

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	8
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 getEgz()	10
4.1.3.2 getGalutinis()	10
4.1.3.3 getNd()	10
4.1.3.4 getPavarde()	11
4.1.3.5 getVardas()	11
4.1.3.6 lvertinimas_med()	11
4.1.3.7 lvertinimas_vid()	11
4.1.3.8 operator=()	11
4.1.3.9 setEgz()	12
4.1.3.10 setGalutinis()	12
4.1.3.11 setNd()	12
4.1.3.12 setPavarde()	12
4.1.3.13 setVardas()	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	15
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/Mylib.h Failo Nuoroda	19
5.2 Mylib.h	19
5.3 include/Stud.h Failo Nuoroda	20
5.3.1 Funkcijos Dokumentacija	21
5.3.1.1 Duom_is_failo()	21
5.3.1.2 Duom_ivedimas()	22
5.3.1.3 Duom_tvarkymas()	22
5.3.1.4 FailasPgalkategorija()	23
5.3.1.5 Info_ivedimas_ranka()	24
5.3.1.6 Kategorijos_Priskirimas1()	24
5.3.1.7 Kategorijos_Priskirimas2()	25
5.3.1.8 Kategorijos_Priskirimas3()	25
5.3.1.9 pasirinkimas_del_duom_strukturos()	26
5.3.1.10 pasirinkimas_del_galutinio()	26
5.3.1.11 pasirinkimas_del_programos_vykdymo()	26
5.3.1.12 pasirinkimas_del_rusiavimo()	26
5.3.1.13 pasirinkimas_del_strategijos()	27
5.3.1.14 pasirinkimas_isvedimo()	27
5.3.1.15 Rez_antraste()	27
5.3.1.16 SpausdinimasRez()	28
5.3.1.17 Stud_failu_generavimas()	29
5.3.1.18 Studentu_rusiavimas()	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	32
5.10 Zmogus.h	33

5.11 src/Stud.cpp Failo Nuoroda	33
5.11.1 Funkcijos Dokumentacija	35
5.11.1.1 Duom_is_failo()	35
5.11.1.2 Duom_is_failo< list< Studentas > >()	35
5.11.1.3 Duom_is_failo< vector< Studentas > >()	35
5.11.1.4 Duom_ivedimas()	35
5.11.1.5 Duom_ivedimas< list< Studentas > >()	36
5.11.1.6 Duom_ivedimas< vector< Studentas > >()	36
5.11.1.7 Duom_tvarkymas()	36
5.11.1.8 Duom_tvarkymas< list< Studentas > >()	37
5.11.1.9 Duom_tvarkymas< vector< Studentas > >()	37
5.11.1.10 FailasPgalKategorija()	38
5.11.1.11 Info_ivedimas_ranka()	38
5.11.1.12 Info_ivedimas_ranka< list< Studentas > >()	39
5.11.1.13 Info_ivedimas_ranka< vector< Studentas > >()	39
5.11.1.14 Kategorijos_Priskirimas1()	39
5.11.1.15 Kategorijos_Priskirimas2()	39
5.11.1.16 Kategorijos_Priskirimas3()	40
5.11.1.17 pasirinkimas_del_duom_strukturos()	40
5.11.1.18 pasirinkimas_del_galutinio()	41
5.11.1.19 pasirinkimas_del_programos_vykdyimo()	41
5.11.1.20 pasirinkimas_del_rusiavimo()	41
5.11.1.21 pasirinkimas_del_strategijos()	41
5.11.1.22 pasirinkimas_isvedimo()	42
5.11.1.23 Rez_antraste()	42
5.11.1.24 SpausdinimasRez()	42
5.11.1.25 SpausdinimasRez< list< Studentas > >()	43
5.11.1.26 SpausdinimasRez< vector< Studentas > >()	43
5.11.1.27 Stud_failu_generavimas()	43
5.11.1.28 Studentu_rusiavimas()	43
5.12 src/Studentai.cpp Failo Nuoroda	44
5.12.1 Funkcijos Dokumentacija	44
5.12.1.1 main()	44
5.13 src/Studentas.cpp Failo Nuoroda	44
5.13.1 Funkcijos Dokumentacija	45
5.13.1.1 operator<<()	45
5.13.1.2 operator>>()	45

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	Žmogaus išvestinė klasė, skirta studentų duomenims saugoti tvarkyti	7
Timer	Laiko matavimo klasė, skirta skaičiuoti programos veikimo laiką	14
Zmogus	Abstrakti bazinė klasė, skirta žmonių duomenims valdyti	15

skyrius 3

Failo Indeksas

3.1 Failai

Visų failų sąrašas su trumpais aprašymais:

include/Mylib.h	19
include/Stud.h	20
include/Studentas.h	31
include/timer.h	32
include/Zmogus.h	32
src/Stud.cpp	33
src/Studentai.cpp	44
src/Studentas.cpp	44

skyrius 4

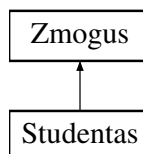
Klasės Dokumentacija

4.1 Studentas Klasė

Žmogaus išvestinė klasė, skirta studentų duomenims saugoti tvarkyti.

