s	3.469655 s	3.39245 s	3.324935 s	3.97346 s
	Bendras veikimo laikas	18.0195 s	18.038 s	24.4464 s	19.2594 s	18.0745 s	18.25805 s	23.7625 s

Failo eilučių	Kriterijus	S1 Vector laikas	S1 List laikas	S2 Vector laikas	S2 List laikas	S3 Vector laikas	S3 List laikas	Fiksuoto Vector laikas
10000000	Nuskaitymas	101.191 s	99.2413 s	147.491 s	107.877 s	99.15715 s	96.1321 s	103.119 s
	Dalijimas į dvi grupes	4.53757 s	5.69798 s	7.85074 s	17.3161 s	1.757185 s	5.74999 s	4.40295 s
	Įrašymas galvociai.↩ txt	46.4808 s	48.4736 s	59.9728 s	50.7344 s	48.2283 s	49.0712 s	48.4059 s
	Įrašymas nuskriaustukai.↩ txt	32.7055 s	33.2447 s	40.1789 s	34.7553 s	33.4136 s	32.9665 s	33.6266 s
	Bendras veikimo laikas	184.915 s	186.658 s	255.494 s	210.683 s	182.556 s	183.92 s	189.554 s

Išvados:

- 1 strategijos atveju vector konteineriuose talpinami duomenys buvo apdorojami greičiau nei list;
- 2 strategijos atveju vector konteineriuose talpinami duomenys buvo apdorojami greičiau nei list;
- list konteinerių dalijimas į dvi grupes 2 strategijos atveju vyksta lėčiau nei 1 strategijos atveju dirbant su didesniais failais;
- vector konteinerių dalijimas į dvi grupes 2 strategijos atveju vyksta lėčiau nei 1 strategijos atveju dirbant su didesniais failais;
- 2 strategijos tobulinimas naudojant std::partition algoritmą leido pasiekti efektyviausią rūšiavimą abiems konteineriams (3 strategija);
- 3 strategijos atveju konteinerių dalijimas į dvi grupes vyksta greičiau nei 2 strategijos atveju, todėl darbas 3 strategijos atveju atliekamas efektyviausiai;
- strategija, skirta tik vektorių apdorojimui, vektorių išskirsto į dvi grupes greičiau nei 3 (greičiausia) strategija, kai failai yra mažesni;
- pagal išskirstymo į grupes greitį strategijos išsidėsto: (lėčiausia) 2 strategija -> 1 strategija -> fiksuoto vector -> 3 strategija (greičiausia).

1.6 v1.1 release

Flag'as	Failo eilučių	Kriterijus	S3 Vector struct	S3 Vector class
O1	100000	Nuskaitymas	1.04574 s	1.05816 s
		Dalijimas į dvi grupes	0.01400595 s	0.02035045 s
		Įrašymas į galvociai.txt	0.4991065 s	0.507221 s
		Įrašymas į nuskriaustukai.txt	0.365311 s	0.369626 s
		Bendras veikimo laikas	1.92416 s	1.955355 s
	1000000	Nuskaitymas	10.02568 s	10.8941 s
		Dalijimas į dvi grupes	0.361507 s	0.1799375 s
		Įrašymas į galvociai.txt	5.40397 s	4.856025 s

Flag'as	Failo eilučių	Kriterijus	S3 Vector struct	S3 Vector class
		Įrašymas į nuskriaustukai.txt	3.55322 s	3.51353 s
		Bendras veikimo laikas	19.34435 s	19.4436 s
O2	100000	Nuskaitymas	1.34642 s	1.03305 s
		Dalijimas į dvi grupes	0.0108586 s	0.01878665 s
		Įrašymas į galvociai.txt	0.57682 s	0.506004 s
		Įrašymas į nuskriaustukai.txt	0.405691 s	0.3537565 s
		Bendras veikimo laikas	2.33979 s	1.911595 s
	1000000	Nuskaitymas	14.0048 s	10.5644 s
		Dalijimas į dvi grupes	0.131964 s	0.181962 s
		Įrašymas į galvociai.txt	5.65232 s	4.762955 s
		Įrašymas į nuskriaustukai.txt	3.97346 s	3.46802 s
		Bendras veikimo laikas	23.7625 s	18.97735 s
O3	100000	Nuskaitymas	1.35413 s	1.057755 s
		Dalijimas į dvi grupes	0.0134315 s	0.01895225 s
		Įrašymas į galvociai.txt	0.5870425 s	0.496599 s
		Įrašymas į nuskriaustukai.txt	0.441253 s	0.3546995 s
		Bendras veikimo laikas	2.395855 s	1.928005 s
	1000000	Nuskaitymas	13.8041 s	10.53145 s
		Dalijimas į dvi grupes	0.1978815 s	0.1855985 s
		Įrašymas į galvociai.txt	5.665575 s	4.69908 s
		Įrašymas į nuskriaustukai.txt	3.91933 s	3.39186 s
		Bendras veikimo laikas	23.5869 s	18.80795 s

.exe failo dydis pagal flag'ą:

Flag'as	struct	class
O1	90 KB	91 KB
O2	99 KB	104 KB
O3	100 KB	105 KB

1.7 v1.2 release

1.7.1 Perdengti metodai

Duomenų įvestis:

1. Rankiniu būdu: atskiriant tarpais vedamas vardas, pavardė, pasirinktas skaičius namų darbų įvertinimų (skalėje nuo 1 iki 10), egzamino įvertinimas (skalėje nuo 1 iki 10).

2. Automatiniu būdu: rankiniu būdu įvedamas vardas ir pavardė, o automatiškai sugeneruojami penkių namų darbų įvertinimai ir egzamino įvertinimas.

3. Nuskaitymas iš failo: iš pasirinkto failo nuskaitomas vardas, pavardė ir visi įvertinimai. Paskutinis įvertinimas priskiriamas egzaminui.

Duomenų išvestis:

1. Į ekraną:

- jei duomenys vesti rankiniu/automatiniu būdu, pateikiama lentelė su pavarde, vardu, galutiniu įvertinimu (įvertinimas skaičiuotas pagal pasirinktą rodiklį - vidurkį arba medianą);

- jei duomenys vesti nuskaitant iš failo, pateikiama lentelė su pavarde, vardu, galutiniu įvertinimu pagal vidurkį, galutiniu įvertinimu pagal medianą.

1. Į failą: duomenys grupuojami į nuskriaustukai.txt (studentai, kurių galutinis įvertinimas $<$ 5), ir galvociai.txt (studentai, kurių galutinis įvertinimas $>$= 5). Duomenys failuose vaizduojami taip pat, kaip išvedimo į ekraną metu.

1.7.2 Operatoriai

Įvesties operatorius:

- nuskaitant failą pirmoji eilutė ignoruojama (antraštinė eilutė);

- nuskaitomas vardas;
- nuskaitoma pavardė, jei jos nėra, imama default reikšmė (tuščia vieta);
- nuskaitomi visi įvertinimai, jei tokių nėra, namų darbų ir egzamino įvertinimui priskiriamas 0;
- jei yra nors vienas įvertinimas, paskutinis jų priskiriamas egzamino įvertinimui;
- suskaičiuojamos galutinių įvertinimų reikšmės.

Išvesties operatorius:

- išveda pavardę, vardą;

- duomenys išvedami tais pačiais duomenų išvesties principais;

- jei buvo pasirinktas Rule of Three testavimas, išvedami pradiniai studento duomenys, demonstruojamas kopijavimas ir priskyrimas, studentas po destruktoriaus panaudojimo.

1.7.3 Programos spartos analizė

O2 flag'as, trečia strategija, testai atlikti su vektoriais.

Failo eilučių	Kriterijus	Be perdengimo	Su perdengimu
100000	Nuskaitymas	1.03305 s	0.984773 s
	Dalijimas į dvi grupes	0.01878665 s	0.011197 s
	Įrašymas į galvociai.txt	0.506004 s	0.488464 s
	Įrašymas į nuskriaustukai.txt	0.3537565 s	0.34143 s
	Bendras veikimo laikas	1.911595 s	1.82586 s
1000000	Nuskaitymas	10.5644 s	9.82788 s
	Dalijimas į dvi grupes	0.181962 s	0.100198 s
	Įrašymas į galvociai.txt	4.762955 s	4.70759 s
	Įrašymas į nuskriaustukai.txt	3.46802 s	3.32145 s
	Bendras veikimo laikas	18.97735 s	17.9571 s

Išvados:

- programa su perdengtais metodais greičiau nuskaito duomenis;
- programa su perdengtais metodais greičiau įrašo duomenis į failus;
- programa su perdengtais metodais bendru atveju veikia greičiau ir sklandžiau.

