

OOP

Sugeneruota Doxygen 1.12.0

1 Hierarchijos Indeksas	1
1.1 Klasių hierarchija	1
2 Klasės Indeksas	3
2.1 Klasės	3
3 Failo Indeksas	5
3.1 Failai	5
4 Klasės Dokumentacija	7
4.1 Studentas Klasė	7
4.1.1 Smulkus aprašymas	9
4.1.2 Konstruktoriaus ir Destruktoriaus Dokumentacija	9
4.1.2.1 Studentas() [1/4]	9
4.1.2.2 Studentas() [2/4]	9
4.1.2.3 Studentas() [3/4]	9
4.1.2.4 Studentas() [4/4]	9
4.1.2.5 ~Studentas()	10
4.1.3 Metodų Dokumentacija	10
4.1.3.1 getEgzaminas()	10
4.1.3.2 getGalutinis()	10
4.1.3.3 getNamudarbai()	10
4.1.3.4 getPavarde()	11
4.1.3.5 getVardas()	11
4.1.3.6 operator=()	11
4.1.3.7 setEgzaminas()	11
4.1.3.8 setGalutinis()	12
4.1.3.9 setNamuDarbai()	12
4.1.3.10 setPavarde()	12
4.1.3.11 setVardas()	12
4.1.3.12 skaiciuotiGalutiniMediana()	13
4.1.3.13 skaiciuotiGalutiniVidurkiu()	13
4.1.4 Draugiškų Ir Susijusių Funkcijų Dokumentacija	13
4.1.4.1 operator<<	13
4.1.4.2 operator>>	13
4.2 Timer Klasė	14
4.2.1 Smulkus aprašymas	14
4.2.2 Konstruktoriaus ir Destruktoriaus Dokumentacija	14
4.2.2.1 Timer()	14
4.2.3 Metodų Dokumentacija	14
4.2.3.1 elapsed()	14
4.2.3.2 reset()	15
4.3 Zmogus Klasė	15

4.3.1 Smulkus aprašymas	16
4.3.2 Konstruktoriaus ir Destruktoriaus Dokumentacija	16
4.3.2.1 Zmogus() [1/2]	16
4.3.2.2 Zmogus() [2/2]	16
4.3.2.3 ~Zmogus()	16
4.3.3 Metodų Dokumentacija	16
4.3.3.1 getPavarde()	16
4.3.3.2 getVardas()	17
4.3.4 Atributų Dokumentacija	17
4.3.4.1 pavarde_	17
4.3.4.2 vardas_	17
5 Failo Dokumentacija	19
5.1 include/Lib.h Failo Nuoroda	19
5.2 Lib.h	19
5.3 include/Stud.h Failo Nuoroda	20
5.3.1 Funkcijos Dokumentacija	21
5.3.1.1 duomenulsvedimasPagalStrategija()	21
5.3.1.2 generuotiFaila()	22
5.3.1.3 isvedimas()	22
5.3.1.4 isvedimasIFaila()	23
5.3.1.5 ivedimas()	24
5.3.1.6 ivedimasRanka()	24
5.3.1.7 nuskaitytiFaila()	25
5.3.1.8 pasirinktiDuomenulvedima()	25
5.3.1.9 pasirinktiGalutinioskaiciavimoMetoda()	26
5.3.1.10 pasirinktiRezultatulsvedimoBuda()	26
5.3.1.11 pasirinktiRusiavimoParametra()	26
5.3.1.12 rusiuotiStudentus()	27
5.3.1.13 skaiciuotiGalutini()	27
5.3.1.14 skirstytiStudentusAntraStrategija()	28
5.3.1.15 skirstytiStudentusPirmaStrategija()	28
5.3.1.16 skirstytiStudentusTreciaStrategija()	29
5.4 Stud.h	30
5.5 include/Studentas.h Failo Nuoroda	31
5.6 Studentas.h	31
5.7 include/Timer.h Failo Nuoroda	32
5.8 Timer.h	32
5.9 include/Zmogus.h Failo Nuoroda	33
5.10 Zmogus.h	33
5.11 src/Main.cpp Failo Nuoroda	33
5.11.1 Funkcijos Dokumentacija	34

5.11.1.1 main()	34
5.12 src/Stud.cpp Failo Nuoroda	34
5.12.1 Funkcijos Dokumentacija	35
5.12.1.1 duomenusvedimasPagalStrategija()	35
5.12.1.2 duomenusvedimasPagalStrategija< list< Studentas > >()	36
5.12.1.3 duomenusvedimasPagalStrategija< vector< Studentas > >()	36
5.12.1.4 generuotiFaila()	36
5.12.1.5 isvedimas()	37
5.12.1.6 isvedimasIFaila()	37
5.12.1.7 ivedimas()	38
5.12.1.8 ivedimas< list< Studentas > >()	38
5.12.1.9 ivedimas< vector< Studentas > >()	39
5.12.1.10 ivedimasRanka()	39
5.12.1.11 ivedimasRanka< list< Studentas > >()	39
5.12.1.12 ivedimasRanka< vector< Studentas > >()	39
5.12.1.13 nuskaitytilsFailo()	40
5.12.1.14 nuskaitytilsFailo< list< Studentas > >()	40
5.12.1.15 nuskaitytilsFailo< vector< Studentas > >()	40
5.12.1.16 pasirinktiDuomenulvedima()	41
5.12.1.17 pasirinktiGalutinoskaiciavimoMetoda()	41
5.12.1.18 pasirinktiRezultatulsvedimoBuda()	41
5.12.1.19 pasirinktiRezultatulsvedimoBuda< list< Studentas > >()	42
5.12.1.20 pasirinktiRezultatulsvedimoBuda< vector< Studentas > >()	42
5.12.1.21 pasirinktiRusiavimoParametra()	42
5.12.1.22 rusiuotiStudentus()	42
5.12.1.23 skaiciuotiGalutini()	43
5.12.1.24 skirstytiStudentusAntraStrategija()	43
5.12.1.25 skirstytiStudentusPirmaStrategija()	44
5.12.1.26 skirstytiStudentusTreciaStrategija()	45
5.13 src/Studentas.cpp Failo Nuoroda	45
5.13.1 Funkcijos Dokumentacija	46
5.13.1.1 operator<<()	46
5.13.1.2 operator>>()	46
Rodyklė	47

skyrius 1

Hierarchijos Indeksas

1.1 Klasių hierarchija

Šis paveldėjimo sąrašas yra beveik surikiuotas abėcėlės tvarka:

Timer	14
Zmogus	15
Studentas	7

skyrius 2

Klasės Indeksas

2.1 Klasės

Klasės, struktūros, sąjungos ir sąsajos su trumpais aprašymais:

Studentas	Klasė, reprezentuojanti studento informaciją	7
Timer	Laiko matavimo klasė, skirta skaičiuoti programos veikimo laiką	14
Zmogus	Abstrakti bazinė klasė žmogaus duomenims saugoti	15

skyrius 3

Failo Indeksas

3.1 Failai

Visų failų sąrašas su trumpais aprašymais:

include/ Lib.h	19
include/ Stud.h	20
include/ Studentas.h	31
include/ Timer.h	32
include/ Zmogus.h	33
src/ Main.cpp	33
src/ Stud.cpp	34
src/ Studentas.cpp	45

skyrius 4

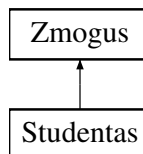
Klasės Dokumentacija

4.1 Studentas Klasė

Klasė, reprezentuojanti studento informaciją.

```
#include <Studentas.h>
```

Paveldimumo diagrama Studentas:



Vieši Metodai

- **Studentas** ()
Numatytasis konstruktorius.
- **Studentas** (const string &vardas, const string &pavarde, const vector< int > &namudarbai, double egzaminas)
Pilnas konstruktorius.
- **Studentas** (const string &vardas, const string &pavarde, int ndSkaicius)
Konstruktorius, sugeneruojantis namų darbų įvertinimus.
- **Studentas** (const **Studentas** &kitas)
Kopijavimo konstruktorius.
- **~Studentas** ()
Destruktorius.
- **Studentas** & **operator=** (const **Studentas** &kitas)
Priskyrimo operatorius.
- void **setVardas** (const string &vardas)
Nustato studento vardą.
- void **setPavarde** (const string &pavarde)
Nustato studento pavardę.
- void **setNamuDarbai** (const vector< int > &namudarbai)
Nustato studento namų darbų pažymius.

- void `setEgzaminas` (const int &egzaminas)
Nustato studento egzamino pažymį.
- void `setGalutinis` (const double &galutinis)
Nustato studento galutinį pažymį.
- vector< int > `getNamudarbai` () const
Grąžina studento namų darbų pažymius.
- int `getEgzaminas` () const
Grąžina studento egzamino pažymį.
- double `getGalutinis` () const
Grąžina studento galutinį pažymį.
- string `getVardas` () const override
Grąžina studento vardą.
- string `getPavarde` () const override
Grąžina studento pavardę.
- void `skaiciuotiGalutiniVidurkiu` ()
Skaiciuoja galutinį įvertinimą pagal vidurkį.
- void `skaiciuotiGalutiniMediana` ()
Skaiciuoja galutinį įvertinimą pagal medianą.

Vieši Metodai inherited from `Zmogus`

- virtual `~Zmogus` ()=default
Virtualus destruktorius.

Draugai

- std::ostream & `operator<<` (std::ostream &os, const `Studentas` &s)
Išvedimo operatorius.
- std::istream & `operator>>` (std::istream &is, `Studentas` &s)
Įvedimo operatorius.

Additional Inherited Members

Apsaugoti Metodai inherited from `Zmogus`

- `Zmogus` ()
Numatytasis konstruktorius.
- `Zmogus` (const string &vardas, const string &pavarde)
Konstruktorius su parametrais.

Apsaugoti Atributai inherited from `Zmogus`

- string `vardas_`
Žmogaus vardas.
- string `pavarde_`
Žmogaus pavardė.

4.1.1 Smulkus aprašymas

Klasė, reprezentuojanti studento informaciją.

Klasė paveldi iš bazinės klasės "Zmogus" ir saugo informaciją apie studento namų darbus, egzamino rezultatą bei galutinį įvertinimą.

4.1.2 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.1.2.1 Studentas() [1/4]

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

Numatytasis konstruktorius.

4.1.2.2 Studentas() [2/4]

```
Studentas::Studentas (  
    const string & vardas,  
    const string & pavarde,  
    const vector< int > & namudarbai,  
    double egzaminas)
```

Pilnas konstruktorius.