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

Paveldimumo diagrama Studentas:



Vieši Metodai

- **Studentas** ()
Numatytasis konstruktorius. Sukuria tuščią Studento klasės objektą.
- **Studentas** (const string &vardas, const string &pavarde, const vector< double > &nd, double egz)
Konstruktorius su parametrais.
- **Studentas** (const string &vardas, const string &pavarde, int kiekis)
Konstruktorius su generuojamu namų darbų įvertinimų kiekiu. Sukuria objektą su atsitiktiniais įvertinimais.
- **Studentas** (const **Studentas** &saltnis)
*Kopijavimo konstruktorius. Leidžia sukurti naują **Studentas** objektą, kopijuojant duomenis iš kito objekto.*
- **~Studentas** ()
Destruktorius.
- **Studentas** & **operator=** (const **Studentas** &saltnis)
Kopijavimo priskyrimo operatorius. Leidžia priskirti vieno Studento duomenis kitam Studento klasės objektui.
- void **setVardas** (const string &vardas)
Nustato studento vardą.
- void **setPavarde** (const string &pavarde)
Nustato studento pavardę.
- void **setNd** (const vector< double > &nd)
Nustato namų darbų įvertinimus.
- void **setEgz** (const double &egz)

- *Nustato egzamino įvertinimą.*
- void `setGalutinis` (const double &ivertinimas)
- *Nustato galutinį įvertinimą.*
- string `getVardas` () const override
- *Gauti studento vardą.*
- string `getPavarde` () const override
- *Gauti studento pavardę.*
- double `getEgz` () const
- *Gauti egzamino įvertinimą.*
- vector< double > `getNd` () const
- *Gauti namų darbų įvertinimų vektorių.*
- double `getGalutinis` () const
- *Gauti galutinį įvertinimą.*
- void `lvertinimas_vid` ()
- *Apskaičiuoja galutinį pažymį pagal vidurkį.*
- void `lvertinimas_med` ()
- *Apskaičiuoja galutinį pažymį pagal medianą.*

Vieši Metodai inherited from `Zmogus`

- virtual `~Zmogus` ()
- *Virtualus destruktorius.*

Draugai

- std::ostream & `operator<<` (std::ostream &os, const `Studentas` &s)
- *Išvesties operatorius.*
- std::istream & `operator>>` (std::istream &is, `Studentas` &s)
- *Įvesties operatorius, skirtas studento duomenims įvesti.*

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_`
- string `pavarde_`

4.1.1 Smulkus aprašymas

Žmogaus išvestinė klasė, skirta studentų duomenims saugoti tvarkyti.

Klasė apibrėžia pagrindinius atributus: vardas, pavarde, namų darbų įvertinimai, galutinis įvertinimas.

4.1.2 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.1.2.1 Studentas() [1/4]

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

Numatytasis konstruktorius. Sukuria tuščią Studento klasės objektą.

4.1.2.2 Studentas() [2/4]

```
Studentas::Studentas (  
    const string & vardas,  
    const string & pavarde,  
    const vector< double > & nd,  
    double egz)
```

Konstruktorius su parametrais.

Parametrai

<i>vardas</i>	Studento vardas.
<i>pavarde</i>	Studento pavardė.
<i>nd</i>	Namų darbų įvertinimų vektorius.
<i>egz</i>	Egzamino įvertinimas.

4.1.2.3 Studentas() [3/4]

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

Konstruktorius su generuojamu namų darbų įvertinimų kiekiu. Sukuria objektą su atsitiktiniais įvertinimais.

Parametrai

<i>vardas</i>	Studento vardas.
<i>pavarde</i>	Studento pavardė.
<i>kiekis</i>	Kiek namų darbų įvertinimų sugeneruoti.

4.1.2.4 Studentas() [4/4]

```
Studentas::Studentas (  
    const Studentas & saltinis)
```

Kopijavimo konstruktorius. Leidžia sukurti naują **Studentas** objektą, kopijuojant duomenis iš kito objekto.

Parametrai

<i>saltnis</i>	Kitas Studento klasės objektas, kurio duomenys kopijuojami.
----------------	-------------------------------------------------------------

4.1.2.5 ~Studentas()

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

Destruktorius.

Atsakingas už išteklį atlaisvinimą, iškviečiamas automatiškai.