1.8 v1.5 release

1.8.1 Klasių atnaujinimai:

Zmogus:

- abstrakti bazinė klasė;
- aprašomas vardas ir pavardė;
- naudojamas konstruktorius;
- naudojami get/set metodai;
- virtuali funkcija klase().

Studentas:

- išvestinė klasė iš Zmogus klasės;
- aprašomi namų darbai, egzaminas, galutiniai pažymiai;

- demonstruojamas polimorfizmas su funkcija klase();
- naudojami konstruktoriai;
- naudojami get/set metodai;
- implementuota trijų metodų taisyklė;
- išvesties/įvesties operatoriai.

Kadang klasė Zmogus yra abstrakti, jos objektų kurti negalima:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

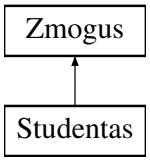
Class Documentation

5.1 Studentas Class Reference

Išvestinė klasė iš klasės Zmogus studentui aprašyti.

`#include <Stud.h>`

Inheritance diagram for Studentas:



Public Member Functions

- vector< int > nd () const

  Studento namų darbų įvertinimų Getter'is.
- int egzaminas () const

  Studento egzamino įvertinimo Getter'is.
- double galutinisPazymys () const

  Studento galutinio pažymio Getter'is.
- double galutinisPazymysVid () const

  Studento galutinio pažymio pagal vidurkį Getter'is.
- double galutinisPazymysMed () const

  Studento galutinio pažymio pagal medianą Getter'is.
- void setNd (const vector< int > &nd)

  Studento namų darbų įvertinimo Setter'is.
- void setEgzaminas (int egzaminas)

  Studento egzamino įvertinimo Setter'is.
- void setGalutinisPazymys (double galutinisPazymys)

  Studento galutinio pažymio Setter'is.
- void setGalutinisPazymysVid (double galutinisPazymysVid)

  Studento galutinio pažymio pagal vidurkį Setter'is.
- void setGalutinisPazymysMed (double galutinisPazymysMed)

Studento galutinio pažymio pagal medianą Setter'is.

- Studentas ()

  Konstruktorius su numatytomis parametrų reikšmėmis.
- Studentas (const string &vardas, const string &pavarde, const vector< int > &nd, int egzaminas)

  Konstruktorius su parametrais, kuris inicializuoja Studentas objektą su pateiktais duomenimis.
- Studentas (const Studentas &other)

  Kopijavimo konstruktorius, kuris sukuria kito Studentas objekto kopiją.
- Studentas & operator= (const Studentas &other)

  Priskyrimo operatorius, kuris priskiria vieną Studentas objektą kitam.
- ~Studentas ()

  Destruktorius.
- void klase () const override

  Išveda informaciją apie klasę, kuriai priklauso objektas.
- void ivestis (bool generavimas)

  Įveda studento duomenis rankinio įvedimo atveju.
- double rezultatai (const string &pasirinkimas)

  Apskaičiuoja galutinį studento pažymį pagal pasirinktą rodiklį.

Public Member Functions inherited from Zmogus

- Zmogus (const string &vardas="", const string &pavarde="")

  Konstruktorius su numatytomis parametrų reikšmėmis.
- virtual ~Zmogus ()=default

  Virtualus destruktorius.
- string vardas () const

  Žmogaus vardo Getter'is.
- string pavarde () const

  Žmogaus pavardės Getter'is.
- void setVardas (const string &vardas)

  Žmogaus vardo Setter'is.
- void setPavarde (const string &pavarde)

  Žmogaus pavardės Setter'is.

Friends

- istream & operator>> (istream &in, Studentas &studentas)

  Įvesties operatorius, skirtas klasės Studentas objekto duomenims nuskaityti.
- ostream & operator<< (ostream &out, const Studentas &studentas)

  Išvesties operatorius, skirtas klasės Studentas objekto duomenims išvesti.
- void isvestis (const Studentas &Lok, int ivestiesPasirinkimas)

  Išveda studento duomenis į ekraną. Priklausomai nuo duomenų įvesties, išvedami duomenys apie studentą su arba be objekto saugojimo adreso.
- bool rusiavimas (const Studentas &pavarde1, const Studentas &pavarde2)

  Lygina du Studentas objektus pagal galutinį pažymį ir pavardę.

Additional Inherited Members

Protected Attributes inherited from Zmogus

- string vardas_

  Žmogaus vardas.
- string pavarde_

  Žmogaus pavardė

5.1.1 Detailed Description

Išvestinė klasė iš klasės Zmogus studentui aprašyti.

Implementuota trijų metodų taisyklė (Rule of Three). Apibrėžti įvesties/išvesties operatoriai.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Studentas() [1/3]

```
Studentas::Studentas ()  [inline]
```

Konstruktorius su numatytomis parametrų reikšmėmis.

Parameters

vardas	Tuščia eilutė (atributas paveldimas iš tėvinės klasės Zmogus).
pavarde	Tuščia eilutė (atributas paveldimas iš tėvinės klasės Zmogus).
egzaminas	Reikšmė 0.
galutinisPazymys	Reikšmė 0.
galutinisPazymysVid	Reikšmė 0.
galutinisPazymysMed	Reikšmė 0.

5.1.2.2 Studentas() [2/3]

```
Studentas::Studentas (
            const string & vardas,
            const string & pavarde,
            const vector< int > & nd,
            int egzaminas)  [inline]
```

Konstruktorius su parametrais, kuris inicializuoja Studentas objektą su pateiktais duomenimis.

Parameters

vardas	Žmogaus vardas string tipo.
pavarde	Žmogaus pavardė string tipo.
nd	Namų darbų įvertinimai vector tipo konteineryje.
egzaminas	Egzamino įvertinimas int tipo.

5.1.2.3 Studentas() [3/3]

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

Kopijavimo konstruktorius, kuris sukuria kito Studentas objekto kopiją.

Parameters

other	Klasės Studentas objektas, kurio duomenys bus nukopijuoti.

5.1.2.4 ~Studentas()

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

Destruktorius.

5.1.3 Member Function Documentation

5.1.3.1 egzaminas()

```
int Studentas::egzaminas () const  [inline]
```

Studento egzamino įvertinimo Getter'is.

Returns

Grąžina int tipo egzamino įvertinimą.

5.1.3.2 galutinisPazymys()

```
double Studentas::galutinisPazymys () const  [inline]
```

Studento galutinio pažymio Getter'is.

Returns

Grąžina double tipo galutinį pažymį.

5.1.3.3 galutinisPazymysMed()

```
double Studentas::galutinisPazymysMed () const  [inline]
```

Studento galutinio pažymio pagal medianą Getter'is.

Returns

Grąžina double tipo galutinį pažymį pagal medianą.

5.1.3.4 galutinisPazymysVid()

```
double Studentas::galutinisPazymysVid () const  [inline]
```

Studento galutinio pažymio pagal vidurkį Getter'is.

Returns

Grąžina double tipo galutinį pažymį pagal vidurkį.

5.1.3.5 ivestis()

```
void Studentas::ivestis (
            bool generavimas)
```

Įveda studento duomenis rankinio įvedimo atveju.

Funkcija leidžia įvesti studento vardą, pavardę, namų darbų ir egzamino įvertinimus. Jei pasirinktas automatinis duomenų generavimas (generavimas == true), tada funkcija sugeneruoja 4 atsitiktinius namų darbų ir egzamino pažymius. Jei generavimas yra false, funkcija leidžia vartotojui įvesti duomenis rankiniu būdu.

Parameters

generavimas	Jei `true`, sugeneruojami atsitiktiniai namų darbų ir egzamino pažymiai. Jei `false`, prašoma vartotojo įvesti duomenis rankiniu būdu.

5.1.3.6 klase()

```
void Studentas::klase () const  [inline], [override], [virtual]
```

Išveda informaciją apie klasę, kuriai priklauso objektas.

Funkcija išveda tekstą, kad objektas priklauso "Studentas" klasei.

Implements Zmogus.

5.1.3.7 nd()

```
vector< int > Studentas::nd () const  [inline]
```

Studento namų darbų įvertinimų Getter'is.