Parametrai

<i>vardas</i>	Studentų vardas.
<i>pavarde</i>	Studentų pavardė.
<i>namudarbai</i>	Namų darbų įvertinimų vektorius.
<i>egzaminas</i>	Egzamino įvertinimas.

4.1.2.3 Studentas() [3/4]

```
Studentas::Studentas (  
    const string & vardas,  
    const string & pavarde,  
    int ndSkaicius)
```

Konstruktorius, sugeneruojantis namų darbų įvertinimus.

Parametrai

<i>vardas</i>	Studentų vardas.
<i>pavarde</i>	Studentų pavardė.
<i>ndSkaicius</i>	Namų darbų skaičius, kuriam sugeneruojami įvertinimai.

4.1.2.4 Studentas() [4/4]

```
Studentas::Studentas (  
    const Studentas & kitas)
```

Kopijavimo konstruktorius.

Parametrai

<i>kitas</i>	Kitas "Studentas" klasės objektas, kurio reikšmės kopijuojamos.
--------------	-----------------------------------------------------------------

4.1.2.5 ~Studentas()

```
Studentas::~~Studentas ()
```

Destruktorius.

Atlaisvina "Studentas" klasės objektų resursus.

4.1.3 Metodų Dokumentacija**4.1.3.1 getEgzaminas()**

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

Gražina studento egzamino pažymį.

Gražina

Egzamino pažymio reikšmė.

4.1.3.2 getGalutinis()

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

Gražina studento galutinį pažymį.

Gražina

Galutinio pažymio reikšmė.

4.1.3.3 getNamudarbai()

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

Gražina studento namų darbų pažymius.

Gražina

Vektorius su namų darbų pažymiais.

4.1.3.4 getPavarde()

```
string Studentas::getPavarde () const [inline], [override], [virtual]
```

Gražina studento pavardę.

Gražina

Studentą reprezentuojanti pavardė.

Realizuoja [Zmogus](#).

4.1.3.5 getVardas()

```
string Studentas::getVardas () const [inline], [override], [virtual]
```

Gražina studento vardą.

Gražina

Studentą reprezentuojantis vardas.

Realizuoja [Zmogus](#).

4.1.3.6 operator=()

```
Studentas & Studentas::operator= (
    const Studentas & kitas)
```

Priskyrimo operatorius.

Priskiria vieno "Studentas" objekto reikšmės kitam objektui.

Parametrai

<i>kitas</i>	Kitas "Studentas" klasės objektas, kurio reikšmės priskiriamos.
--------------	-----------------------------------------------------------------

Gražina

Nuoroda į patį "Studentas" objektą.

4.1.3.7 setEgzaminas()

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

Nustato studento egzamino pažymį.

Parametrai

<i>egzaminas</i>	Egzamino pažymio reikšmė.
------------------	---------------------------

4.1.3.8 setGalutinis()

```
void Studentas::setGalutinis (  
    const double & galutinis) [inline]
```

Nustato studento galutinį pažymį.

Parametrai

<i>galutinis</i>	Galutinio pažymio reikšmė.
------------------	----------------------------

4.1.3.9 setNamuDarbai()

```
void Studentas::setNamuDarbai (  
    const vector< int > & namudarbai) [inline]
```

Nustato studento namų darbų pažymius.

Parametrai

<i>namudarbai</i>	Vektorius, kuriame yra namų darbų pažymiai.
-------------------	---------------------------------------------

4.1.3.10 setPavarde()

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

Nustato studento pavardę.

Parametrai

<i>pavarde</i>	Nauja studento pavardė.
----------------	-------------------------

4.1.3.11 setVardas()

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

Nustato studento vardą.

Parametrai

<i>vardas</i>	Naujas studento vardas.
---------------	-------------------------

4.1.3.12 skaiciuotiGalutiniMediana()

```
void Studentas::skaiciuotiGalutiniMediana ()
```

Skaiciuoja galutinį įvertinimą pagal medianą.

Jei namų darbų sąrašas yra tuščias, galutinis įvertinimas skaičiuojamas tik iš egzamino balo. Jei yra namų darbų balai, jie surūšiuojami ir apskaičiuojama mediana, kuri sudaro 40% galutinio įvertinimo, o likusieji 60% - egzamino balas.

4.1.3.13 skaiciuotiGalutiniVidurkiu()

```
void Studentas::skaiciuotiGalutiniVidurkiu ()
```

Skaiciuoja galutinį įvertinimą pagal vidurkį.

Jei namų darbų sąrašas yra tuščias, galutinis įvertinimas skaičiuojamas tik iš egzamino balo. Jei yra namų darbų balai, jie vidurkinami ir sudaromas galutinis įvertinimas, kur 40% sudaro vidurkis, o 60% - egzamino balas.

4.1.4 Draugiškų Ir Susijusių Funkcijų Dokumentacija

4.1.4.1 operator<<

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

Išvedimo operatorius.

Formatuotai išveda studento informaciją (pavardę, vardą ir galutinį įvertinimą) į srautą.

Parametrai

<i>os</i>	Išvedimo srautas (pavyzdžiui, std::cout).
<i>s</i>	<i>Studentas</i> , kurio informacija bus išvedama.

Gražina

Nuoroda į išvedimo srautą.

4.1.4.2 operator>>

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

Įvedimo operatorius.

Leidžia įvesti studento informaciją iš srauto (pavyzdžiui, std::cin ar failo). Jei įvestis yra iš konsolės, vartotojui pateikiami klausimai ir pasirinkimai apie duomenų įvedimą.

Parametrai

is	Įvedimo srautas (pavyzdžiui, std::cin).
s	Studentas , kurio duomenys bus įvedami.

Gražina

Nuoroda į įvedimo srautą.

Dokumentacija šiai klasei sugeneruota iš šių failų:

- [include/Studentas.h](#)
- [src/Studentas.cpp](#)

4.2 Timer Klasė

Laiko matavimo klasė, skirta skaičiuoti programos veikimo laiką.

```
#include <Timer.h>
```

Vieši Metodai

- [Timer](#) ()
- void [reset](#) ()
- double [elapsed](#) () const

4.2.1 Smulkus aprašymas

Laiko matavimo klasė, skirta skaičiuoti programos veikimo laiką.

Klasė [Timer](#) naudoja `high_resolution_clock`, leidžiantį tiksliai nustatyti praėjusį laiką.

4.2.2 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.2.2.1 Timer()

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

4.2.3 Metodų Dokumentacija

4.2.3.1 elapsed()

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

4.2.3.2 reset()

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

Dokumentacija šiai klasei sugeneruota iš šio failo:

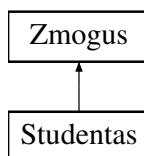
- include/Timer.h

4.3 Zmogus Klasė

Abstrakti bazinė klasė žmogaus duomenims saugoti.

```
#include <Zmogus.h>
```

Paveldimumo diagrama Zmogus:



Vieši Metodai

- virtual `~Zmogus()` = default
Virtualus destruktorius.
- virtual string `getVardas()` const = 0
Grynai virtuali funkcija žmogaus vardui gauti.
- virtual string `getPavarde()` const = 0
Grynai virtuali funkcija žmogaus pavardei gauti.

Apsaugoti Metodai

- `Zmogus()`
Numatytasis konstruktorius.
- `Zmogus(const string &vardas, const string &pavarde)`
Konstruktorius su parametrais.

Apsaugoti Atributai

- string `vardas_`
Žmogaus vardas.
- string `pavarde_`
Žmogaus pavardė.

4.3.1 Smulkus aprašymas

Abstrakti bazinė klasė žmogaus duomenims saugoti.

Klasė apibrėžia bendras savybes ir metodus, reikalingus žmogaus duomenims apdoroti, negalima sukurti žmogaus tipo objektų, o tik objektus iš jos išvestinių klasių. Klasė apibrėžia pagrindinius atributus (vardas, pavardė). Ji yra abstrakti, nes turi virtualią funkciją, kuri turi būti įgyvendinta išvestinėje klasėje.

4.3.2 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.3.2.1 Zmogus() [1/2]

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

Numatytasis konstruktorius.

Konstruktoriumi galima naudotis tik išvestinėse klasėse.

4.3.2.2 Zmogus() [2/2]

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

Konstruktorius su parametrais.

Inicializuoja vardą ir pavardę pateiktais parametrais.

Parametrai

<i>vardas</i>	Žmogaus vardas.
<i>pavarde</i>	Žmogaus pavardė.

4.3.2.3 ~Zmogus()

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

Virtualus destruktoriaus.

Naudojamas atminties atlaisvinimui paveldėtose klasėse.

4.3.3 Metodų Dokumentacija

4.3.3.1 getPavarde()

```
virtual string Zmogus::getPavarde () const [pure virtual]
```

Grynai virtuali funkcija žmogaus pavardei gauti.

Ši funkcija turi būti įgyvendinta išvestinėse klasėse.

Gražina

Žmogaus pavardė kaip string.

Realizuota [Studentas](#).

4.3.3.2 getVardas()

```
virtual string Zmogus::getVardas () const [pure virtual]
```

Grynai virtuali funkcija žmogaus vardui gauti.

Ši funkcija turi būti įgyvendinta išvestinėse klasėse.

Gražina

Žmogaus vardas kaip string.

Realizuota [Studentas](#).

4.3.4 Atributų Dokumentacija

4.3.4.1 pavarde_

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

Žmogaus pavardė.

4.3.4.2 vardas_

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

Žmogaus vardas.