4.1.3 Metodų Dokumentacija**4.1.3.1 getEgz()**

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

Gauti egzamino įvertinimą.

Gražina

Egzamino įvertinimą.

4.1.3.2 getGalutinis()

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

Gauti galutinį įvertinimą.

Gražina

Galutinis įvertinimas.

4.1.3.3 getNd()

```
vector< double > Studentas::getNd () const [inline]
```

Gauti namų darbų įvertinimų vektorių.

Gražina

Namų darbų įvertinimų vektorius.

4.1.3.4 getPavarde()

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

Gauti studento pavardę.

Gražina

`Studentas` pavarde.

Realizuoja `Zmogus`.

4.1.3.5 getVardas()

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

Gauti studento vardą.

Gražina

`Studentas` vardas.

Realizuoja `Zmogus`.

4.1.3.6 Ivertinimas_med()

```
void Studentas::Ivertinimas_med ()
```

Apskaičiuoja galutinį pažymį pagal medianą.

Naudoja namų darbų įvertinimų medianą ir egzamino įvertinimą pagal formulę: Galutinis = 0.4 * NamųDarbų↔
Mediana + 0.6 * Egzaminas.

4.1.3.7 Ivertinimas_vid()

```
void Studentas::Ivertinimas_vid ()
```

Apskaičiuoja galutinį pažymį pagal vidurkį.

Naudoja namų darbų įvertinimų vidurkį ir egzamino įvertinimą pagal formulę: Galutinis = 0.4 * NamųDarbųVidurkis
+ 0.6 * Egzaminas.

4.1.3.8 operator=()

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

Kopijavimo priskyrimo operatorius. Leidžia priskirti vieno Studento duomenis kitam Studento klasės objektui.

Parametrai

<i>saltnis</i>	Kitas Studento klasės objektas, kurio duomenys priskiriami.
----------------	-------------------------------------------------------------

Gražina

Nuoroda į šį objektą.

4.1.3.9 setEgz()

```
void Studentas::setEgz (
    const double & egz) [inline]
```

Nustato egzamino įvertinimą.

Parametrai

<i>egz</i>	Naujas egzamino įvertinimas.
------------	------------------------------

4.1.3.10 setGalutinis()

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

Nustato galutinį įvertinimą.

Parametrai

<i>ivertinimas</i>	Naujas galutinis įvertinimas.
--------------------	-------------------------------

4.1.3.11 setNd()

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

Nustato namų darbų įvertinimus.

Parametrai

<i>nd</i>	Naujas namų darbų įvertinimų vektorius.
-----------	-----------------------------------------

4.1.3.12 setPavarde()

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

Nustato studento pavardę.

Parametrai

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

4.1.3.13 setVardas()

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

Nustato studento vardą.

Parametrai

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

4.1.4 Draugiškų Ir Susijusių Funkcijų Dokumentacija

4.1.4.1 operator<<

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

Išvesties operatorius.

Operatorius naudojamas tiek išvedimui į terminalą `std::cout`, tiek rašymui į failą `std::ofstream`. Išveda studento informaciją (vardą, pavardę, galutinį balą) į išvesties srautą.

Parametrai

<i>os</i>	Išvesties srautas.
<i>s</i>	Studento klasės objektas, kurio duomenys spausdinami.

Gražina

Modifikuotas išvesties srautas.

4.1.4.2 operator>>

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

Įvesties operatorius, skirtas studento duomenims įvesti.

Naudojamas perdengtas operatorius `istream& operator>>(std::istream& is, Studentas& s)`, kuris leidžia įvesti duomenis dviem būdais:

1. Rankinis įvedimas iš terminalo, naudojant `std::cin`. Vartotojas turi pateikti studentų vardus bei pavardes, įvertinimai gali būti įvesti vartotojo rankiniu būdu arba sugeneruojami atsitiktinai pagal vartotojo pasirinkimą.
2. Duomenų nuskaitymas iš teksto srauto (pvz.: `std::stringstream`). Ši funkcija leidžia įvesti studento vardą, pavardę, namų darbų balus ir egzamino balą.

Parametrai

<i>is</i>	Įvesties srautas (std::cin arba failo srautas).
<i>s</i>	Studento klasės objektas, į kurį įrašomi ir saugomi duomenys: vardas, pavardė, namų darbai, egzamino įvertinimas.

Grazina

Modifikuotas įvesties srautas.

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:

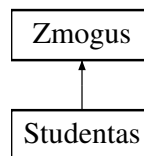
- include/[timer.h](#)

4.3 Zmogus Klasė

Abstrakti bazinė klasė, skirta žmonių duomenims valdyti.

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

Paveldimumo diagrama Zmogus:



Vieši Metodai

- virtual string [getVardas](#) () const =0
Gauti žmogaus vardą.
- virtual string [getPavarde](#) () const =0
Gauti žmogaus pavardę.
- virtual [~Zmogus](#) ()
Virtualus destruktorius.