Returns

Grąžina vector tipo namų darbų konteinerį.

5.1.3.8 operator=()

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

Priskyrimo operatorius, kuris priskiria vieną Studentas objektą kitam.

Parameters

other	Klasės Studentas objektas, kurio reikšmės bus priskirtos šiam objektui.

Returns

Studentas& Grąžina nuorodą į objektą.

5.1.3.9 rezultatai()

```
double Studentas::rezultatai (
            const string & pasirinkimas)
```

Apskaičiuoja galutinį studento pažymį pagal pasirinktą rodiklį.

Rodikliai: vidurkis ("Vid") ir mediana ("Med"). Abu rodikliai skaičiuojami pagal namų darbus.

Parameters

pasirinkimas	Pasirinktas metodas pažymio apskaičiavimui: • "Vid" – vidurkis. • "Med" – mediana.

Returns

Apskaičiuotas galutinis pažymys, atsižvelgiant į pasirinkimą:

- 0.4 ∗ namų darbų vidurkis + 0.6 ∗ egzaminas (jei pasirinktas "Vid").
- 0.4 ∗ namų darbų medianą + 0.6 ∗ egzaminas (jei pasirinktas "Med").
- 0, jei nėra namų darbų ir egzaminas neįvestas.

5.1.3.10 setEgzaminas()

```
void Studentas::setEgzaminas (
            int egzaminas)  [inline]
```

Studento egzamino įvertinimo Setter'is.

Parameters

egzaminas	Naujas įvestas egzamino įvertinimas.

5.1.3.11 setGalutinisPazymys()

```
void Studentas::setGalutinisPazymys (
            double galutinisPazymys)  [inline]
```

Studento galutinio pažymio Setter'is.

Parameters

galutinisPazymys	Naujas apskaičiuotas galutinis pažymys.

5.1.3.12 setGalutinisPazymysMed()

```
void Studentas::setGalutinisPazymysMed (
            double galutinisPazymysMed)  [inline]
```

Studento galutinio pažymio pagal medianą Setter'is.

Parameters

galutinisPazymysMed	Naujas apskaičiuotas galutinis pažymys pagal medianą.

5.1.3.13 setGalutinisPazymysVid()

```
void Studentas::setGalutinisPazymysVid (
            double galutinisPazymysVid)  [inline]
```

Studento galutinio pažymio pagal vidurkį Setter'is.

Parameters

galutinisPazymysVid	Naujas apskaičiuotas galutinis pažymys pagal vidurkį.

5.1.3.14 setNd()

```
void Studentas::setNd (
            const vector< int > & nd)  [inline]
```

Studento namų darbų įvertinimo Setter'is.

Parameters

egzaminas	Nauji įvesti namų darbų įvertinimai.

5.1.4 Friends And Related Symbol Documentation

5.1.4.1 isvestis

```
void isvestis (
            const Studentas & Lok,
            int ivestiesPasirinkimas)  [friend]
```

Išveda studento duomenis į ekraną. Priklausomai nuo duomenų įvesties, išvedami duomenys apie studentą su arba be objekto saugojimo adreso.

Parameters

Lok	Studentas, kurio duomenys bus išvedami.
ivestiesPasirinkimas	Pasirinkimas, nurodantis, kokius duomenis išvesti: • 1: Išveda studento duomenis ir objekto saugojimo adresą. • 2: Išveda tik studento duomenis.

5.1.4.2 operator<<

```
ostream & operator<< (
            ostream & out,
            const Studentas & studentas)  [friend]
```

Išvesties operatorius, skirtas klasės Studentas objekto duomenims išvesti.

Duomenys išvedami į ekraną arba į .txt failus.

Parameters

out	Išvesties srautas, kuriuo bus atspausdinta studento informacija.
studentas	Studentas, kurio duomenys bus išvedami.

Returns

Grąžina išvesties srautą.

5.1.4.3 operator>>

```
istream & operator>> (
            istream & in,
            Studentas & studentas)  [friend]
```

Įvesties operatorius, skirtas klasės Studentas objekto duomenims nuskaityti.

Operatorius nuskaitymo metu apdoroja studento vardą, pavardę, namų darbų įvertinimus ir egzamino įvertinimą. Taip pat nustato galutinius pažymius.

Parameters

in	Įvedimo srautas, iš kurio bus nuskaityti duomenys.
studentas	Klasės Studentas objekto nuoroda, kur bus įrašyti duomenys.

Returns

Grąžina įvesties srautą.

5.1.4.4 rusiavimas

```
bool rusiavimas (
            const Studentas & pavarde1,
            const Studentas & pavarde2)  [friend]
```

Lygina du Studentas objektus pagal galutinį pažymį ir pavardę.

Funkcija palygina du klasės Studentas objektus pagal jų galutinį pažymį. Jei pažymiai vienodi, tuomet lyginamos pavardės pagal abėcėlę. Palyginimo sąlygą galima nustatyti pagal kintamąjį `rikiavimoSalyga` (pažymiai bus rikiuojami didėjimo/mažėjimo tvarka).

Parameters

pavarde1	Pirmas Studentas objektas, su kuriuo lyginamas antras.
pavarde2	Antras Studentas objektas, su kuriuo lyginamas pirmas.

Returns

`true`, jei pirmo objekto pažymys yra mažesnis/didesnis (pagal nustatytą rikiavimo sąlygą); `false`, jei pirmo objekto pažymys nėra mažesnis/didesnis (pagal nustatytą rikiavimo sąlygą).

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

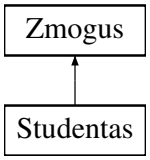
- Include/Stud.h
- src/Stud.cpp

5.2 Zmogus Class Reference

Abstrakti klasė žmogui aprašyti.

`#include <Stud.h>`

Inheritance diagram for Zmogus:



Public Member Functions

- Zmogus (const string &vardas="", const string &pavarde="")
  Konstruktorius su numatytomis parametrų reikšmėmis.
- virtual ~Zmogus ()=default
  Virtualus destruktorius.
- string vardas () const
  Žmogaus vardo Getter'is.
- string pavarde () const
  Žmogaus pavardės Getter'is.
- void setVardas (const string &vardas)
  Žmogaus vardo Setter'is.
- void setPavarde (const string &pavarde)
  Žmogaus pavardės Setter'is.
- virtual void klase () const =0
  Abstrakti funkcija, skirta klasei identifikuoti.

Protected Attributes

- string vardas_

  Žmogaus vardas.
- string pavarde_

  Žmogaus pavardė

5.2.1 Detailed Description

Abstrakti klasė žmogui aprašyti.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Zmogus()

```
Zmogus::Zmogus (
            const string & vardas = "",
            const string & pavarde = "")  [inline]
```

Konstruktorius su numatytomis parametrų reikšmėmis.

Parameters

vardas	Tuščias string.
pavarde	Tuščias string.

5.2.2.2 ~Zmogus()

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

Virtualus destruktorius.

5.2.3 Member Function Documentation

5.2.3.1 klase()

```
virtual void Zmogus::klase () const  [pure virtual]
```

Abstrakti funkcija, skirta klasei identifikuoti.

Funkcija turi būti realizuota išvestinėse klasėse objekto klasės tipui nustatyti.

Implemented in Studentas.

5.2.3.2 pavarde()

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

Žmogaus pavardės Getter'is.

Returns

Grąžina string tipo žmogaus pavardę.

5.2.3.3 setPavarde()

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

Žmogaus pavardės Setter'is.

Parameters

pavarde	Nauja nustatyta žmogaus pavardė.

5.2.3.4 setVardas()

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

Žmogaus vardo Setter'is.

Parameters

vardas	Naujas nustatytas žmogaus vardas.

5.2.3.5 vardas()

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

Žmogaus vardo Getter'is.

Returns

Grąžina string tipo žmogaus vardą.

5.2.4 Member Data Documentation

5.2.4.1 pavarde_

```
string Zmogus::pavarde_  [protected]
```

Žmogaus pavardė

5.2.4.2 vardas_

```
string Zmogus::vardas_  [protected]
```

Žmogaus vardas.

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

- Include/Stud.h

Chapter 6

File Documentation

6.1 Include/Mylib.h File Reference

Bibliotekų header failas.

```
#include <iostream>
#include <string>
#include <vector>
#include <iomanip>
#include <algorithm>
#include <fstream>
#include <sstream>
#include <cstdlib>
#include <ctime>
#include <stdexcept>
#include <chrono>
#include <random>
#include <limits>
#include <list>
#include <functional>
#include <iterator>
```

6.1.1 Detailed Description

Bibliotekų header failas.