Dokumentacija šiai klasei sugeneruota iš šio failo:

- include/[Zmogus.h](#)

skyrius 5

Failo Dokumentacija

5.1 include/Lib.h Failo Nuoroda

```
#include <iostream>
#include <string>
#include <vector>
#include <iomanip>
#include <algorithm>
#include <sstream>
#include <fstream>
#include <stdexcept>
#include <random>
#include <chrono>
#include <utility>
#include <list>
#include <limits>
```

5.2 Lib.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef LIB_H
00002 #define LIB_H
00003
00004 #include <iostream>
00005 #include <string>
00006 #include <vector>
00007 #include <iomanip>
00008 #include <algorithm>
00009 #include <sstream>
00010 #include <fstream>
00011 #include <stdexcept>
00012 #include <random> //random_device and uniform_int_distribution
00013 #include <chrono>
00014 #include <utility>
00015 #include <list>
00016 #include <limits>
00017 #include <stdexcept> //runtime_error
00018
00019 using std::string;
00020 using std::vector;
00021 using std::cout;
00022 using std::cin;
00023 using std::endl;
00024 using std::left;
00025 using std::setprecision;
```

```

00026 using std::fixed;
00027 using std::setw;
00028 using std::stringstream;
00029 using std::ifstream;
00030 using std::cerr;
00031 using std::ofstream;
00032 using std::invalid_argument;
00033 using std::out_of_range;
00034 using std::random_device;
00035 using std::uniform_int_distribution;
00036 using std::to_string;
00037 using namespace std::chrono;
00038 using std::move;
00039 using std::list;
00040 using std::is_same_v;
00041 using std::numeric_limits;
00042 using std::max;
00043 using std::streamsize;
00044 using std::runtime_error;
00045 using std::ostream;
00046 using std::istream;
00047
00048 #endif

```

5.3 include/Stud.h Failo Nuoroda

```

#include "Lib.h"
#include "Studentas.h"

```

Funkcijos

- `template<typename Container >`
`void ivedimas` (Container &studentai, string vardas, string pavarde)
Leidžia vartotojui įvesti studento pažymius (namų darbų pažymius ir egzamino rezultata) ranka.
- `void skaiciuotiGalutini` (Studentas &s, char pasirinkimas)
Funkcija apskaičiuoti galutinį balą studentui.
- `template<typename Container >`
`void isvedimas` (const Container &studentai, char pasirinkimas)
Funkcija išvesti studentų duomenis į terminalą.
- `template<typename Container >`
`void nuskaitytiIlsFailo` (Container &studentai)
Funkcija skaityti studentų duomenis iš failo.
- `template<typename Container >`
`void isvedimasIFaila` (const Container &studentai, char pasirinkimas, const string &failoPavadinimas)
Funkcija įrašyti studentų rezultatus į failą.
- `void generuotiFaila` (int studentuSkaicius, const string &failoPavadinimas)
Funkcija generuoti failą su atsitiktiniais studentų duomenimis.
- `template<typename Container >`
`void skirstytiStudentusPirmaStrategija` (const Container &studentai, Container &vargsiukai, Container &kie-
 tiakai)
Funkcija studentų skirstymui į dvi grupes naudojant pirmąją strategiją.
- `template<typename Container >`
`void skirstytiStudentusAntraStrategija` (Container &studentai, Container &vargsiukai)
Funkcija studentų skirstymui į dvi grupes naudojant antrąją strategiją.
- `template<typename Container >`
`void skirstytiStudentusTreciaStrategija` (Container &studentai, Container &vargsiukai)
Funkcija studentų skirstymui į dvi grupes naudojant trečiąją strategiją.

- `template<typename Container >`
`void rusiuotiStudentus` (Container &studentai, char parametras)
Funkcija studentų rūšiavimui pagal pasirinktą parametą.
- `char pasirinktiGalutinioskaiciavimoMetoda` ()
Funkcija leidžia vartotojui pasirinkti galutinio balo skaičiavimo metodą.
- `char pasirinktiRusivimoParametra` ()
Funkcija leidžia vartotojui pasirinkti studentų rūšiavimo parametą.
- `template<typename Container >`
`void pasirinktiRezultatulsvedimoBuda` (const Container &studentai, char pasirinkimas)
Funkcija leidžia vartotojui pasirinkti rezultatų išvedimo būdą (į ekraną arba į failą).
- `char pasirinktiDuomenulvedima` ()
Funkcija leidžia vartotojui pasirinkti duomenų įvedimo būdą.
- `template<typename Container >`
`void ivedimasRanka` (Container &studentas)
Funkcija leidžia vartotojui rankiniu būdu įvesti studentų duomenis.
- `template<typename Container >`
`void duomenulsvedimasPagalStrategija` (Container &studentai, char pasirinkimas)
Funkcija leidžia vartotojui pasirinkti studentų rūšiavimo strategiją, skirsto studentus į „vargsiukus“ ir „kietiakus“ pagal pasirinktą strategiją, rūšiuoja pagal pasirinktą parametą ir išveda į failus.

5.3.1 Funkcijos Dokumentacija

5.3.1.1 duomenulsvedimasPagalStrategija()

```
template<typename Container >
void duomenulsvedimasPagalStrategija (
    Container & studentai,
    char pasirinkimas)
```

Funkcija leidžia vartotojui pasirinkti studentų rūšiavimo strategiją, skirsto studentus į „vargsiukus“ ir „kietiakus“ pagal pasirinktą strategiją, rūšiuoja pagal pasirinktą parametą ir išveda į failus.

Ši funkcija leidžia vartotojui pasirinkti vieną iš trijų strategijų studentų skirstymui į dvi grupes: „vargsiukus“ ir „kietiakus“. Po skirstymo grupės surūšiuojamos pagal pasirinktą parametą, o rezultatai išvedami į failus.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame saugomi studentų duomenys (pvz., std::vector arba std::list).
------------------	------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, kuriame saugomi studentų duomenys.
<i>pasirinkimas</i>	Pasirinkimas, nurodantis, kaip rezultatai turėtų būti išvedami. Galimos reikšmės: <ul style="list-style-type: none"> • 'V' - galutinis balas bus apskaičiuojamas naudojant vidurkį; • 'M' - galutinis balas bus apskaičiuojamas naudojant medianą;

Funkcijos eiga:

1. Leidžia vartotojui pasirinkti studentų rūšiavimo strategiją (1, 2 arba 3):
 - Strategija 1: Studentų sąrašas skirstomas į dvi naujas grupes (vargsiukai ir kietiakai).
 - Strategija 2: „Vargsiukai“ išskiriami į atskirą konteinerį, likusieji paliekami originaliame konteineryje.
 - Strategija 3: Panaši į strategiją 2, bet su papildoma optimizacijoma.
2. Vykdomas pasirinktų grupių rūšiavimas pagal pasirinktą parametą (pvz., vardą ar pažymį).
3. Rezultatai įrašomi į atskirus failus:
 - „vargsiukai.txt“: Studentai, kurių galutinis mažesnis už 5;
 - „kietiakai.txt“: Likusieji studentai.

Išimtys

<i>invalid_argument</i>	jei vartotojo įvestis yra netinkama arba klaidinga.
-------------------------	-----------------------------------------------------

5.3.1.2 generuotiFaila()

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

Funkcija generuoti failą su atsitiktiniais studentų duomenimis.

Ši funkcija sukuria tekstinį failą su studentų pavardėmis, vardais, 15 namų darbų įvertinimais ir egzamino balais. Sugeneruoti duomenys yra išsaugomi nurodytame faile.

Parametrai

<i>studentuSkaicius</i>	Studentų įrašų, kuriuos reikia sugeneruoti, skaičius.
<i>failoPavadinimas</i>	Failo, kuriame bus saugomi sugeneruoti duomenys, pavadinimas.

Funkcijos veikimo eiga:

- Į failą rašoma antraštė su stulpelių pavadinimais: "Pavarde", "Vardas", 15 namų darbų (ND1-ND15) ir "↔ Egzaminas".
- Ciklo pagalba sugeneruojami studentų duomenys:
 - Vardas ir pavardė formuojami kaip "Vardas1", "Pavarde1" ir pan.
 - Atsirtiktinai sugeneruojami 15 namų darbų įvertinimų ir vienas egzamino balas.
 - Visi sugeneruoti duomenys įrašomi į failą formatuotu būdu.

5.3.1.3 isvedimas()

```
template<typename Container >
void isvedimas (
    const Container & studentai,
    char pasirinkimas)
```

Funkcija išvesti studentų duomenis į terminalą.

Ši funkcija išveda studentų pavardes, vardus, galutinius balus (vidurkiu arba mediana) ir jų atminties adresus.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame saugomi studentų duomenys.
------------------	-------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, kuriame yra studentų objektai.
<i>pasirinkimas</i>	Pasirinkimo simbolis, kuris nurodo, kokią antraštę naudoti galutinį balą išvesti: <ul style="list-style-type: none"> • 'V': Galutinis balas apskaičiuotas vidurkiu. • 'M': Galutinis balas apskaičiuotas mediana.

5.3.1.4 isvedimasIFaila()

```
template<typename Container >
void isvedimasIFaila (
    const Container & studentai,
    char pasirinkimas,
    const string & failoPavadinimas)
```

Funkcija įrašyti studentų rezultatus į failą.

Ši funkcija įrašo pateiktame konteineryje esančių studentų duomenis į nurodytą failą. Galutinis balas gali būti apskaičiuotas vidurkiu arba mediana, priklausomai nuo vartotojo pasirinkimo.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame saugomi studentų duomenys (pvz., vector arba list).
------------------	--------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, kuriame yra studentų objektai.
<i>pasirinkimas</i>	Pasirinkimo simbolis, kuris nurodo, kokį galutinį balą įrašyti: <ul style="list-style-type: none"> • 'V': Galutinis balas apskaičiuotas vidurkiu. • 'M': Galutinis balas apskaičiuotas mediana.
<i>failoPavadinimas</i>	Failo pavadinimas, į kurį bus įrašomi rezultatai.

Išimtys

--	--

Jei failo nepavyksta atidaryti, klaida pranešama per standartinį klaidų srautą (`std::cerr`).

5.3.1.5 ivedimas()

```
template<typename Container >
void ivedimas (
    Container & studentai,
    string vardas,
    string pavarde)
```

Leidžia vartotojui įvesti studento pažymius (namų darbų pažymius ir egzamino rezultatą) ranka.

Ši funkcija leidžia vartotojui ranka įvesti studento namų darbų pažymius ir egzamino rezultatą. Įvedus šiuos duomenis, jie išsaugomi paduotame konteineryje.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame bus saugoma studentų informacija (pvz., std::vector arba std::list).
------------------	-------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, kuriame bus išsaugoti įvesti studentai.
<i>vardas</i>	Studentų vardas.
<i>pavarde</i>	Studentų pavardė.

5.3.1.6 ivedimasRanka()