Apsaugoti Metodai

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

Apsaugoti Atributai

- string [vardas_](#)
- string [pavarde_](#)

4.3.1 Smulkus aprašymas

Abstrakti bazinė klasė, skirta žmonių duomenims valdyti.

Klasė yra abstrakčioji, tai yra negalima sukurti žmogaus tipo objektų, o tik objektus iš jos išvestinių klasių.

Klasė apibrėžia pagrindinius atributus (vardas, pavardė) ir funkcijas, kurios turi būti įgyvendintos paveldėtose klasėse.

4.3.2 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.3.2.1 Zmogus() [1/2]

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

Numatytasis konstruktorius.

Inicijuoja tuščią `Zmogus` objektą. Naudojamas paveldėtos klasės konstruktoriuose.

4.3.2.2 Zmogus() [2/2]

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

Konstruktorius su parametrais.

Parametrai

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

4.3.2.3 ~Zmogus()

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

Virtualus destruktoriaus.

Naudojamas išteklių atlaisvinimui paveldėtose klasėse.

4.3.3 Metodų Dokumentacija

4.3.3.1 getPavarde()

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

Gauti žmogaus pavardę.

Šis metodas turi būti įgyvendintas paveldėtose klasėse, kurios apibrėžia konkretų pavardžių tvarkymą.

Gražina

Žmogaus pavardė.

Realizuota `Studentas`.

4.3.3.2 getVardas()

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

Gauti žmogaus vardą.

Šis metodas turi būti įgyvendintas paveldėtose klasėse, kurios apibrėžia konkretų vardų tvarkymą.

Gražina

Žmogaus vardas.

Realizuota [Studentas](#).

4.3.4 Atributų Dokumentacija

4.3.4.1 pavarde_

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

Žmogaus pavardė. Paveldimos klasės gali tiesiogiai naudoti šį narį.

4.3.4.2 vardas_

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

Žmogaus vardas. Paveldimos klasės gali tiesiogiai naudoti šį narį.

Dokumentacija šiai klasei sugeneruota iš šio failo:

- [include/Zmogus.h](#)

skyrius 5

Failo Dokumentacija

5.1 include/Mylib.h Failo Nuoroda

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

5.2 Mylib.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #include <iostream>
00002 #include <iomanip>
00003 #include <string>
00004 #include <vector>
00005 #include <list>
00006
00007 #include <algorithm>    // sort() funkcijai
00008 #include <cctype>
00009
00010 #include <random>
00011 #include <chrono>
00012
00013 #include <fstream>
00014 #include <sstream>
00015
00016 #include <stdexcept>
00017
00018
00019 using std::endl;
00020 using std::cout;
00021 using std::cin;
00022
00023 using std::string;
00024 using std::vector;
00025 using std::stringstream;
00026 using std::to_string;
00027 using std::list;
```

```

00028
00029 using std::fixed;
00030 using std::setprecision;
00031 using std::setw;
00032 using std::left;
00033 using std::move;
00034
00035 using std::ifstream;
00036 using std::ofstream;
00037 using std::ostream;
00038 using std::istream;
00039
00040 using std::invalid_argument;
00041 using std::out_of_range;
00042 using std::runtime_error;
00043
00044 using std::random_device;
00045 using std::uniform_int_distribution;
00046
00047 using std::is_same_v;

```

5.3 include/Stud.h Failo Nuoroda