Header failas apima įvairias standartines bibliotekas ir pagerina kodo skaitomumą, naudojant dažnai reikalingus elementus.

6.2 Mylib.h

Go to the documentation of this file.

```
00001
00008 #pragma once
00009
00010 #include <iostream>
00011 #include <string>
00012 #include <vector>
00013 #include <iomanip>
00014 #include <algorithm>
00015 #include <fstream>
00016 #include <sstream>
00017 #include <cstdlib>
00018 #include <ctime>
00019 #include <stdexcept>
00020 #include <chrono>
00021 #include <random>
00022 #include <limits>
00023 #include <list>
00024 #include <functional>
00025 #include <iterator>
00026
00027 using std::cout;
00028 using std::cin;
00029 using std::endl;
00030 using std::string;
00031 using std::vector;
00032 using std::numeric_limits;
00033 using std::streamsize;
00034 using std::setprecision;
00035 using std::setw;
00036 using std::left;
00037 using std::right;
00038 using std::fixed;
00039 using std::sort;
00040 using std::ifstream;
00041 using std::ofstream;
00042 using std::getline;
00043 using std::istringstream;
00044 using std::invalid_argument;
00045 using std::out_of_range;
00046 using std::runtime_error;
00047 using std::cerr;
00048 using std::exception;
00049 using std::to_string;
00050 using std::chrono::high_resolution_clock;
00051 using std::chrono::duration;
00052 using std::ostringstream;
00053 using std::random_device;
00054 using std::mt19937;
00055 using std::uniform_int_distribution;
00056 using std::exit;
00057 using std::list;
00058 using std::conditional_t;
00059 using std::partition;
00060 using std::function;
00061 using std::copy_if;
00062 using std::remove_if;
00063 using std::back_inserter;
00064 using std::is_same_v;
00065 using std::remove_copy_if;
00066 using std::find_if;
00067 using std::back_inserter;
00068 using std::is_same;
00069 using std::istream;
00070 using std::ostream;
```

6.3 Include/Stud.h File Reference

Header failas su apibrėžtomis Zmogus ir Studentas klasėmis bei funkcijų deklaracijomis.

```
#include "Mylib.h"
```

Classes

- class Zmogus

  Abstrakti klasė žmogui aprašyti.
- class Studentas

  Išvestinė klasė iš klasės Zmogus studentui aprašyti.

Functions

- template<typename Struktura >
  int ivestisIsFailo (const string &failas, Struktura &struktura)

  Nuskaito duomenis iš .txt failo į struktūrą (vector/list).
- bool rusiavimas (const Studentas &pavarde1, const Studentas &pavarde2)

  Lygina du Studentas objektus pagal galutinį pažymį ir pavardę.
- void generuotiFaila (int studentuSkaicius, const string &failoPavadinimas)

  Generuoja studentų duomenų failą su atsitiktiniais namų darbų ir egzamino įvertinimais.
- template<typename Struktura >
  void studentoKategorija1 (const Struktura &struktura, int duomenuIvedimoBudas, Struktura &galvociai, Struktura &nuskriaustukai)

  Padalina studentus į du konteinerius: 'galvociai' (galutinis įvertinimas >= 5) ir 'nuskriaustukai' (galutinis įvertinimas < 5).
- template<typename Struktura >
  void studentoKategorija2 (Struktura &struktura, int duomenuIvedimoBudas, Struktura &nuskriaustukai)

  Rūšiuoja studentus pagal jų pažymius ir perkelia blogus studentus į 'nuskriaustukai' konteinerį (vector/list).
- template<typename Struktura >
  void studentoKategorija3 (Struktura &struktura, int duomenuIvedimoBudas, Struktura &nuskriaustukai)

  Padalina studentus į dvi kategorijas pagal studentoKategorija2() funkciją, naudojant partition metodą.
- void studentoKategorijaVector (vector< Studentas > &struktura, int duomenuIvedimoBudas, vector< Studentas > &nuskriaustukai)

  studentoKategorija3() funkcija, pritaikyta išskirtinai tik vector tipo konteineriams.
- template<typename Struktura >
  void isvestisIFaila (const Struktura &galvociai, const Struktura &nuskriaustukai, int duomenuIvedimoBudas, string pazymioTipas, duration< double > &trukmeGalvociu, duration< double > &trukmeNuskriaustuku)

  Pagalbinė duomenų išvedimo į failus (galvociai.txt/nuskriaustukai.txt) funkcija.

Variables

- int rikiavimoSalyga

  Rikiavimo sąlyga.
- int duomenuIvedimoBudas

  Duomenų įvedimo būdas.

6.3.1 Detailed Description

Header failas su apibrėžtomis Zmogus ir Studentas klasėmis bei funkcijų deklaracijomis.

Faile implementuotas "Mylib.h" bibliotekų failas, apibrėžti globalieji kintamieji, abstrakti klasė Zmogus ir jos išvestinė klasė Studentas. Taip pat yra funckijų deklaracija.

6.3.2 Function Documentation

6.3.2.1 generuotiFaila()

```
void generuotiFaila (
            int studentuSkaicius,
            const string & failoPavadinimas)
```

Generuoja studentų duomenų failą su atsitiktiniais namų darbų ir egzamino įvertinimais.

Ši funkcija sukuria failą su studentų vardais, pavardėmis, atsitiktiniais 5 namų darbų įvertinimais ir egzamino įvertinimu.

Parameters

studentuSkaicius	Kiek studentų bus generuojama.
failoPavadinimas	Failo pavadinimas, kuriame bus saugomi sugeneruotų studentų duomenys.

Exceptions

runtime_error	Jei failo nepavyksta sukurti.

6.3.2.2 isvestisIFaila()

```
template<typename Struktura >
void isvestisIFaila (
            const Struktura & galvociai,
            const Struktura & nuskriaustukai,
            int duomenuIvedimoBudas,
            string pazymioTipas,
            duration< double > & trukmeGalvociu,
            duration< double > & trukmeNuskriaustuku)
```

Pagalbinė duomenų išvedimo į failus (galvociai.txt/nuskriaustukai.txt) funkcija.

Parameters

galvociai	Konteineris studentų, kurių galutiniai pažymiai yra >= 5.
nuskriaustukai	Konteineris studentų, kurių galutiniai pažymiai yra < 5.
duomenuIvedimoBudas	Nurodo, kaip bus išvedami duoemys (priklausomai jei buvo įvesti ranka ar nuskaityti iš failo).
pazymioTipas	Galutinio pažymio tipas ("Vid."/"Med.").
trukmeGalvociu	Laikas, per kurį duomenys buvo įrašyti į "galvociai.txt" failą.
trukmeNuskriaustuku	Laikas, per kurį duomenys buvo įrašyti į "nuskriaustukai.txt" failą.

6.3.2.3 ivestisIsFailo()

```
template<typename Struktura >
int ivestisIsFailo (
            const string & failas,
            Struktura & struktura)
```

Nuskaito duomenis iš .txt failo į struktūrą (vector/list).

Parameters

failas	Failo pavadinimas, iš kurio bus nuskaitomi duomenys.
struktura	Struktūra, į kurią bus įrašyti duomenys (vector/list).

Returns

Grąžina 0, jei duomenys sėkmingai įrašyti į struktūrą, arba -1 kitu atveju.

6.3.2.4 rusiavimas()

```
bool rusiavimas (
            const Studentas & pavarde1,
            const Studentas & pavarde2)
```

Lygina du Studentas objektus pagal galutinį pažymį ir pavardę.

Funkcija palygina du klasės Studentas objektus pagal jų galutinį pažymį. Jei pažymiai vienodi, tuomet lyginamos pavardės pagal abėcėlę. Palyginimo sąlygą galima nustatyti pagal kintamąjį `rikiavimoSalyga` (pažymiai bus rikiuojami didėjimo/mažėjimo tvarka).

Parameters

pavarde1	Pirmas Studentas objektas, su kuriuo lyginamas antras.
pavarde2	Antras Studentas objektas, su kuriuo lyginamas pirmas.

Returns

`true`, jei pirmo objekto pažymys yra mažesnis/didesnis (pagal nustatytą rikiavimo sąlygą); `false`, jei pirmo objekto pažymys nėra mažesnis/didesnis (pagal nustatytą rikiavimo sąlygą).