```
template<typename Container >
void ivedimasRanka (
    Container & studentas)
```

Funkcija leidžia vartotojui rankiniu būdu įvesti studentų duomenis.

Ši funkcija leidžia įvesti nurodyto skaičiaus studentų vardus, pavardes ir pažymius. Vartotojas gali pasirinkti, ar įvesti pažymius rankiniu būdu, ar sugeneruoti juos automatiškai.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame bus saugomi studentų duomenys.
------------------	-----------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, kuriame bus išsaugoti įvesti studentų duomenys.
------------------	--------------------------------------------------------------

Išimtys

<i>invalid_argument</i>	jei įvestis nėra skaičius (pvz., studentų skaičius arba pažymių skaičius).
-------------------------	----------------------------------------------------------------------------

Funkcijos veiksmi:

1. Vartotojo prašoma nurodyti studentų skaičių.
2. Įvedami kiekvieno studento vardas ir pavardė.
3. Pasirenkamas pažymių įvedimo būdas:
 - Rankiniu būdu įvedant kiekvieną pažymį.
 - Automatiškai sugeneruojant nurodytą skaičių pažymių.

5.3.1.7 nuskaitytiIsFailo()

```
template<typename Container >
void nuskaitytiIsFailo (
    Container & studentai)
```

Funkcija skaityti studentų duomenis iš failo.

Ši funkcija nuskaityti studentų duomenis iš tekstinio failo ir užpildo pateiktą konteinerį studentų objektais.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame bus saugomi studentų objektai (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	-------------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, į kurį bus įrašomi nuskaityti studentų duomenys.
------------------	---------------------------------------------------------------

- Vartotojui prašoma įvesti failo pavadinimą. Jei failo atidaryti nepavyksta, išmetama klaida ir leidžiama bandyti dar kartą.
- Failo pirmoji eilutė (antraštė) praleidžiama.
- Kiekviena kita eilutė apdorojama: pirmieji du žodžiai interpretuojami kaip pavardė ir vardas, o likę skaičiai – kaip pažymiai.
- Paskutinis pažymys laikomas egzamino rezultatu; kiti įrašomi į namų darbų sąrašą.
- Netinkamos reikšmės (simboliai ar pažymiai už intervalo) ignoruojamos ir apie tai pranešama vartotojui.
- Laikmatis matuoja, kiek laiko užtruko failo nuskaitymas.

Išimtys



Išimtis `std::runtime_error` išmetama, jei nepavyksta atidaryti failo.

- Klaidos dėl netinkamų reikšmių (pvz., `invalid_argument` ar `out_of_range`) yra tvarkomos lokaliai ir apie jas informuojamas vartotojas.

5.3.1.8 pasirinktiDuomenuIvedima()

```
char pasirinktiDuomenuIvedima ()
```

Funkcija leidžia vartotojui pasirinkti duomenų įvedimo būdą.

Ši funkcija paklausia ir leidžia vartotojui pasirinkti, ar duomenis įvesti rankiniu būdu, nuskaityti iš failo, sugeneruoti naują failą ar demonstruoti metodus.

Gražina

Pasirinktas duomenų įvedimo būdas:

- 'I' – Įvesti duomenis rankiniu būdu.
- 'F' – Nuskaityti duomenis iš failo.
- 'G' – Sugeneruoti naują failą.
- 'D' – Demonstruoti metodus.

Pastaba

Funkcija užtikrina, kad vartotojas įves tinkamą pasirinkimą, kitaip prašoma įvesti dar kartą.

5.3.1.9 pasirinktiGalutinioskaiciavimoMetoda()

```
char pasirinktiGalutinioskaiciavimoMetoda ()
```

Funkcija leidžia vartotojui pasirinkti galutinio balo skaičiavimo metodą.

Ši funkcija prašo vartotojo pasirinkti, kuris metodas bus naudojamas galutiniam balui apskaičiuoti: vidurkiu arba mediana. Pasirinktas metodas grąžinamas kaip simbolis.

Gražina

char Grąžinamas vartotojo pasirinktas metodas:

- "V" – galutinio balo skaičiavimas naudojant vidurkį. –'M' – galutinio balo skaičiavimas naudojant medianą.

Pastaba

Funkcija užtikrina, kad grąžinama reikšmė visada bus validi (V arba M).

5.3.1.10 pasirinktiRezultatuIsvedimoBuda()

```
template<typename Container >
void pasirinktiRezultatuIsvedimoBuda (
    const Container & studentai,
    char pasirinkimas)
```

Funkcijaa leidžia vartotojui pasirinkti rezultatų išvedimo būdą (į ekraną arba į failą).

Ši funkcija vartotojui leidžia pasirinkti, ar išvesti studentų rezultatus į ekraną, ar išsaugoti juos į failą. Jei pasirenkamas failas, vartotojas turi nurodyti failo pavadinimą.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame saugomi studentų duomenys.
------------------	-------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris su studentų duomenimis.
<i>pasirinkimas</i>	Vartotojo pasirinkimas, naudojamas rezultatų išvedimui.

5.3.1.11 pasirinktiRusiavimoParametra()

```
char pasirinktiRusiavimoParametra ()
```

Funkcija leidžia vartotojui pasirinkti studentų rūšiavimo parametą.

Ši funkcija prašo vartotojo pasirinkti, pagal kokį parametą studentai bus surūšiuoti: pagal vardą, pavardę arba galutinį rezultatą mažėjimo ar didėjimo tvarka.

Gražina

char Grąžinamas vartotojo pasirinktas rūšiavimo parametras:

- "V" – rūšiavimas pagal vardą. –'P'– rūšiavimas pagal pavardę. –'M'– rūšiavimas pagal galutinį rezultatą mažėjimo tvarka. –'D' – rūšiavimas pagal galutinį rezultatą didėjimo tvarka.

Pastaba

Funkcija užtikrina, kad grąžinama reikšmė visada bus validi (V, P, M, arba D).

5.3.1.12 rusiuotiStudentus()

```
template<typename Container >
void rusiuotiStudentus (
    Container & studentai,
    char parametras)
```

Funkcija studentų rūšiavimui pagal pasirinktą parametą.

Ši funkcija rūšiuoja studentus pagal nurodytą kriterijų (vardas, pavardė arba galutinis pažymys (didėjančia arba mažėjančia tvarka)). Funkcija automatiškai pritaiko tinkamą rūšiavimo algoritmą, atsižvelgdama į naudojamo konteinerio tipą (`std::vector` arba `std::list`).

Template Parameters

<i>Container</i>	Konteinerio, kuriame saugomi studentų duomenys, tipas (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	----------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris su studentų duomenimis, kuris bus surūšiuotas.
<i>parametras</i>	Rūšiavimo kriterijus: <ul style="list-style-type: none"> • "V" - rūšiavimas pagal vardą (abėcėlės tvarka). Jei vardai sutampa, rūšiuojama pagal pavardę. • "P" - rūšiavimas pagal pavardę (abėcėlės tvarka). Jei pavardės sutampa, rūšiuojama pagal vardą. • "M" - rūšiavimas pagal galutinį pažymį mažėjimo tvarka. • "D" - rūšiavimas pagal galutinį pažymį didėjimo tvarka.

5.3.1.13 skaiciuotiGalutini()

```
void skaiciuotiGalutini (
    Studentas & s,
    char pasirinkimas)
```

Funkcija apskaičiuoti galutinį balą studentui.

Ši funkcija leidžia apskaičiuoti studento galutinį balą naudojant vidurkį arba medianą, priklausomai nuo vartotojo pasirinkimo.

Parametrai

<i>s</i>	Studentas , kurio galutinis balas bus apskaičiuotas.
<i>pasirinkimas</i>	Pasirinkimo reikšmė, kuri nurodo skaičiavimo metodą: <ul style="list-style-type: none"> • "V": Skaičiuoti galutinį balą vidurkiu. • "M": Skaičiuoti galutinį balą mediana.

5.3.1.14 skirstytiStudentusAntraStrategija()

```
template<typename Container >
void skirstytiStudentusAntraStrategija (
    Container & studentai,
    Container & vargsiukai)
```

Funkcija studentų skirstymui į dvi grupes naudojant antrąją strategiją.

Ši funkcija studentus padalina į dvi grupes: "vargšiukai" (studentai, kurių galutinis balas mažesnis už 5) ir likusieji studentai. Studentai rūšiuojami pagal galutinį balą mažėjančia tvarka, o "vargšiukai" perkeliama iš pradinio konteinerio į atskirą konteinerį.

Template Parameters

<i>Container</i>	Konteinerio, kuriame saugomi studentų duomenys, tipas (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	----------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Pradinis konteineris su visais studentų duomenimis.
<i>vargsiukai</i>	Konteineris, į kurį bus perkelti studentai su galutiniu balu mažesniu nei 5.

Funkcijos veikimo eiga:

- Studentai rūšiuojami pagal galutinį balą mažėjančia tvarka
- Pereinama per konteinerį *studentai* nuo galo:
 - Jei studento galutinis balas < 5.0 , jis perkeliama į konteinerį *vargsiukai*.
 - Kai randamas pirmasis studentas su balu ≥ 5.0 , iteracija nutraukiama, nes likę studentai taip pat turi balus ≥ 5.0 .

Pastaba

Funkcija modifikuoja pradinį *studentai* konteinerį, pašalindama iš jo studentus, kurie perkeliama į *vargsiukai*.

5.3.1.15 skirstytiStudentusPirmaStrategija()

```
template<typename Container >
void skirstytiStudentusPirmaStrategija (
    const Container & studentai,
    Container & vargsiukai,
    Container & kietiakai)
```

Funkcija studentų skirstymui į dvi grupes naudojant pirmąją strategiją.

Ši funkcija studentus padalina į dvi grupes: "vargšiukai" (studentai, kurių galutinis balas mažesnis už 5) ir "kietiakai" (studentai, kurių galutinis balas yra 5 arba didesnis).

Template Parameters

<i>Container</i>	Konteinerio, kuriame saugomi studentų duomenys, tipas (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	----------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Pirminis konteineris su visais studentų duomenimis.
<i>vargsiukai</i>	Konteineris, į kurį bus perkelti studentai su galutiniu balu mažesniu nei 5.
<i>kietiakai</i>	Konteineris, į kurį bus perkelti studentai su galutiniu balu 5 arba didesniu.

Funkcijos veikimo eiga:

- Pereinama per kiekvieną studentą iš pradinio konteinerio *studentai*.
- Pagal studento galutinį balą, gautą per metodą `getGalutinis()`, studentas įtraukiamas į atitinkamą grupę:
 - Jei galutinis balas < 5.0 , studentas perkeliamas į konteinerį *vargsiukai*.
 - Priešingu atveju, studentas perkeliamas į konteinerį *kietiakai*.

Pastaba

Funkcija nekeičia pradinio *studentai* konteinerio turinio.

5.3.1.16 skirstytiStudentusTreciaStrategija()

```
template<typename Container >
void skirstytiStudentusTreciaStrategija (
    Container & studentai,
    Container & vargsiukai)
```

Funkcija studentų skirstymui į dvi grupes naudojant trečiąją strategiją.

Ši funkcija studentus padalina į dvi grupes: "vargšiukai" (studentai, kurių galutinis balas mažesnis už 5) ir likusieji studentai. Naudojamas `std::stable_partition`, kad būtų paspartintas funkcijos veikimo laikas.

Template Parameters

<i>Container</i>	Konteinerio, kuriame saugomi studentų duomenys, tipas (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	----------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Pradinis konteineris su visais studentų duomenimis.
<i>vargsiukai</i>	Konteineris, į kurį bus perkelti studentai su galutiniu balu mažesniu nei 5.

Funkcijos veikimo eiga:

- Naudojama `std::stable_partition`, kad išskirstytų studentus į dvi dalis:

- Studentai, kurių galutinis balas ≥ 5.0 , lieka priekyje.
- Studentai, kurių galutinis balas < 5.0 , perkelti į pabaigą.
- Pereinama per konteinerį studentai nuo galo:
 - Jei studento galutinis balas < 5.0 , jis perkeliamas į konteinerį vargsiukai ir pašalinamas iš studentai.
 - Kai randamas pirmasis studentas su balu ≥ 5.0 , iteracija nutraukiama.

Pastaba

Funkcija modifikuoja pradinį studentai konteinerį, pašalindama iš jo studentus, kurie perkeliama į vargsiukai.

5.4 Stud.h

Eiti į šio failo dokumentaciją.