```

#include "Mylib.h"
#include "Studentas.h"

```

Funkcijos

- `template<typename Container >`
`void Duom_ivedimas` (Container &stud, string vardas, string pavarde)
Funkcija, skirta studento duomenų įvedimui rankiniu būdu.
- `template<typename Container >`
`void Info_ivedimas_ranka` (Container &stud, int n)
Surenkama bendra informacija apie studentų duomenis. Klausama studentų vardų bei pavardžių, koks įvertinimų įvedimas (rankinis ar generavimas) ir pagal tai įvykdoma.
- `template<typename Container >`
`void Duom_is_failo` (Container &stud)
Funkcija skirta nuskaityti studento įvertinimus iš failo.
- `void Stud_failu_generavimas` (int kiekis)
Funkcija, skirta sugeneruoti studentų duomenų failą su nurodytu įrašų kiekiu.
- `void Rez_antraste` (string pasirinkimas, ostream &out, string isvedimo_pasirinkimas)
Funkcija, skirta atspausdinti antraštei pagal vartoto įvertinimo pasirinkimą.
- `template<typename Container >`
`void Studentu_rusiavimas` (Container &stud, string pasirinkimas)
Funkcija skirta studentų rūšiavimui.
- `template<typename Container >`
`void SpausdinimasRez` (Container &stud, string isvedimo_pasirinkimas, string rez_pasirinkimas, string rusiavimo_p, string ivedimo_skaitymo_p)
Funkcija skirta rezultatams atspausdinti į terminalą (vartotojui pasirinkus 'T') arba įrašyti į failą (vartotojui pasirinkus 'F').
- `template<typename Container >`
`void Kategorijos_Priskirimas1` (Container &stud, Container &stud_Vargsiukai, Container &stud_Kietiakai)
Funkcija skirsto studentus į dvi grupes: "vargsiukus" ir "kietiakus".
- `template<typename Container >`
`void Kategorijos_Priskirimas2` (Container &stud, Container &stud_Vargsiukai)
Funkcija skirsto studentus į dvi grupes: "vargsiukus" ir likusius, pašalindama "vargsiukus" iš pradinio konteinerio.

- `template<typename Container >`
`void Kategorijos_Priskirimas3` (`Container &stud`, `Container &stud_Vargsiukai`)
Skirsto studentus į dvi grupes: "vargsiukus" ir "kietiokus", naudojant `std::stable_partition`.
- `template<typename Container >`
`void FailasPgaliKategorija` (`Container &studentai`, `string pasirinkimas`, `string isvedimo_pasirinkimas`, `string pav`)
Įrašo pateiktą studentų konteinerį į nurodytą failą.
- `string pasirinkimas_del_programos_vykdymo` ()
Leidžia vartotojui pasirinkti programos vykdymo režimą.
- `string pasirinkimas_del_duom_strukturos` ()
Leidžia vartotojui pasirinkti duomenų struktūrą.
- `string pasirinkimas_del_galutinio` ()
Leidžia vartotojui pasirinkti galutinio įvertinimo skaičiavimo metodą.
- `string pasirinkimas_del_rusiavimo` ()
Leidžia vartotojui pasirinkti rūšiavimo būdą.
- `string pasirinkimas_isvedimo` ()
Leidžia vartotojui pasirinkti, kur nori matyti rezultatą: terminale ar faile.
- `int pasirinkimas_del_strategijos` ()
Leidžia vartotojui pasirinkti strategiją studentų skirstymui į dvi grupes.
- `template<typename Container >`
`void Duom_tvarkymas` (`Container &stud`, `Container &stud_Vargsiukai`, `Container &stud_Kietiakai`, `string rez↵`
`_pasirinkimas`, `string rusiavimo_p`, `string isvedimo_pasirinkimas`, `string ivedimo_skaitymo_p`, `int kategorijos↵`
`_strategija`, `int kiekis`)
Apdoroja studentų duomenis, skirsto į grupes ir išsaugo į failus.

5.3.1 Funkcijos Dokumentacija

5.3.1.1 Duom_is_failo()

```
template<typename Container >
void Duom_is_failo (
    Container & stud)
```

Funkcija skirta nuskaityti studento įvertinimus iš failo.

Template Parameters

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

Parametrai

<i>stud</i>	Konteineris, į kurį bus įvedami studentų duomenys.
-------------	----------------------------------------------------

Pastaba

Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.3.1.2 Duom_ivedimas()

```
template<typename Container >
void Duom_ivedimas (
    Container & stud,
    string vardas,
    string pavarde)
```

Funkcija, skirta studento duomenų įvedimui rankiniu būdu.

Ši funkcija leidžia vartotojui ranka įvesti studento namų darbų įvertinimus ir egzamino rezultatą.

Template Parameters

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

Parametrai

<i>stud</i>	Konteineris, į kurį bus įvedami studentų duomenys.
<i>vardas</i>	Studentų vardas.
<i>pavarde</i>	Studentų pavardė.

Pastaba

Funkcija veikia tiek su vector<Studentas>, tiek su list<Studentas>.

5.3.1.3 Duom_tvarkymas()