6.3.2.5 studentoKategorija1()

```
template<typename Struktura >
void studentoKategorija1 (
            const Struktura & struktura,
            int duomenuIvedimoBudas,
            Struktura & galvociai,
            Struktura & nuskriaustukai)
```

Padalina studentus į du konteinerius: 'galvociai' (galutinis įvertinimas $>=$ 5) ir 'nuskriaustukai' (galutinis įvertinimas $<$ 5).

Funkcija padalina studentus į du konteinerius (vector/list) pagal jų galutinį pažymį (vidurkinį arba medianinį), priklausomai nuo įvesto parametro.

Parameters

struktura	Studentų konteineris, kurį reikia apdoroti.
duomenuIvedimoBudas	Duomenų įvedimo metodas, nustatantis, ar turi būti naudojamas "Vid" ar "Med" įvertinimas.
galvociai	Konteineris, kuriame bus laikomi geriausi studentai.
nuskriaustukai	Konteineris, kuriame bus laikomi blogiausi studentai.

6.3.2.6 studentoKategorija2()

```
template<typename Struktura >
void studentoKategorija2 (
            Struktura & struktura,
            int duomenuIvedimoBudas,
            Struktura & nuskriaustukai)
```

Rūšiuoja studentus pagal jų pažymius ir perkelia blogus studentus į 'nuskriaustukai' konteinerį (vector/list).

Funkcija rūšiuoja studentus pagal jų galutinį pažymį (vidurkinį arba medianinį), priklausomai nuo įvesto parametro. Po rūšiavimo studentai su mažesniu nei 5 pažymiu perkeliami į 'nuskriaustukai' konteinerį ir pašalinami iš bendro 'struktura' konteinerio.

Parameters

struktura	Studentų konteineris, kurį reikia apdoroti.
duomenuIvedimoBudas	Duomenų įvedimo metodas, nustatantis, ar turi būti naudojamas "Vid" ar "Med" įvertinimas.
nuskriaustukai	Konteineris, kuriame bus laikomi blogiausi studentai.

6.3.2.7 studentoKategorija3()

```
template<typename Struktura >
void studentoKategorija3 (
            Struktura & struktura,
            int duomenuIvedimoBudas,
            Struktura & nuskriaustukai)
```

Padalina studentus į dvi kategorijas pagal studentoKategorija2() funkciją, naudojant partition metodą.

Funkcija padalina studentus į dvi grupes: tuos, kurie atitinka galutinio įvertinimo sąlygą (>= 5) ir tuos, kurie ne. Blogi studentai (su pažymiu < 5) perkeliami į 'nuskriaustukai' konteinerį ir ištrinami iš bendro 'struktura' konteinerio.

Parameters

struktura	Studentų konteineris, kurį reikia apdoroti.
duomenuIvedimoBudas	Duomenų įvedimo metodas, nustatantis, ar turi būti naudojamas "Vid" ar "Med" įvertinimas.
nuskriaustukai	Konteineris, kuriame bus laikomi blogiausi studentai.

6.3.2.8 studentoKategorijaVector()

```
void studentoKategorijaVector (
            vector< Studentas > & struktura,
            int duomenuIvedimoBudas,
            vector< Studentas > & nuskriaustukai)
```

studentoKategorija3() funkcija, pritaikyta išskirtinai tik vector tipo konteineriams.

Parameters

struktura	Studentų vektorius, kurį reikia apdoroti.
duomenuIvedimoBudas	Duomenų įvedimo metodas, nustatantis, ar turi būti naudojamas "Vid" ar "Med" įvertinimas.
nuskriaustukai	vektorius, kuriame bus laikomi blogiausi studentai.

6.3.3 Variable Documentation

6.3.3.1 duomenuIvedimoBudas

```
int duomenuIvedimoBudas  [extern]
```

Duomenų įvedimo būdas.

Kintamasis naudojamas nustatyti, kokiu būdu bus įvesti duomenys ar kaip jais bus manipuliuojama: Įvestis ranka, nuskaitymas iš failo, failų generavimas, Rule of Three demonstracija.

Duomenų įvedimo būdas.

6.3.3.2 rikiavimoSalyga

```
int rikiavimoSalyga  [extern]
```

Rikiavimo sąlyga.

Kintamasis naudojamas nustatyti, kokia tvarka bus rikiuojami išvadami duomenys (pagal galutinį pažymį)↩ : didėjančia ar mažėjančia.

Rikiavimo sąlyga.

6.4 Stud.h

Go to the documentation of this file.

```
00001
00008 #pragma once
00009 #include "Mylib.h"
00010
00016 extern int rikiavimoSalyga;
00017
00024 extern int duomenuIvedimoBudas;
00025
00030 class Zmogus {
00031 protected:
00032     string vardas_;
00033     string pavarde_;
00034
00035 public:
00041     Zmogus(const string& vardas = "", const string& pavarde = "")
00042         : vardas_(vardas), pavarde_(pavarde) {
00043     }
00044
00046     virtual ~Zmogus() = default;
00047
00052     string vardas() const { return vardas_; }
00053
00058     string pavarde() const { return pavarde_; }
00059
```

```
00064     void setVardas(const string& vardas) { vardas_ = vardas; }
00065
00070     void setPavarde(const string& pavarde) { pavarde_ = pavarde; }
00071
00077     virtual void klase() const = 0;
00078 };
00079
00087 class Studentas : public Zmogus {
00088 private:
00089     vector<int> nd_;
00090     int egzaminas_;
00091     double galutinisPazymys_;
00092     double galutinisPazymysVid_;
00093     double galutinisPazymysMed_;
00094
00095 public:
00096
00101     inline vector<int> nd() const { return nd_; }
00102
00107     inline int egzaminas() const { return egzaminas_; }
00108
00113     inline double galutinisPazymys() const { return galutinisPazymys_; }
00114
00119     inline double galutinisPazymysVid() const { return galutinisPazymysVid_; }
00120
00125     inline double galutinisPazymysMed() const { return galutinisPazymysMed_; }
00126
00127
00132     inline void setNd(const vector<int>& nd) { nd_ = nd; }
00133
00138     inline void setEgzaminas(int egzaminas) { egzaminas_ = egzaminas; }
00139
00144     inline void setGalutinisPazymys(double galutinisPazymys) { galutinisPazymys_ = galutinisPazymys; }
00145
00150     inline void setGalutinisPazymysVid(double galutinisPazymysVid) { galutinisPazymysVid_ =
      galutinisPazymysVid; }
00151
00156     inline void setGalutinisPazymysMed(double galutinisPazymysMed) { galutinisPazymysMed_ =
      galutinisPazymysMed; }
00157
00158
00168     Studentas()
00169         : Zmogus("", ""), egzaminas_(0),
00170         galutinisPazymys_(0), galutinisPazymysVid_(0), galutinisPazymysMed_(0) {}
00171
00179     Studentas(const string& vardas, const string& pavarde, const vector<int>& nd, int egzaminas)
00180         : Zmogus(vardas, pavarde), nd_(nd), egzaminas_(egzaminas) {
00181         galutinisPazymysVid_ = rezultatai("Vid");
00182         galutinisPazymysMed_ = rezultatai("Med");
00183     }
00184
00185
00190     Studentas(const Studentas& other)
00191         : Zmogus(other), nd_(other.nd_), egzaminas_(other.egzaminas_),
00192         galutinisPazymys_(other.galutinisPazymys_),
00193         galutinisPazymysVid_(other.galutinisPazymysVid_),
00194         galutinisPazymysMed_(other.galutinisPazymysMed_) {}
00195
00196
00202     Studentas& operator=(const Studentas& other) {
00203         if (this != &other) {
00204             vardas_ = other.vardas_;
00205             pavarde_ = other.pavarde_;
00206             nd_ = other.nd_;
00207             egzaminas_ = other.egzaminas_;
00208             galutinisPazymys_ = other.galutinisPazymys_;
00209             galutinisPazymysVid_ = other.galutinisPazymysVid_;
00210             galutinisPazymysMed_ = other.galutinisPazymysMed_;
00211         }
00212         return *this;
00213     }
00214
00216     ~Studentas() {
00217         nd_.clear();
00218         egzaminas_ = 0;
00219     }
00220
00221
00232     friend istream& operator»(istream& in, Studentas& studentas) {
00233         string line;
00234         getline(in, line);
00235         if (line.empty()) return in;
00236
00237         istringstream iss(line);
00238         iss » studentas.vardas_;
00239         if (!(iss » studentas.pavarde_)) studentas.pavarde_ = "";
00240         studentas.nd_.clear();
```

```
00241
00242         int pazymys;
00243         while (iss » pazymys) {
00244             studentas.nd_.push_back(pazymys);
00245         }
00246
00247         if (!studentas.nd_.empty()) {
00248             studentas.egzaminas_ = studentas.nd_.back();
00249             studentas.nd_.pop_back();
00250         }
00251         else {
00252             studentas.egzaminas_ = 0;
00253         }
00254
00255         if (studentas.nd_.empty() && studentas.egzaminas_ != 0) {
00256             studentas.nd_.push_back(0);
00257         }
00258
00259         studentas.galutinisPazymysVid_ = studentas.rezultatai("Vid");
00260         studentas.galutinisPazymysMed_ = studentas.rezultatai("Med");
00261
00262         return in;
00263     }
00264
00265
00275     friend ostream& operator«(ostream& out, const Studentas& studentas) {
00276         out « setw(15) « left « studentas.pavarde_
00277             « setw(15) « left « studentas.vardas_;
00278
00279         if (duomenuIvedimoBudas == 1) {
00280             out « setw(20) « left « fixed « setprecision(2) « studentas.galutinisPazymys_;
00281         }
00282         else if (duomenuIvedimoBudas == 2) {
00283             out « setw(20) « left « fixed « setprecision(2) « studentas.galutinisPazymysVid_
00284                 « setw(10) « left « fixed « setprecision(2) « studentas.galutinisPazymysMed_;
00285         }
00286         else if (duomenuIvedimoBudas == 4) {
00287             for (const auto& nd : studentas.nd_) {
00288                 out « nd « " ";
00289             }
00290             out « studentas.egzaminas_ « " ";
00291         }
00292
00293         return out;
00294     }
00295
00296
00302     void klase() const override { cout « "Studentas klase\n"; }
00303
00304     void ivestis(bool generavimas);
00305     friend void isvestis(const Studentas& Lok, int ivestiesPasirinkimas);
00306     double rezultatai(const string& pasirinkimas);
00307     friend bool rusiavimas(const Studentas & pavarde1, const Studentas & pavarde2);
00308
00309 };
00310
00311 template <typename Struktura>
00312 int ivestisIsFailo(const string& failas, Struktura& struktura);
00313 bool rusiavimas(const Studentas& pavarde1, const Studentas& pavarde2);
00314 void generuotiFaila(int studentuSkaicius, const string& failoPavadinimas);
00315 template <typename Struktura>
00316 void studentoKategorija1(const Struktura& struktura, int duomenuIvedimoBudas, Struktura& galvociai,
      Struktura& nuskriaustukai);
00317 template <typename Struktura>
00318 void studentoKategorija2(Struktura& struktura, int duomenuIvedimoBudas, Struktura& nuskriaustukai);
00319 template <typename Struktura>
00320 void studentoKategorija3(Struktura& struktura, int duomenuIvedimoBudas, Struktura& nuskriaustukai);
00321 void studentoKategorijaVector(vector<Studentas>& struktura, int duomenuIvedimoBudas,
      vector<Studentas>& nuskriaustukai);
00322 template <typename Struktura>
00323 void isvestisIFaila(const Struktura& galvociai, const Struktura& nuskriaustukai, int
      duomenuIvedimoBudas, string pazymioTipas, duration<double>& trukmeGalvociu, duration<double>&
      trukmeNuskriaustuku);
```