```

00001
00002 #ifndef STUD_H_INCLUDED
00003 #define STUD_H_INCLUDED
00004
00005 #include "Lib.h"
00006 #include "Studentas.h"
00007
00008
00009 // //struktura studento duomenims saugoti
00010 // struct Studentas {
00011 //     string vardas;
00012 //     string pavarde;
00013 //     vector<int> namuDarbai;
00014 //     int egzaminas;
00015 //     double galutinis;
00016 // };
00017
00029 template <typename Container>
00030 void ivedimas(Container& studentai, string vardas, string pavarde);
00031
00032
00044 void skaiciuotiGalutini(Studentas& s, char pasirinkimas);
00045
00046
00059 template <typename Container>
00060 void isvedimas(const Container& studentai, char pasirinkimas);
00061
00062
00084 template <typename Container>
00085 void nuskaitytiIsFailo(Container& studentai);
00086
00087
00105 template <typename Container>
00106 void isvedimasIFaila(const Container& studentai, char pasirinkimas, const string& failoPavadinimas);
00107
00108
00126 void generuotiFaila(int studentuSkaicius, const string& failoPavadinimas);
00127
00128
00149 template <typename Container>
00150 void skirstytiStudentusPirmaStrategija(const Container& studentai, Container& vargsiukai, Container&
kietiakai);
00151
00152
00173 template <typename Container>
00174 void skirstytiStudentusAntraStrategija(Container& studentai, Container& vargsiukai);
00175
00176
00198 template <typename Container>
00199 void skirstytiStudentusTreciaStrategija(Container& studentai, Container& vargsiukai);
00200
00201
00217 template <typename Container>
00218 void rusiuotiStudentus(Container& studentai, char parametras);
00219
00220
00233 char pasirinktiGalutiniosKaiciavimoMetoda();

```

```

00234
00235
00250 char pasirinktiRusiavimoParametra();
00251
00252
00263 template <typename Container>
00264 void pasirinktiRezultatuIsvedimoBuda(const Container& studentai, char pasirinkimas);
00265
00266
00281 char pasirinktiDuomenuIvedima();
00282
00283
00303 template <typename Container>
00304 void ivedimasRanka(Container& studentas);
00305
00306
00333 template <typename Container>
00334 void duomenuIsvedimasPagalStrategija(Container& studentai, char pasirinkimas);
00335
00336
00337 #endif

```

5.5 include/Studentas.h Failo Nuoroda

```

#include "../include/Zmogus.h"
#include <iostream>
#include <iomanip>
#include <string>
#include <vector>

```

Klasės

- class **Studentas**

Klasė, reprezentuojanti studento informaciją.

5.6 Studentas.h

[Eiti į šio failo dokumentaciją.](#)

```

00001 #ifndef STUDENTAS_H_INCLUDED
00002 #define STUDENTAS_H_INCLUDED
00003
00004 #include "../include/Zmogus.h"
00005 #include <iostream>
00006 #include <iomanip>
00007 #include <string>
00008 #include <vector>
00009
00010 using std::endl;
00011 using std::cout;
00012 using std::cin;
00013
00014 using std::string;
00015 using std::vector;
00016
00025 class Studentas : public Zmogus{
00026     private:
00027
00029         vector<int> namudarbai_;
00030
00032         int egzaminas_;
00033
00035         double galutinis_;
00036
00037     public:
00038         // konstruktoriai
00042         Studentas(){ };
00043

```

```

00052     Studentas(const string& vardas, const string& pavarde, const vector<int>& namudarbai, double
egzaminas);
00053
00061     Studentas(const string& vardas, const string& pavarde, int ndSkaicius);
00062
00068     Studentas(const Studentas& kitas);
00069
00075     ~Studentas();
00076
00077     // operatoriai
00087     friend std::ostream& operator<< (std::ostream& os, const Studentas& s);
00088
00099     friend std::istream& operator>> (std::istream& is, Studentas& s);
00100
00109     Studentas& operator=(const Studentas& kitas);
00110
00111     // seteriai
00112     // Seteriai
00113
00116     void setVardas(const string& vardas) { vardas_ = vardas; }
00117
00120     void setPavarde(const string& pavarde) { pavarde_ = pavarde; }
00121
00124     void setNamuDarbai(const vector<int>& namudarbai) { namudarbai_ = namudarbai; }
00125
00128     void setEgzaminas(const int& egzaminas) { egzaminas_ = egzaminas; }
00129
00132     void setGalutinis(const double& galutinis){ galutinis_ = galutinis; }
00133
00134     // geteriai
00135
00138     inline vector<int> getNamudarbai() const { return namudarbai_; }
00139
00142     inline int getEgzaminas() const { return egzaminas_; }
00143
00146     inline double getGalutinis() const { return galutinis_; }
00147
00148     // implementuoti virtualus geteriai
00151     string getVardas() const override { return vardas_; }
00152
00155     string getPavarde() const override { return pavarde_; }
00156
00157     // metodai
00166     void skaiciuotiGalutiniVidurkiu();
00167
00176     void skaiciuotiGalutiniMediana();
00177
00178 };
00179
00180 #endif

```

5.7 include/Timer.h Failo Nuoroda

```
#include "Lib.h"
```

Klasės

- class [Timer](#)

Laiko matavimo klasė, skirta skaičiuoti programos veikimo laiką.

5.8 Timer.h

Eiti į šio failo dokumentaciją.

```

00001 #include "Lib.h"
00002
00009 class Timer {
00010     private:
00011         using hrClock = std::chrono::high_resolution_clock;
00012         using durationDouble = std::chrono::duration<double>;

```

```

00013     std::chrono::time_point<hrClock> start;
00014 public:
00015     Timer() : start{ hrClock::now() } {}
00016     void reset() {
00017         start = hrClock::now();
00018     }
00019     double elapsed() const {
00020         return durationDouble (hrClock::now() - start).count();
00021     }
00022 };

```

5.9 include/Zmogus.h Failo Nuoroda

```

#include <iostream>
#include <string>

```

Klasės

- class [Zmogus](#)

Abstrakti bazinė klasė žmogaus duomenims saugoti.

5.10 Zmogus.h

[Eiti į šio failo dokumentaciją.](#)

```

00001 #ifndef ZMOGUS_H_INCLUDED
00002 #define ZMOGUS_H_INCLUDED
00003 #include <iostream>
00004 #include <string>
00005
00006 using std::string;
00007
00014 class Zmogus{
00015     protected:
00019         string vardas_;
00020
00024         string pavarde_;
00025
00031         Zmogus(){};
00032
00041         Zmogus(const string& vardas, const string& pavarde)
00042             : vardas_(vardas), pavarde_(pavarde) {}
00043
00044     public:
00050         virtual ~Zmogus() = default;
00051
00059         virtual string getVardas() const = 0;
00060
00068         virtual string getPavarde() const = 0;
00069 };
00070
00071 #endif

```

5.11 src/Main.cpp Failo Nuoroda

```

#include "../include/Lib.h"
#include "../include/Stud.h"
#include "../include/Timer.h"
#include "../include/Studentas.h"

```

Funkcijos

- int [main](#) ()

5.11.1 Funkcijos Dokumentacija

5.11.1.1 main()