```
template<typename Container >
void Duom_tvarkymas (
    Container & stud,
    Container & stud_Vargsiukai,
    Container & stud_Kietiakai,
    string rez_pasirinkimas,
    string rusiavimo_p,
    string isvedimo_pasirinkimas,
    string ivedimo_skaitymo_p,
    int kategorijos_strategija,
    int kiekis)
```

Apdoroja studentų duomenis, skirsto į grupes ir išsaugo į failus.

Ši funkcija vykdo šiuos veiksmus:

1. Spausdina pradinis studentų duomenis.
2. Skirsto studentus į dvi grupes (vargsiukai ir kietiakai) pagal pasirinktą strategiją.
3. Rūšiuoja studentų duomenis pagal vartotojo pasirinktą kriterijų.
4. Įrašo rezultatus į atskirus failus (Vargsiukai.txt ir Kietiakai.txt).

Template Parameters

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

Parametrai

<i>stud</i>	Pagrindinis konteineris, kuriame yra visi studentų duomenys.
<i>stud_Vargsiukai</i>	Konteineris, kuriame bus saugomi studentai, nepasiekę nustatyto rezultato (vargsiukai).
<i>stud_Kietiakai</i>	Konteineris, kuriame bus saugomi geriausi studentai (kietiakai).
<i>rez_pasirinkimas</i>	Rezultatų išvedimo būdas (pvz., į failą ar terminalą).
<i>rusiavimo_p</i>	Pasirinkimas nurodo, pagal ką studentų konteineris bus rūšiuojamas (pvz.: pagal vardą ir pavardę, pavardę ir vardą, galutinį įvertinimą mažėjančia tvarka, pagal galutinį įvertinimą didėjančia tvarka).
<i>isvedimo_pasirinkimas</i>	Rezultatų failo pavadinimas (pvz., Rezultatai.txt).
<i>ivedimo_skaitymo_p</i>	Įvedimo failo pavadinimas.
<i>kategorijos_strategija</i>	Pasirinkta strategija studentų skirstymui į grupes (1, 2 arba 3).
<i>kiekis</i>	Studentų įrašų skaičius (naudojamas rezultatų laikui spausdinti).

Pastaba

Funkcija veikia tiek su vector<Studentas>, tiek su list<Studentas>.

5.3.1.4 FailasPgalKategorija()

```
template<typename Container >
void FailasPgalKategorija (
    Container & studentai,
    string pasirinkimas,
    string isvedimo_pasirinkimas,
    string pav)
```

Įrašo pateiktą studentų konteinerį į nurodytą failą.

Ši funkcija atidaro failą pagal nurodytą pavadinimą ir įrašo studentų duomenis iš pateikto konteinerio. Taip pat antraštė generuojama naudojant funkciją Rez_antraste.

Template Parameters

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

Parametrai

<i>studentai</i>	Studentų konteineris, kuris bus įrašytas į failą.
<i>pasirinkimas</i>	Pasirinkimas nurodantis pagal ką buvo skaičiuotas galutinis rezultatas (vidurkis ar mediana). (Skirta antraštei)
<i>isvedimo_pasirinkimas</i>	Rezultatų išvedimo pasirinkimas (T - terminalas, F - failas). (Skirta antraštei)
<i>pav</i>	Failo pavadinimas, į kurį bus įrašomi duomenys.

Pastaba

Funkcija veikia tiek su vector<Studentas>, tiek su list<Studentas>.

5.3.1.5 Info_ivedimas_ranka()

```
template<typename Container >
void Info_ivedimas_ranka (
    Container & stud,
    int n)
```

Surenkama bendra informacija apie studentų duomenis. Klausama studentų vardų bei pavardžių, koks įvertinimų įvedimas (rankinis ar generavimas) ir pagal tai įvykdoma.

Template Parameters

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

Parametrai

<i>stud</i>	Konteineris, į kurį bus įvedami studentų duomenys.
<i>n</i>	Studentų skaičius, kurį vartotojas nori įtraukti.

Pastaba

Funkcija veikia tiek su vector<Studentas>, tiek su list<Studentas>.

5.3.1.6 Kategorijos_Priskirimas1()

```
template<typename Container >
void Kategorijos_Priskirimas1 (
    Container & stud,
    Container & stud_Vargsiukai,
    Container & stud_Kietiakai)
```

Funkcija skirsto studentus į dvi grupes: "vargsiukus" ir "kietiakus".

Funkcija patikrina kiekvieno studento galutinį pažymį (getGalutinis()) ir priskiria jį į vieną iš dviejų grupių:

- stud_Vargsiukai – jei galutinis pažymys mažesnis už 5.0.
- stud_Kietiakai – jei galutinis pažymys lygus arba didesnis už 5.0.

Template Parameters

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

Parametrai

<i>stud</i>	Pradinė studentų konteineris, kurį reikia suskirstyti.
<i>stud_Vargsiukai</i>	konteineris, į kurią bus priskirti "vargsiukai".
<i>stud_Kietiakai</i>	konteineris, į kurią bus priskirti "kietiakai".

Pastaba

Funkcija nekeičia pradinio stud konteinerio, o tik kopijuoja elementus į naujas grupes. Funkcija veikia tiek su vector<Studentas>, tiek su list<Studentas>.

5.3.1.7 Kategorijos_Priskirimas2()