6.5 README.md File Reference

6.6 src/pradineVersija.cpp File Reference

Pagrindinis vykdomasis failas.

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

Functions

- int main ()

  Programos pagrindinė vykdomoji funkcija.

Variables

- int rikiavimoSalyga = 0

  Globalus kintamasis naudojamas nustatyti, kokia tvarka bus rikiuojami išvadami duomenys (pagal galutinį pažymį): didėjančia ar mažėjančia.
- int duomenuIvedimoBudas = 0

  Globalus kintamasis naudojamas nustatyti, kokiu būdu bus įvesti duomenys ar kaip jais bus manipuliuojama: įvestis ranka, nuskaitymas iš failo, failų generavimas, Rule of Three demonstracija.

6.6.1 Detailed Description

Pagrindinis vykdomasis failas.

Vartotojui leidžiama pasirinkti, kaip bus įvedami studentų duomenys, kokiuose konteineriuose jie bus saugomi, kokia tvarka rikiuojami duomenys, kokia strategija rūšiuojami į failus, pasirenkama išvestis. Taip pat programa atlieka veikimo spartos analizę, yra klaidų valdymas.

6.6.2 Function Documentation

6.6.2.1 main()

```
int main ()
```

Programos pagrindinė vykdomoji funkcija.

Funkcija apdoroja vartotojo pasirinkimus, tokius kaip duomenų įvedimo būdas, konteinerio tipas, rikiavimo/rūšiavimo strategija, išvedimas.

Programos struktūra priklauso nuo vartotojo pasirinkimų, todėl gali būti atliekami įvairūs veiksmai:

- Duomenų įvedimas rankiniu būdu arba iš failo;
- Generavimas ir studentų rūšiavimas pagal strategijas;
- Programos spartos analizė arba duomenų išvestis į ekraną.

Laikinas Studentas klasės objektas.

Studentų skaičius.

Pasirinkimas galutinį pažymį skaičiuoti pagal "Vid." (vidurkį) ar "Med." (medianą) rankinio įvedimo metu.

Pasirinkimas, ar įvertinimai bus generuojami atsitiktinai, ar vedami ranka.

pasirinkimasGeneravimo bool tipo.

Pasirenkamas failas nuskaityti.

Generuojamų failų dydžiai (kiek studentų bus generuojama).

Laiko kintamieji spartos analizei

Pasirinkimas išvesti į ekraną surikiuotus studentus su apskaičiuotais galutiniais įvertinimais ar spartos analizę.

Pasirinkimas, kokiame konteineryje (vector/list) bus saugomi duomenys.

Pasirinkimas, kokia strategija bus naudojama studentams padalinti į galvočius ir nuskriaustukus.

Galimų nuskaityti failų vektorius.

Pasirinkimas vektoriuje saugomus duomenis rūšiuoti į dvi grupes naudojant universalią ar tik vektoriams pritaikytą funkciją.

Vartotojo pasirinkimas ką daryti su duomenimis

Vartotojo pasirinkimas, kokį konteinerį naudoti duomenims saugoti

Vartotojo pasirinkimas, kokią duomenų skirstymo į dvi grupes (galvočiai ir nuskriaustukai) strategiją naudoti

Vartotojo pasirinkimas, ar dirbant su vektoriumi duomenis į dvi grupes dalinti universalia ar tik vektoriui pritaikyta funkcija

Kintamasis nustatyti, ar toliau bus naudojamas vector (jei 'true'), ar list (jei 'false') konteineris.

Vartotojo pasirinkimas, kokia tvarka rikiuoti studentus pagal pažymį (didėjančia/mažėjančia)

Vartotojo pasirinkimas į ekraną išvesti studentų rezultatus ar spartos analizės rezultatus

Vartotojo pasirinkimas pagal kokį rodiklį (vidurkį/medianą) skaičiuoti galutinį pažymį

Vartotojo įvedamas studentų, kuriuos norės įvesti, skaičius

Vartotojo pasirinkimas studento namų darbų ir egzamino įvertinimus vesti ranka ar sugeneruoti automatiškai

Vartotojo pasirinkimas, kurio failo duomenis nuskaityti

Failų generavimas

Atliekama Rule of Three ir klasių (ne)veikimo demonstracija

vector konteineryje saugomi studentai išrikiuojami, išrūšiuojami pagal galutinį įvertinimą į dvi grupes ir išvedami į du failus 'galvociai.txt' ir 'nuskriaustukai.txt'

list konteineryje saugomi studentai išrikiuojami, išrūšiuojami pagal galutinį įvertinimą į dvi grupes ir išvedami į du failus 'galvociai.txt' ir 'nuskriaustukai.txt'

Studentų apdorotų duomenų išvedimas į ekraną priklausomai nuo duomenų įvedimo būdo (įvedus ranka papildomai rodomas objekto saugojimo adresas)

Į ekraną išvedami programos veikimo spartos analizės rezultatai

6.6.3 Variable Documentation

6.6.3.1 duomenuIvedimoBudas

```
int duomenuIvedimoBudas = 0
```

Globalus kintamasis naudojamas nustatyti, kokiu būdu bus įvesti duomenys ar kaip jais bus manipuliuojama: įvestis ranka, nuskaitymas iš failo, failų generavimas, Rule of Three demonstracija.