```
int main ()
```

5.12 src/Stud.cpp Failo Nuoroda

```
#include "../include/Stud.h"
#include "../include/Timer.h"
#include "../include/Studentas.h"
```

Funkcijos

- template<typename Container >
void [ivedimas](#) (Container &studentai, string vardas, string pavarde)
Leidžia vartotojui įvesti studento pažymius (namų darbų pažymius ir egzamino rezultata) ranka.
- template void [ivedimas](#)< vector< [Studentas](#) > > (vector< [Studentas](#) > &, string, string)
- template void [ivedimas](#)< list< [Studentas](#) > > (list< [Studentas](#) > &, string, string)
- void [skaiciuotiGalutini](#) ([Studentas](#) &s, char pasirinkimas)
Funkcija apskaičiuoti galutinį balą studentui.
- template<typename Container >
void [isvedimas](#) (const Container &studentai, char pasirinkimas)
Funkcija išvesti studentų duomenis į terminalą.
- template<typename Container >
void [nuskaitytiFailo](#) (Container &studentai)
Funkcija skaityti studentų duomenis iš failo.
- template void [nuskaitytiFailo](#)< vector< [Studentas](#) > > (vector< [Studentas](#) > &)
- template void [nuskaitytiFailo](#)< list< [Studentas](#) > > (list< [Studentas](#) > &)
- template<typename Container >
void [isvedimasIFaila](#) (const Container &studentai, char pasirinkimas, const string &failoPavadinimas)
Funkcija įrašyti studentų rezultatus į failą.
- void [generuotiFaila](#) (int studentuSkaicius, const string &failoPavadinimas)
Funkcija generuoti failą su atsitiktiniais studentų duomenimis.
- template<typename Container >
void [skirstytiStudentusPirmaStrategija](#) (const Container &studentai, Container &vargsiukai, Container &kie-
tiakai)
Funkcija studentų skirstymui į dvi grupes naudojant pirmąją strategiją.
- template<typename Container >
void [skirstytiStudentusAntraStrategija](#) (Container &studentai, Container &vargsiukai)
Funkcija studentų skirstymui į dvi grupes naudojant antrąją strategiją.
- template<typename Container >
void [skirstytiStudentusTreciaStrategija](#) (Container &studentai, Container &vargsiukai)
Funkcija studentų skirstymui į dvi grupes naudojant trečiąją strategiją.

- `template<typename Container >`
`void rusiuotiStudentus` (Container &studentai, char parametras)
Funkcija studentų rūšiavimui pagal pasirinktą parametą.
- `char pasirinktiGalutinioskaiciavimoMetoda` ()
Funkcija leidžia vartotojui pasirinkti galutinio balo skaičiavimo metodą.
- `char pasirinktiRusiavimoParametra` ()
Funkcija leidžia vartotojui pasirinkti studentų rūšiavimo parametą.
- `template<typename Container >`
`void pasirinktiRezultatulsvedimoBuda` (const Container &studentai, char pasirinkimas)
Funkcija leidžia vartotojui pasirinkti rezultatų išvedimo būdą (į ekraną arba į failą).
- `template void pasirinktiRezultatulsvedimoBuda< vector< Studentas > >` (const vector< Studentas > &, char pasirinkimas)
- `template void pasirinktiRezultatulsvedimoBuda< list< Studentas > >` (const list< Studentas > &, char pasirinkimas)
- `char pasirinktiDuomenulvedima` ()
Funkcija leidžia vartotojui pasirinkti duomenų įvedimo būdą.
- `template<typename Container >`
`void ivedimasRanka` (Container &studentai)
Funkcija leidžia vartotojui rankiniu būdu įvesti studentų duomenis.
- `template void ivedimasRanka< vector< Studentas > >` (vector< Studentas > &)
- `template void ivedimasRanka< list< Studentas > >` (list< Studentas > &)
- `template<typename Container >`
`void duomenulsvedimasPagalStrategija` (Container &studentai, char pasirinkimas)
Funkcija leidžia vartotojui pasirinkti studentų rūšiavimo strategiją, skirsto studentus į „vargsiukus“ ir „kietiakus“ pagal pasirinktą strategiją, rūšiuoja pagal pasirinktą parametą ir išveda į failus.
- `template void duomenulsvedimasPagalStrategija< vector< Studentas > >` (vector< Studentas > &, char pasirinkimas)
- `template void duomenulsvedimasPagalStrategija< list< Studentas > >` (list< Studentas > &, char pasirinkimas)

5.12.1 Funkcijos Dokumentacija

5.12.1.1 duomenulsvedimasPagalStrategija()

```
template<typename Container >
void duomenulsvedimasPagalStrategija (
    Container & studentai,
    char pasirinkimas)
```

Funkcija leidžia vartotojui pasirinkti studentų rūšiavimo strategiją, skirsto studentus į „vargsiukus“ ir „kietiakus“ pagal pasirinktą strategiją, rūšiuoja pagal pasirinktą parametą ir išveda į failus.

Ši funkcija leidžia vartotojui pasirinkti vieną iš trijų strategijų studentų skirstymui į dvi grupes: „vargsiukus“ ir „kietiakus“. Po skirstymo grupės surūšiuojamos pagal pasirinktą parametą, o rezultatai išvedami į failus.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame saugomi studentų duomenys (pvz., std::vector arba std::list).
------------------	------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, kuriame saugomi studentų duomenys.
------------------	-------------------------------------------------

Parametrai

<i>pasirinkimas</i>	Pasirinkimas, nurodantis, kaip rezultatai turėtų būti išvedami. Galimos reikšmės: <ul style="list-style-type: none"> • 'V' - galutinis balas bus apskaičiuojamas naudojant vidurkį; • 'M' - galutinis balas bus apskaičiuojamas naudojant medianą;
---------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Funkcijos eiga:

1. Leidžia vartotojui pasirinkti studentų rūšiavimo strategiją (1, 2 arba 3):
 - Strategija 1: Studentų sąrašas skirstomas į dvi naujas grupes (vargsiukai ir kietiakai).
 - Strategija 2: „Vargsiukai“ išskiriami į atskirą konteinerį, likusieji paliekami originaliame konteineryje.
 - Strategija 3: Panaši į strategiją 2, bet su papildoma optimizacijoma.
2. Vykdomas pasirinktų grupių rūšiavimas pagal pasirinktą parametą (pvz., vardą ar pažymį).
3. Rezultatai įrašomi į atskirus failus:
 - „vargsiukai.txt“: Studentai, kurių galutinis mažesnis už 5;
 - „kietiakai.txt“: Likusieji studentai.

Išimtys

<i>invalid_argument</i>	jei vartotojo įvestis yra netinkama arba klaidinga.
-------------------------	-----------------------------------------------------

5.12.1.2 duomenusvedimasPagalStrategija< list< Studentas > >()

```
template void duomenusvedimasPagalStrategija< list< Studentas > > (
    list< Studentas > & ,
    char pasirinkimas)
```

5.12.1.3 duomenusvedimasPagalStrategija< vector< Studentas > >()

```
template void duomenusvedimasPagalStrategija< vector< Studentas > > (
    vector< Studentas > & ,
    char pasirinkimas)
```

5.12.1.4 generuotiFaila()

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

Funkcija generuoti failą su atsitiktiniais studentų duomenimis.

Ši funkcija sukuria tekstinį failą su studentų pavardėmis, vardais, 15 namų darbų įvertinimais ir egzamino balais. Sugeneruoti duomenys yra išsaugomi nurodytame faile.

Parametrai

<i>studentuSkaicius</i>	Studentų įrašų, kuriuos reikia sugeneruoti, skaičius.
<i>failoPavadinimas</i>	Failo, kuriame bus saugomi sugeneruoti duomenys, pavadinimas.

Funkcijos veikimo eiga:

- Į failą rašoma antraštė su stulpelių pavadinimais: "Pavarde", "Vardas", 15 namų darbų (ND1-ND15) ir "↔ Egzaminas".
- Ciklo pagalba sugeneruojami studentų duomenys:
 - Vardas ir pavardė formuojami kaip "Vardas1", "Pavarde1" ir pan.
 - Atsiktikotinai sugeneruojami 15 namų darbų įvertinimų ir vienas egzamino balas.
 - Visi sugeneruoti duomenys įrašomi į failą formatuotu būdu.

5.12.1.5 isvedimas()

```
template<typename Container >
void isvedimas (
    const Container & studentai,
    char pasirinkimas)
```

Funkcija išvesti studentų duomenis į terminalą.

Ši funkcija išveda studentų pavardes, vardus, galutinius balus (vidurkiu arba mediana) ir jų atminties adresus.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame saugomi studentų duomenys.
------------------	-------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, kuriame yra studentų objektai.
<i>pasirinkimas</i>	Pasirinkimo simbolis, kuris nurodo, kokią antraštę naudoti galutinį balą išvesti: <ul style="list-style-type: none"> • 'V': Galutinis balas apskaičiuotas vidurkiu. • 'M': Galutinis balas apskaičiuotas mediana.

5.12.1.6 isvedimasIFaila()

```
template<typename Container >
void isvedimasIFaila (
    const Container & studentai,
    char pasirinkimas,
    const string & failoPavadinimas)
```

Funkcija įrašyti studentų rezultatus į failą.

Ši funkcija įrašo pateiktame konteineryje esančių studentų duomenis į nurodytą failą. Galutinis balas gali būti apskaičiuotas vidurkiu arba mediana, priklausomai nuo vartotojo pasirinkimo.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame saugomi studentų duomenys (pvz., vector arba list).
------------------	--------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, kuriame yra studentų objektai.
<i>pasirinkimas</i>	Pasirinkimo simbolis, kuris nurodo, kokį galutinį balą įrašyti: <ul style="list-style-type: none"> • 'V': Galutinis balas apskaičiuotas vidurkiu. • 'M': Galutinis balas apskaičiuotas mediana.
<i>failoPavadinimas</i>	Failo pavadinimas, į kurį bus įrašomi rezultatai.

Išimtys

--	--

Jei failo nepavyksta atidaryti, klaida pranešama per standartinį klaidų srautą (`std::cerr`).

5.12.1.7 ivedimas()

```
template<typename Container >
void ivedimas (
    Container & studentai,
    string vardas,
    string pavarde)
```

Leidžia vartotojui įvesti studento pažymius (namų darbų pažymius ir egzamino rezultatą) ranka.

Ši funkcija leidžia vartotojui ranka įvesti studento namų darbų pažymius ir egzamino rezultatą. Įvedus šiuos duomenis, jie išsaugomi paduotame konteineryje.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame bus saugoma studentų informacija (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	----------------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, kuriame bus išsaugoti įvesti studentai.
<i>vardas</i>	Studentų vardas.
<i>pavarde</i>	Studentų pavardė.

5.12.1.8 ivedimas< list< Studentas > >()

```
template void ivedimas< list< Studentas > > (
    list< Studentas > & ,
    string ,
    string )
```

5.12.1.9 ivedimas< vector< Studentas > >()

```
template void ivedimas< vector< Studentas > > (
    vector< Studentas > & ,
    string ,
    string )
```

5.12.1.10 ivedimasRanka()

```
template<typename Container >
void ivedimasRanka (
    Container & studentas)
```

Funkcija leidžia vartotojui rankiniu būdu įvesti studentų duomenis.

Ši funkcija leidžia įvesti nurodyto skaičiaus studentų vardus, pavardes ir pažymius. Vartotojas gali pasirinkti, ar įvesti pažymius rankiniu būdu, ar sugeneruoti juos automatiškai.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame bus saugomi studentų duomenys.
------------------	-----------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, kuriame bus išsaugoti įvesti studentų duomenys.
------------------	--------------------------------------------------------------

Išimtys

<i>invalid_argument</i>	jei įvestis nėra skaičius (pvz., studentų skaičius arba pažymių skaičius).
-------------------------	----------------------------------------------------------------------------

Funkcijos veiksmi:

1. Vartotojo prašoma nurodyti studentų skaičių.
2. Įvedami kiekvieno studento vardas ir pavardė.
3. Pasirenkamas pažymių įvedimo būdas:
 - Rankiniu būdu įvedant kiekvieną pažymį.
 - Automatiškai sugeneruojant nurodytą skaičių pažymių.

5.12.1.11 ivedimasRanka< list< Studentas > >()