```
template<typename Container >
void Kategorijos_Priskirimas2 (
    Container & stud,
    Container & stud_Vargsiukai)
```

Funkcija skirsto studentus į dvi grupes: "vargsiukus" ir likusius, pašalindama "vargsiukus" iš pradinio konteinerio.

Funkcija rūšiuoja studentų konteinerį mažėjančia tvarka pagal galutinį pažymį (getGalutinis()). Tada perkelia studentus, kurių galutinis pažymys mažesnis už 5.0, į "vargsiukų" konteinerį (stud_Vargsiukai) ir pašalina juos iš pradinio konteinerio (stud).

Template Parameters

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

Parametrai

<i>stud</i>	Pradinis studentų konteineris, kuri bus modifikuojama (iš jos pašalinami "vargsiukai").
<i>stud_Vargsiukai</i>	Konteineris, į kurį bus perkelti "vargsiukai".

Pastaba

Pradinis konteineris *stud* bus modifikuota – joje liks tik studentai, kurių galutinis pažymys ≥ 5.0 . Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.3.1.8 Kategorijos_Priskirimas3()

```
template<typename Container >
void Kategorijos_Priskirimas3 (
    Container & stud,
    Container & stud_Vargsiukai)
```

Skirsto studentus į dvi grupes: "vargsiukus" ir "kietiokus", naudojant `std::stable_partition`.

Ši funkcija naudoja `std::stable_partition` metodą, kad stabiliai atskirtų studentus su galutiniu švertinimu mažesniu nei 5.0 ("vargsiukai") nuo tų, kurių pažymys ≥ 5.0 ("kietiakai"). "Vargsiukai" perkelti į atskirą konteinerį, o pradinis konteineris modifikuojama, pašalinant "vargsiukus".

Template Parameters

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

Parametrai

<i>stud</i>	Pradinis studentų konteineris, iš kurios bus pašalinti "vargsiukai".
<i>stud_Vargsiukai</i>	Konteineris, į kurį bus perkelti "vargsiukai".

Pastaba

Pradinis konteineris *stud* modifikuojamas – iš jo pašalinami "vargsiukai". "Vargsiukai" yra perkelti į `stud_Vargsiukai`, išsaugant jų pradinės eiles. Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.3.1.9 pasirinkimas_del_duom_strukturos()

```
string pasirinkimas_del_duom_strukturos ()
```

Leidžia vartotojui pasirinkti duomenų struktūrą.

Funkcija pateikia vartotojui klausimą, kokią duomenų struktūrą norima naudoti, ir užtikrina, kad įvestis būtų tinkama (V - vektorius, S - sąrašas).

Gražina

Pasirinktas duomenų struktūros tipas kaip string (V arba S).

5.3.1.10 pasirinkimas_del_galutinio()

```
string pasirinkimas_del_galutinio ()
```

Leidžia vartotojui pasirinkti galutinio įvertinimo skaičiavimo metodą.

Funkcija pateikia klausimą vartotojui, ar galutinį įvertinimą skaičiuoti pagal vidurkį, ar pagal medianą. Užtikrina, kad įvestis būtų tinkama (V - vidurkis, M - mediana).

Gražina

Pasirinktas galutinio įvertinimo metodas kaip string (V arba M).

5.3.1.11 pasirinkimas_del_programos_vykdymo()

```
string pasirinkimas_del_programos_vykdymo ()
```

Leidžia vartotojui pasirinkti programos vykdymo režimą.

Funkcija pateikia vartotojui klausimą, kokį veikimo režimą pasirinkti, ir užtikrina, kad įvestis būtų viena iš galimų reikšmių: I, N, S, T, arba D. Jei įvestis netinkama, funkcija kartoja klausimą tol, kol gaunama tinkama reikšmė.

Gražina

Pasirinktas veikimo režimas kaip string (I, N, S, T, arba D).

5.3.1.12 pasirinkimas_del_rusiavimo()

```
string pasirinkimas_del_rusiavimo ()
```

Leidžia vartotojui pasirinkti rūšiavimo būdą.

Funkcija pateikia klausimą vartotojui, pagal ką reikia surūšiuoti studentus. Užtikrina, kad įvestis būtų tinkama (VP, PV, GM, GD).

Gražina

Pasirinktas rūšiavimo būdas kaip string (VP, PV, GM arba GD).

5.3.1.13 pasirinkimas_del_strategijos()

```
int pasirinkimas_del_strategijos ()
```

Leidžia vartotojui pasirinkti strategiją studentų skirstymui į dvi grupes.

Funkcija pateikia klausimą vartotojui apie pasirenkamą strategiją. Galimos strategijos:

- '1' - Pirmoji strategija,
- '2' - Antroji strategija,
- '3' - Trečioji strategija.

Gražina

Pasirinkta strategija kaip sveikasis skaičius (1, 2 arba 3).

5.3.1.14 pasirinkimas_isvedimo()