Duomenų įvedimo būdas.

6.6.3.2 rikiavimoSalyga

```
int rikiavimoSalyga = 0
```

Globalus kintamasis naudojamas nustatyti, kokia tvarka bus rikiuojami išvadami duomenys (pagal galutinį pažymį): didėjančia ar mažėjančia.

Rikiavimo sąlyga.

6.7 src/Stud.cpp File Reference

Failas su funkcijomis. Faile implementuotas "Stud.h" header failas, kuriame yra funkcijų deklaracijos ir reikalingos klasės.

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

Functions

- void isvestis (const Studentas &Lok, int ivestiesPasirinkimas)

  Išveda studento duomenis į ekraną. Priklausomai nuo duomenų įvesties, išvedami duomenys apie studentą su arba be objekto saugojimo adreso.
- template<typename Struktura >
  int ivestisIsFailo (const string &failas, Struktura &struktura)

  Nuskaito duomenis iš .txt failo į struktūrą (vector/list).
- bool rusiavimas (const Studentas &pavarde1, const Studentas &pavarde2)

  Lygina du Studentas objektus pagal galutinį pažymį ir pavardę.
- void generuotiFaila (int studentuSkaicius, const string &failoPavadinimas)

  Generuoja studentų duomenų failą su atsitiktiniais namų darbų ir egzamino įvertinimais.
- template<typename Struktura >
  void studentoKategorija1 (const Struktura &struktura, int duomenuIvedimoBudas, Struktura &galvociai, Struktura &nuskriaustukai)

  Padalina studentus į du konteinerius: 'galvociai' (galutinis įvertinimas >= 5) ir 'nuskriaustukai' (galutinis įvertinimas < 5).
- template<typename Struktura >
  void studentoKategorija2 (Struktura &struktura, int duomenuIvedimoBudas, Struktura &nuskriaustukai)

  Rūšiuoja studentus pagal jų pažymius ir perkelia blogus studentus į 'nuskriaustukai' konteinerį (vector/list).
- template<typename Struktura >
  void studentoKategorija3 (Struktura &struktura, int duomenuIvedimoBudas, Struktura &nuskriaustukai)

Padalina studentus į dvi kategorijas pagal studentoKategorija2() funkciją, naudojant partition metodą.

- void studentoKategorijaVector (vector< Studentas > &struktura, int duomenuIvedimoBudas, vector< Studentas > &nuskriaustukai)

  studentoKategorija3() funkcija, pritaikyta išskirtinai tik vector tipo konteineriams.

- template<typename Struktura >
  void iFaila (const string &failas, const Struktura &studentai, int duomenuIvedimoBudas, const string &pazymioTipas, duration< double > &trukme)

  Įrašo studentų duomenis į failą.

- template<typename Struktura >
  void isvestisIFaila (const Struktura &galvociai, const Struktura &nuskriaustukai, int duomenuIvedimoBudas, string pazymioTipas, duration< double > &trukmeGalvociu, duration< double > &trukmeNuskriaustuku)

  Pagalbinė duomenų išvedimo į failus (galvociai.txt/nuskriaustukai.txt) funkcija.

- template int ivestisIsFailo< vector< Studentas > > (const string &failas, vector< Studentas > &struktura)
- template int ivestisIsFailo< list< Studentas > > (const string &failas, list< Studentas > &struktura)
- template void studentoKategorija1< vector< Studentas > > (const vector< Studentas > &struktura, int duomenuIvedimoBudas, vector< Studentas > &galvociai, vector< Studentas > &nuskriaustukai)
- template void studentoKategorija1< list< Studentas > > (const list< Studentas > &struktura, int duomenuIvedimoBudas, list< Studentas > &galvociai, list< Studentas > &nuskriaustukai)
- template void studentoKategorija2< vector< Studentas > > (vector< Studentas > &struktura, int duomenuIvedimoBudas, vector< Studentas > &nuskriaustukai)
- template void studentoKategorija2< list< Studentas > > (list< Studentas > &struktura, int duomenuIvedimoBudas, list< Studentas > &nuskriaustukai)
- template void studentoKategorija3< vector< Studentas > > (vector< Studentas > &struktura, int duomenuIvedimoBudas, vector< Studentas > &nuskriaustukai)
- template void studentoKategorija3< list< Studentas > > (list< Studentas > &struktura, int duomenuIvedimoBudas, list< Studentas > &nuskriaustukai)
- template void isvestisIFaila< vector< Studentas > > (const vector< Studentas > &galvociai, const vector< Studentas > &nuskriaustukai, int duomenuIvedimoBudas, string pazymioTipas, duration< double > &trukmeGalvociu, duration< double > &trukmeNuskriaustuku)
- template void isvestisIFaila< list< Studentas > > (const list< Studentas > &galvociai, const list< Studentas > &nuskriaustukai, int duomenuIvedimoBudas, string pazymioTipas, duration< double > &trukmeGalvociu, duration< double > &trukmeNuskriaustuku)

6.7.1 Detailed Description

Failas su funkcijomis. Faile implementuotas "Stud.h" header failas, kuriame yra funkcijų deklaracijos ir reikalingos klasės.

6.7.2 Function Documentation

6.7.2.1 generuotiFaila()

```
void generuotiFaila (
            int studentuSkaicius,
            const string & failoPavadinimas)
```

Generuoja studentų duomenų failą su atsitiktiniais namų darbų ir egzamino įvertinimais.

Ši funkcija sukuria failą su studentų vardais, pavardėmis, atsitiktiniais 5 namų darbų įvertinimais ir egzamino įvertinimu.

Parameters

studentuSkaicius	Kiek studentų bus generuojama.
failoPavadinimas	Failo pavadinimas, kuriame bus saugomi sugeneruotų studentų duomenys.

Exceptions

runtime_error	Jei failo nepavyksta sukurti.

6.7.2.2 iFaila()

```
template<typename Struktura >
void iFaila (
             const string & failas,
             const Struktura & studentai,
             int duomenuIvedimoBudas,
             const string & pazymioTipas,
             duration< double > & trukme)
```

Įrašo studentų duomenis į failą.

Funkcija išveda studentų duomenis į nurodytą failą (galvociai.txt/nuskriaustukai.txt). Funkcija taip pat matuoja duomenų įrašymo į failą trukmę.

Parameters

failas	Failo pavadinimas, į kurį bus įrašyti duomenys.
studentai	Studentų struktūra, kurios duomenis norima įrašyti.
duomenuIvedimoBudas	Nurodo, kaip bus išvedami duoemys (priklausomai jei buvo įvesti ranka ar nuskaityti iš failo).
pazymioTipas	Galutinio pažymio tipas ("Vid."/"Med.").
trukme	Laikas, per kurį buvo atliktas įrašymas į failą.

Exceptions

runtime_error	Jei failo nepavyko sukurti.

6.7.2.3 isvestis()

```
void isvestis (
             const Studentas & Lok,
             int ivestiesPasirinkimas)
```

Išveda studento duomenis į ekraną. Priklausomai nuo duomenų įvesties, išvedami duomenys apie studentą su arba be objekto saugojimo adreso.

Parameters

Lok	Studentas, kurio duomenys bus išvedami.
ivestiesPasirinkimas	Pasirinkimas, nurodantis, kokius duomenis išvesti: • 1: Išveda studento duomenis ir objekto saugojimo adresą. • 2: Išveda tik studento duomenis.

6.7.2.4 isvestisIFaila()

```
template<typename Struktura >
void isvestisIFaila (
            const Struktura & galvociai,
            const Struktura & nuskriaustukai,
            int duomenuIvedimoBudas,
            string pazymioTipas,
            duration< double > & trukmeGalvociu,
            duration< double > & trukmeNuskriaustuku)
```

Pagalbinė duomenų išvedimo į failus (galvociai.txt/nuskriaustukai.txt) funkcija.