```
template void ivedimasRanka< list< Studentas > > (
    list< Studentas > & )
```

5.12.1.12 ivedimasRanka< vector< Studentas > >()

```
template void ivedimasRanka< vector< Studentas > > (
    vector< Studentas > & )
```

5.12.1.13 nuskaitytiFailo()

```
template<typename Container >
void nuskaitytiIsFailo (
    Container & studentai)
```

Funkcija skaityti studentų duomenis iš failo.

Ši funkcija nuskaityti studentų duomenis iš tekstinio failo ir užpildo pateiktą konteinerį studentų objektais.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame bus saugomi studentų objektai (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	-------------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris, į kurį bus įrašomi nuskaityti studentų duomenys.
------------------	---------------------------------------------------------------

- Vartotojui prašoma įvesti failo pavadinimą. Jei failo atidaryti nepavyksta, išmetama klaida ir leidžiama bandyti dar kartą.
- Failo pirmoji eilutė (antraštė) praleidžiama.
- Kiekviena kita eilutė apdorojama: pirmieji du žodžiai interpretuojami kaip pavardė ir vardas, o likę skaičiai – kaip pažymiai.
- Paskutinis pažymys laikomas egzamino rezultatu; kiti įrašomi į namų darbų sąrašą.
- Netinkamos reikšmės (simboliai ar pažymiai už intervalo) ignoruojamos ir apie tai pranešama vartotojui.
- Laikmatis matuoja, kiek laiko užtruko failo nuskaitymas.

Išimtys

--	--

Išimtis `std::runtime_error` išmetama, jei nepavyksta atidaryti failo.

- Klaidos dėl netinkamų reikšmių (pvz., `invalid_argument` ar `out_of_range`) yra tvarkomos lokaliai ir apie jas informuojamas vartotojas.

5.12.1.14 nuskaitytiFailo< list< Studentas > >()

```
template void nuskaitytiIsFailo< list< Studentas > > (
    list< Studentas > & )
```

5.12.1.15 nuskaitytiFailo< vector< Studentas > >()

```
template void nuskaitytiIsFailo< vector< Studentas > > (
    vector< Studentas > & )
```

5.12.1.16 pasirinktiDuomenulvedima()

```
char pasirinktiDuomenulvedima ()
```

Funkcija leidžia vartotojui pasirinkti duomenų įvedimo būdą.

Ši funkcija paklausia ir leidžia vartotojui pasirinkti, ar duomenis įvesti rankiniu būdu, nuskaityti iš failo, sugeneruoti naują failą ar demonstruoti metodus.

Gražina

Pasirinktas duomenų įvedimo būdas:

- 'I' – Įvesti duomenis rankiniu būdu.
- 'F' – Nuskaityti duomenis iš failo.
- 'G' – Sugeneruoti naują failą.
- 'D' – Demonstruoti metodus.

Pastaba

Funkcija užtikrina, kad vartotojas įves tinkamą pasirinkimą, kitaip prašoma įvesti dar kartą.

5.12.1.17 pasirinktiGalutinioskaiciavimoMetoda()

```
char pasirinktiGalutinioskaiciavimoMetoda ()
```

Funkcija leidžia vartotojui pasirinkti galutinio balo skaičiavimo metodą.

Ši funkcija prašo vartotojo pasirinkti, kuris metodas bus naudojamas galutiniam balui apskaičiuoti: vidurkiu arba mediana. Pasirinktas metodas grąžinamas kaip simbolis.

Gražina

char Grąžinamas vartotojo pasirinktas metodas:

- "V" – galutinio balo skaičiavimas naudojant vidurkį. "M" – galutinio balo skaičiavimas naudojant medianą.

Pastaba

Funkcija užtikrina, kad grąžinama reikšmė visada bus validi (V arba M).

5.12.1.18 pasirinktiRezultatulsvedimoBuda()

```
template<typename Container >  
void pasirinktiRezultatulsvedimoBuda (  
    const Container & studentai,  
    char pasirinkimas)
```

Funkcija leidžia vartotojui pasirinkti rezultatų išvedimo būdą (į ekraną arba į failą).

Ši funkcija vartotojui leidžia pasirinkti, ar išvesti studentų rezultatus į ekraną, ar išsaugoti juos į failą. Jei pasirinktas failas, vartotojas turi nurodyti failo pavadinimą.

Template Parameters

<i>Container</i>	Konteinerio tipas, kuriame saugomi studentų duomenys.
------------------	-------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris su studentų duomenimis.
<i>pasirinkimas</i>	Vartotojo pasirinkimas, naudojamas rezultatų išvedimui.

5.12.1.19 pasirinktiRezultatulsvedimoBuda< list< Studentas > >()

```
template void pasirinktiRezultatulsvedimoBuda< list< Studentas > > (
    const list< Studentas > & ,
    char pasirinkimas)
```

5.12.1.20 pasirinktiRezultatulsvedimoBuda< vector< Studentas > >()

```
template void pasirinktiRezultatulsvedimoBuda< vector< Studentas > > (
    const vector< Studentas > & ,
    char pasirinkimas)
```

5.12.1.21 pasirinktiRusiavimoParametra()

```
char pasirinktiRusiavimoParametra ()
```

Funkcija leidžia vartotojui pasirinkti studentų rūšiavimo parametą.

Ši funkcija prašo vartotojo pasirinkti, pagal kokį parametą studentai bus surūšiuoti: pagal vardą, pavardę arba galutinį rezultatą mažėjimo ar didėjimo tvarka.

Gražina

char Gražinamas vartotojo pasirinktas rūšiavimo parametras:

- "V"– rūšiavimas pagal vardą. "P"– rūšiavimas pagal pavardę. "M"– rūšiavimas pagal galutinį rezultatą mažėjimo tvarka. "D" - rūšiavimas pagal galutinį rezultatą didėjimo tvarka.

Pastaba

Funkcija užtikrina, kad gražinama reikšmė visada bus validi (V, P, M, arba D).

5.12.1.22 rusiuotiStudentus()

```
template<typename Container >
void rusiuotiStudentus (
    Container & studentai,
    char parametras)
```

Funkcija studentų rūšiavimui pagal pasirinktą parametą.

Ši funkcija rūšiuoja studentus pagal nurodytą kriterijų (vardas, pavardė arba galutinis pažymys (didėjančia arba mažėjančia tvarka)). Funkcija automatiškai pritaiko tinkamą rūšiavimo algoritmą, atsižvelgdama į naudojamo konteinerio tipą (std::vector arba std::list).

Template Parameters

<i>Container</i>	Konteinerio, kuriame saugomi studentų duomenys, tipas (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	----------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Konteineris su studentų duomenimis, kuris bus surūšiuotas.
<i>parametras</i>	Rūšiavimo kriterijus: <ul style="list-style-type: none"> • "V"- rūšiavimas pagal vardą (abėcėlės tvarka). Jei vardai sutampa, rūšiuojama pagal pavardę. -'P'- rūšiavimas pagal pavardę (abėcėlės tvarka). Jei pavardės sutampa, rūšiuojama pagal vardą. -'M'- rūšiavimas pagal galutinį pažymį mažėjimo tvarka. -'D' - rūšiavimas pagal galutinį pažymį didėjimo tvarka.

5.12.1.23 skaiciuotiGalutini()

```
void skaiciuotiGalutini (
    Studentas & s,
    char pasirinkimas)
```

Funkcija apskaičiuoti galutinį balą studentui.

Ši funkcija leidžia apskaičiuoti studento galutinį balą naudojant vidurkį arba medianą, priklausomai nuo vartotojo pasirinkimo.

Parametrai

<i>s</i>	Studentas , kurio galutinis balas bus apskaičiuotas.
<i>pasirinkimas</i>	Pasirinkimo reikšmė, kuri nurodo skaičiavimo metodą: <ul style="list-style-type: none"> • 'V': Skaičiuoti galutinį balą vidurkiu. • 'M': Skaičiuoti galutinį balą mediana.

5.12.1.24 skirstytiStudentusAntraStrategija()

```
template<typename Container >
void skirstytiStudentusAntraStrategija (
    Container & studentai,
    Container & vargsiukai)
```

Funkcija studentų skirstymui į dvi grupes naudojant antrąją strategiją.

Ši funkcija studentus padalina į dvi grupes: "vargšiukai" (studentai, kurių galutinis balas mažesnis už 5) ir likusieji studentai. Studentai rūšiuojami pagal galutinį balą mažėjančia tvarka, o "vargšiukai" perkeliama iš pradinio konteinerio į atskirą konteinerį.

Template Parameters

<i>Container</i>	Konteinerio, kuriame saugomi studentų duomenys, tipas (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	----------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Pradinis konteineris su visais studentų duomenimis.
<i>vargsiukai</i>	Konteineris, į kurį bus perkelti studentai su galutiniu balu mažesniu nei 5.

Funkcijos veikimo eiga:

- Studentai rūšiuojami pagal galutinį balą mažėjančia tvarka
- Pereinama per konteinerį *studentai* nuo galo:
 - Jei studento galutinis balas < 5.0 , jis perkliamas į konteinerį *vargsiukai*.
 - Kai randamas pirmasis studentas su balu ≥ 5.0 , iteracija nutraukiama, nes likę studentai taip pat turi balus ≥ 5.0 .

Pastaba

Funkcija modifikuoja pradinį *studentai* konteinerį, pašalindama iš jo studentus, kurie perkeliama į *vargsiukai*.

5.12.1.25 skirstytiStudentusPirmaStrategija()

```
template<typename Container >
void skirstytiStudentusPirmaStrategija (
    const Container & studentai,
    Container & vargsiukai,
    Container & kietiakai)
```

Funkcija studentų skirstymui į dvi grupes naudojant pirmąją strategiją.

Ši funkcija studentus padalina į dvi grupes: "vargšiukai" (studentai, kurių galutinis balas mažesnis už 5) ir "kietiakai" (studentai, kurių galutinis balas yra 5 arba didesnis).

Template Parameters

<i>Container</i>	Konteinerio, kuriame saugomi studentų duomenys, tipas (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	----------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Pirminis konteineris su visais studentų duomenimis.
<i>vargsiukai</i>	Konteineris, į kurį bus perkelti studentai su galutiniu balu mažesniu nei 5.
<i>kietiakai</i>	Konteineris, į kurį bus perkelti studentai su galutiniu balu 5 arba didesniu.

Funkcijos veikimo eiga:

- Pereinama per kiekvieną studentą iš pradinio konteinerio *studentai*.
- Pagal studento galutinį balą, gautą per metodą `getGalutinis()`, studentas įtraukiamas į atitinkamą grupę:
 - Jei galutinis balas < 5.0 , studentas perkliamas į konteinerį *vargsiukai*.
 - Priešingu atveju, studentas perkliamas į konteinerį *kietiakai*.

Pastaba

Funkcija nekeičia pradinio `studentai` konteinerio turinio.

5.12.1.26 skirstytiStudentusTreciaStrategija()