Parameters

galvociai	Konteineris studentų, kurių galutiniai pažymiai yra >= 5.
nuskriaustukai	Konteineris studentų, kurių galutiniai pažymiai yra < 5.
duomenuIvedimoBudas	Nurodo, kaip bus išvedami duoemys (priklausomai jei buvo įvesti ranka ar nuskaityti iš failo).
pazymioTipas	Galutinio pažymio tipas ("Vid."/"Med.").
trukmeGalvociu	Laikas, per kurį duomenys buvo įrašyti į "galvociai.txt" failą.
trukmeNuskriaustuku	Laikas, per kurį duomenys buvo įrašyti į "nuskriaustukai.txt" failą.

6.7.2.5 isvestisIFaila< list< Studentas > >()

```
template void isvestisIFaila< list< Studentas > > (
            const list< Studentas > & galvociai,
            const list< Studentas > & nuskriaustukai,
            int duomenuIvedimoBudas,
            string pazymioTipas,
            duration< double > & trukmeGalvociu,
            duration< double > & trukmeNuskriaustuku)
```

6.7.2.6 isvestisIFaila< vector< Studentas > >()

```
template void isvestisIFaila< vector< Studentas > > (
            const vector< Studentas > & galvociai,
            const vector< Studentas > & nuskriaustukai,
            int duomenuIvedimoBudas,
            string pazymioTipas,
            duration< double > & trukmeGalvociu,
            duration< double > & trukmeNuskriaustuku)
```

6.7.2.7 ivestisIsFailo()

```
template<typename Struktura >
int ivestisIsFailo (
            const string & failas,
            Struktura & struktura)
```

Nuskaito duomenis iš .txt failo į struktūrą (vector/list).

Parameters

failas	Failo pavadinimas, iš kurio bus nuskaitomi duomenys.
struktura	Struktūra, į kurią bus įrašyti duomenys (vector/list).

Returns

Grąžina 0, jei duomenys sėkmingai įrašyti į struktūrą, arba -1 kitu atveju.

6.7.2.8 ivestisIsFailo< list< Studentas > >()

```
template int ivestisIsFailo< list< Studentas > > (
            const string & failas,
            list< Studentas > & struktura)
```

6.7.2.9 ivestisIsFailo< vector< Studentas > >()

```
template int ivestisIsFailo< vector< Studentas > > (
            const string & failas,
            vector< Studentas > & struktura)
```

6.7.2.10 rusiavimas()

```
bool rusiavimas (
            const Studentas & pavarde1,
            const Studentas & pavarde2)
```

Lygina du Studentas objektus pagal galutinį pažymį ir pavardę.

Funkcija palygina du klasės Studentas objektus pagal jų galutinį pažymį. Jei pažymiai vienodi, tuomet lyginamos pavardės pagal abėcėlę. Palyginimo sąlygą galima nustatyti pagal kintamąjį `rikiavimoSalyga` (pažymiai bus rikiuojami didėjimo/mažėjimo tvarka).

Parameters

pavarde1	Pirmas Studentas objektas, su kuriuo lyginamas antras.
pavarde2	Antras Studentas objektas, su kuriuo lyginamas pirmas.

Returns

`true`, jei pirmo objekto pažymys yra mažesnis/didesnis (pagal nustatytą rikiavimo sąlygą); `false`, jei pirmo objekto pažymys nėra mažesnis/didesnis (pagal nustatytą rikiavimo sąlygą).

6.7.2.11 studentoKategorija1()

```
template<typename Struktura >
void studentoKategorija1 (
            const Struktura & struktura,
            int duomenuIvedimoBudas,
            Struktura & galvociai,
            Struktura & nuskriaustukai)
```

Padalina studentus į du konteinerius: 'galvociai' (galutinis įvertinimas >= 5) ir 'nuskriaustukai' (galutinis įvertinimas < 5).

Funkcija padalina studentus į du konteinerius (vector/list) pagal jų galutinį pažymį (vidurkinį arba medianinį), priklausomai nuo įvesto parametro.

Parameters

struktura	Studentų konteineris, kurį reikia apdoroti.
duomenuIvedimoBudas	Duomenų įvedimo metodas, nustatantis, ar turi būti naudojamas "Vid" ar "Med" įvertinimas.
galvociai	Konteineris, kuriame bus laikomi geriausi studentai.
nuskriaustukai	Konteineris, kuriame bus laikomi blogiausi studentai.

6.7.2.12 studentoKategorija1< list< Studentas > >()

```
template void studentoKategorija1< list< Studentas > > (
            const list< Studentas > & struktura,
            int duomenuIvedimoBudas,
            list< Studentas > & galvociai,
            list< Studentas > & nuskriaustukai)
```

6.7.2.13 studentoKategorija1< vector< Studentas > >()

```
template void studentoKategorija1< vector< Studentas > > (
            const vector< Studentas > & struktura,
            int duomenuIvedimoBudas,
            vector< Studentas > & galvociai,
            vector< Studentas > & nuskriaustukai)
```

6.7.2.14 studentoKategorija2()

```
template<typename Struktura >
void studentoKategorija2 (
            Struktura & struktura,
            int duomenuIvedimoBudas,
            Struktura & nuskriaustukai)
```

Rūšiuoja studentus pagal jų pažymius ir perkelia blogus studentus į 'nuskriaustukai' konteinerį (vector/list).

Funkcija rūšiuoja studentus pagal jų galutinį pažymį (vidurkinį arba medianinį), priklausomai nuo įvesto parametro. Po rūšiavimo studentai su mažesniu nei 5 pažymiu perkeliami į 'nuskriaustukai' konteinerį ir pašalinami iš bendro 'struktura' konteinerio.

Parameters

struktura	Studentų konteineris, kurį reikia apdoroti.
duomenuIvedimoBudas	Duomenų įvedimo metodas, nustatantis, ar turi būti naudojamas "Vid" ar "Med" įvertinimas.
nuskriaustukai	Konteineris, kuriame bus laikomi blogiausi studentai.

6.7.2.15 studentoKategorija2< list< Studentas > >()

```
template void studentoKategorija2< list< Studentas > > (
            list< Studentas > & struktura,
            int duomenuIvedimoBudas,
            list< Studentas > & nuskriaustukai)
```

6.7.2.16 studentoKategorija2< vector< Studentas > >()

```
template void studentoKategorija2< vector< Studentas > > (
            vector< Studentas > & struktura,
            int duomenuIvedimoBudas,
            vector< Studentas > & nuskriaustukai)
```

6.7.2.17 studentoKategorija3()

```
template<typename Struktura >
void studentoKategorija3 (
            Struktura & struktura,
            int duomenuIvedimoBudas,
            Struktura & nuskriaustukai)
```

Padalina studentus į dvi kategorijas pagal studentoKategorija2() funkciją, naudojant partition metodą.

Funkcija padalina studentus į dvi grupes: tuos, kurie atitinka galutinio įvertinimo sąlygą (>= 5) ir tuos, kurie ne. Blogi studentai (su pažymiu < 5) perkeliami į 'nuskriaustukai' konteinerį ir ištrinami iš bendro 'struktura' konteinerio.

Parameters

struktura	Studentų konteineris, kurį reikia apdoroti.
duomenuIvedimoBudas	Duomenų įvedimo metodas, nustatantis, ar turi būti naudojamas "Vid" ar "Med" įvertinimas.
nuskriaustukai	Konteineris, kuriame bus laikomi blogiausi studentai.

6.7.2.18 studentoKategorija3< list< Studentas > >()

```
template void studentoKategorija3< list< Studentas > > (
            list< Studentas > & struktura,
            int duomenuIvedimoBudas,
            list< Studentas > & nuskriaustukai)
```

6.7.2.19 studentoKategorija3< vector< Studentas > >()

```
template void studentoKategorija3< vector< Studentas > > (
            vector< Studentas > & struktura,
            int duomenuIvedimoBudas,
            vector< Studentas > & nuskriaustukai)
```

6.7.2.20 studentoKategorijaVector()

```
void studentoKategorijaVector (
            vector< Studentas > & struktura,
            int duomenuIvedimoBudas,
            vector< Studentas > & nuskriaustukai)
```

studentoKategorija3() funkcija, pritaikyta išskirtinai tik vector tipo konteineriams.

Parameters

struktura	Studentų vektorius, kurį reikia apdoroti.
duomenuIvedimoBudas	Duomenų įvedimo metodas, nustatantis, ar turi būti naudojamas "Vid" ar "Med" įvertinimas.
nuskriaustukai	vektorius, kuriame bus laikomi blogiausi studentai.

Index