```
template<typename Container >
void skirstytiStudentusTreciaStrategija (
    Container & studentai,
    Container & vargsiukai)
```

Funkcija studentų skirstymui į dvi grupes naudojant trečiąją strategiją.

Ši funkcija studentus padalina į dvi grupes: "vargšiukai" (studentai, kurių galutinis balas mažesnis už 5) ir likusieji studentai. Naudojamas `std::stable_partition`, kad būtų paspartintas funkcijos veikimo laikas.

Template Parameters

<i>Container</i>	Konteinerio, kuriame saugomi studentų duomenys, tipas (pvz., <code>std::vector</code> arba <code>std::list</code>).
------------------	----------------------------------------------------------------------------------------------------------------------

Parametrai

<i>studentai</i>	Pradinis konteineris su visais studentų duomenimis.
<i>vargsiukai</i>	Konteineris, į kurį bus perkelti studentai su galutiniu balu mažesniu nei 5.

Funkcijos veikimo eiga:

- Naudojama `std::stable_partition`, kad išskirstytų studentus į dvi dalis:
 - Studentai, kurių galutinis balas ≥ 5.0 , lieka priekyje.
 - Studentai, kurių galutinis balas < 5.0 , perkelti į pabaigą.
- Pereinama per konteinerį `studentai` nuo galo:
 - Jei studento galutinis balas < 5.0 , jis perkeliamas į konteinerį `vargsiukai` ir pašalinamas iš `studentai`.
 - Kai randamas pirmasis studentas su balu ≥ 5.0 , iteracija nutraukiama.

Pastaba

Funkcija modifikuoja pradinį `studentai` konteinerį, pašalindama iš jo studentus, kurie perkeltami į `vargsiukai`.

5.13 src/Studentas.cpp Failo Nuoroda

```
#include "../include/Stud.h"
#include "../include/Studentas.h"
```

Funkcijos

- ostream & `operator<<` (ostream &os, const `Studentas` &s)
- istream & `operator>>` (istream &is, `Studentas` &s)

5.13.1 Funkcijos Dokumentacija

5.13.1.1 `operator<<()`

```
ostream & operator<< (  
    ostream & os,  
    const Studentas & s)
```

5.13.1.2 `operator>>()`

```
istream & operator>> (  
    istream & is,  
    Studentas & s)
```

Rodyklė

~Studentas
 Studentas, [10](#)
~Zmogus
 Zmogus, [16](#)

duomenusvedimasPagalStrategija
 Stud.cpp, [35](#)
 Stud.h, [21](#)
duomenusvedimasPagalStrategija< list< Studentas >
 >
 Stud.cpp, [36](#)
duomenusvedimasPagalStrategija< vector< Studentas
 > >
 Stud.cpp, [36](#)

elapsed
 Timer, [14](#)

generuotiFaila
 Stud.cpp, [36](#)
 Stud.h, [22](#)
getEgzaminas
 Studentas, [10](#)
getGalutinis
 Studentas, [10](#)
getNamudarbai
 Studentas, [10](#)
getPavarde
 Studentas, [10](#)
 Zmogus, [16](#)
getVardas
 Studentas, [11](#)
 Zmogus, [16](#)

include/Lib.h, [19](#)
include/Stud.h, [20](#), [30](#)
include/Studentas.h, [31](#)
include/Timer.h, [32](#)
include/Zmogus.h, [33](#)
isvedimas
 Stud.cpp, [37](#)
 Stud.h, [22](#)
isvedimasIFaila
 Stud.cpp, [37](#)
 Stud.h, [23](#)
ivedimas
 Stud.cpp, [38](#)
 Stud.h, [23](#)
ivedimas< list< Studentas > >
 Stud.cpp, [38](#)

ivedimas< vector< Studentas > >
 Stud.cpp, [38](#)
ivedimasRanka
 Stud.cpp, [39](#)
 Stud.h, [24](#)
ivedimasRanka< list< Studentas > >
 Stud.cpp, [39](#)
ivedimasRanka< vector< Studentas > >
 Stud.cpp, [39](#)

main
 Main.cpp, [34](#)
Main.cpp
 main, [34](#)

nuskaitytiFailo
 Stud.cpp, [39](#)
 Stud.h, [24](#)
nuskaitytiFailo< list< Studentas > >
 Stud.cpp, [40](#)
nuskaitytiFailo< vector< Studentas > >
 Stud.cpp, [40](#)

operator<<
 Studentas, [13](#)
 Studentas.cpp, [46](#)
operator>>
 Studentas, [13](#)
 Studentas.cpp, [46](#)
operator=
 Studentas, [11](#)

pasirinktiDuomenulvedima
 Stud.cpp, [40](#)
 Stud.h, [25](#)
pasirinktiGalutinoskaiciavimoMetoda
 Stud.cpp, [41](#)
 Stud.h, [25](#)
pasirinktiRezultatulsvedimoBuda
 Stud.cpp, [41](#)
 Stud.h, [26](#)
pasirinktiRezultatulsvedimoBuda< list< Studentas > >
 Stud.cpp, [42](#)
pasirinktiRezultatulsvedimoBuda< vector< Studentas
 > >
 Stud.cpp, [42](#)
pasirinktiRusiavimoParametra
 Stud.cpp, [42](#)
 Stud.h, [26](#)
pavarde_

- Zmogus, [17](#)
- reset
 - Timer, [14](#)
- rusiuotiStudentus
 - Stud.cpp, [42](#)
 - Stud.h, [26](#)
- setEgzaminas
 - Studentas, [11](#)
- setGalutinis
 - Studentas, [12](#)
- setNamuDarbai
 - Studentas, [12](#)
- setPavarde
 - Studentas, [12](#)
- setVardas
 - Studentas, [12](#)
- skaiciuotiGalutini
 - Stud.cpp, [43](#)
 - Stud.h, [27](#)
- skaiciuotiGalutiniMediana
 - Studentas, [13](#)
- skaiciuotiGalutiniVidurkiu
 - Studentas, [13](#)
- skirstytiStudentusAntraStrategija
 - Stud.cpp, [43](#)
 - Stud.h, [27](#)
- skirstytiStudentusPirmaStrategija
 - Stud.cpp, [44](#)
 - Stud.h, [28](#)
- skirstytiStudentusTreciaStrategija
 - Stud.cpp, [45](#)
 - Stud.h, [29](#)
- src/Main.cpp, [33](#)
- src/Stud.cpp, [34](#)
- src/Studentas.cpp, [45](#)
- Stud.cpp
 - duomenulsvedimasPagalStrategija, [35](#)
 - duomenulsvedimasPagalStrategija< list< Studentas > >, [36](#)
 - duomenulsvedimasPagalStrategija< vector< Studentas > >, [36](#)
 - generuotiFaila, [36](#)
 - isvedimas, [37](#)
 - isvedimasIFaila, [37](#)
 - ivedimas, [38](#)
 - ivedimas< list< Studentas > >, [38](#)
 - ivedimas< vector< Studentas > >, [38](#)
 - ivedimasRanka, [39](#)
 - ivedimasRanka< list< Studentas > >, [39](#)
 - ivedimasRanka< vector< Studentas > >, [39](#)
 - nuskaitytilsFailo, [39](#)
 - nuskaitytilsFailo< list< Studentas > >, [40](#)
 - nuskaitytilsFailo< vector< Studentas > >, [40](#)
 - pasirinktiDuomenulvedima, [40](#)
 - pasirinktiGalutinoskaiciavimoMetoda, [41](#)
 - pasirinktiRezultatulsvedimoBuda, [41](#)
 - pasirinktiRezultatulsvedimoBuda< list< Studentas > >, [42](#)
 - pasirinktiRezultatulsvedimoBuda< vector< Studentas > >, [42](#)
 - pasirinktiRusiavimoParametra, [42](#)
 - rusiuotiStudentus, [42](#)
 - skaiciuotiGalutini, [43](#)
 - skirstytiStudentusAntraStrategija, [43](#)
 - skirstytiStudentusPirmaStrategija, [44](#)
 - skirstytiStudentusTreciaStrategija, [45](#)
- Stud.h
 - duomenulsvedimasPagalStrategija, [21](#)
 - generuotiFaila, [22](#)
 - isvedimas, [22](#)
 - isvedimasIFaila, [23](#)
 - ivedimas, [23](#)
 - ivedimasRanka, [24](#)
 - nuskaitytilsFailo, [24](#)
 - pasirinktiDuomenulvedima, [25](#)
 - pasirinktiGalutinoskaiciavimoMetoda, [25](#)
 - pasirinktiRezultatulsvedimoBuda, [26](#)
 - pasirinktiRusiavimoParametra, [26](#)
 - rusiuotiStudentus, [26](#)
 - skaiciuotiGalutini, [27](#)
 - skirstytiStudentusAntraStrategija, [27](#)
 - skirstytiStudentusPirmaStrategija, [28](#)
 - skirstytiStudentusTreciaStrategija, [29](#)
- Studentas, [7](#)
 - ~Studentas, [10](#)
 - getEgzaminas, [10](#)
 - getGalutinis, [10](#)
 - getNamuDarbai, [10](#)
 - getPavarde, [10](#)
 - getVardas, [11](#)
 - operator<<, [13](#)
 - operator>>, [13](#)
 - operator=, [11](#)
 - setEgzaminas, [11](#)
 - setGalutinis, [12](#)
 - setNamuDarbai, [12](#)
 - setPavarde, [12](#)
 - setVardas, [12](#)
 - skaiciuotiGalutiniMediana, [13](#)
 - skaiciuotiGalutiniVidurkiu, [13](#)
 - Studentas, [9](#)
- Studentas.cpp
 - operator<<, [46](#)
 - operator>>, [46](#)
- Timer, [14](#)
 - elapsed, [14](#)
 - reset, [14](#)
 - Timer, [14](#)
- vardas_
 - Zmogus, [17](#)
- Zmogus, [15](#)
 - ~Zmogus, [16](#)

getPavarde, [16](#)
getVardas, [16](#)
pavarde_, [17](#)
vardas_, [17](#)
Zmogus, [16](#)