```
string pasirinkimas_isvedimo ()
```

Leidžia vartotojui pasirinkti, kur nori matyti rezultatą: terminale ar faile.

Funkcija pateikia klausimą vartotojui, kur norėtų jis matyti rezultatą. Vartotojas gali pasirinkti:

- 'T' - terminale,
- 'F' - faile.

Gražina

Pasirinktas rezultato pateikimo būdas kaip string ('T' arba 'F').

5.3.1.15 Rez_antraste()

```
void Rez_antraste (
    string pasirinkimas,
    ostream & out,
    string isvedimo_pasirinkimas)
```

Funkcija, skirta atspausdinti antraštei pagal vartoto įvertinimo pasirinkimą.

Parametrai

<i>pasirinkimas</i>	Pasirinkimas nurodantis pagal ką buvo skaičiuotas galutinis rezultatas (vidurkis ar mediana).
<i>out</i>	Srautas, į kurį spausdinama (pvz., terminalas ar failas).
<i>isvedimo_pasirinkimas</i>	Rezultatų išvedimo pasirinkimas (T - terminalas, F - failas).

5.3.1.16 SpausdinimasRez()

```
template<typename Container >
void SpausdinimasRez (
    Container & stud,
    string isvedimo_pasirinkimas,
    string rez_pasirinkimas,
    string rusiavimo_p,
    string ivedimo_skaitymo_p)
```

Funkcija skirta rezultatams atspausdinti į terminalą (vartotojui pasirinkus 'T') arba įrašyti į failą (vartotojui pasirinkus 'F').

Template Parameters

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

Parametrai

<i>isvedimo_pasirinkimas</i>	Rezultatų išvedimo pasirinkimas (T - terminalas, F - failas).
<i>rez_pasirinkimas</i>	Pasirinkimas pagal ką skaičiuotas galutinis rezultatas (vidurkis ar mediana).
<i>rusiavimo_p</i>	Pasirinkimas nurodo, pagal ką studentų konteineris bus rūšiuojamas (pvz.: pagal vardą ir pavardę, pavardę ir vardą, galutinį įvertinimą mažėjančia tvarka, pagal galutinį įvertinimą didėjančia tvarka).
<i>ivedimo_skaitymo_p</i>	Pasirinkimas, kuris tikrinamas ar lygus T - Testavimui, tuo atveju bendras visų studentų konteineris yra atspausdinamas į failą Rez.txt.

Pastaba

Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.3.1.17 Stud_failu_generavimas()

```
void Stud_failu_generavimas (
    int kiekis)
```

Funkcija, skirta sugeneruoti studentų duomenų failą su nurodytu įrašų kiekiu.

Parametrai

<i>kiekis</i>	Failo įrašų kiekis.
---------------	---------------------

5.3.1.18 Studentu_rusiavimas()

```
template<typename Container >
void Studentu_rusiavimas (
    Container & stud,
    string pasirinkimas)
```

Funkcija skirta studentų rūšiavimui.

Template Parameters

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

Parametrai

<i>pasirinkimas</i>	Pasirinkimas nurodo, pagal ką studentų konteineris bus rūšiuojamas (pvz.: pagal vardą ir pavardę, pavardę ir vardą, galutinį įvertinimą mažėjančia tvarka, pagal galutinį įvertinimą didėjančia tvarka).
---------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Pastaba

Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.4 Stud.h

Eiti į šio failo dokumentaciją.

```
00001 #ifndef STUD_H_INCLUDED
00002 #define STUD_H_INCLUDED
00003
00004 #include "Mylib.h"
00005 #include "Studentas.h"
00006
00007
00008 // Funkcija, skirta studento duomenų įvedimui rankiniu būdu.
00021 template <typename Container>
00022 void Duom_ivedimas(Container &stud, string vardas, string pavarde);
00023
00024 // Funkcija, kurioje klausiama kiek studentų vartotojas norėtų įtraukti, klausiama studentų vardų bei
    pavardžių,
00025 // klausiama koks įvertinimų įvedimas(rankinis ar generavimas) ir pagal tai įvykdoma.
00034 template <typename Container>
00035 void Info_ivedimas_ranka(Container &stud, int n);
00036
00037 // Funkcija skirta nuskaityti studento įvertinimus iš failo.
00045 template <typename Container>
00046 void Duom_is_failo(Container &stud);
00047
00048 // Funkcija skirta failo generavimui pagal įrašų kiekį.
00053 void Stud_failu_generavimas(int kiekis);
00054
00055 // Funkcija, skirta atspausdinti antraštei pagal vartoto įvertinimo pasirinkimą.
00062 void Rez_antraste(string pasirinkimas, ostream &out, string isvedimo_pasirinkimas);
00063
00064 // Funkcija skirta studentų rūšiavimui.
00072 template <typename Container>
00073 void Studentu_rusivimas(Container &stud, string pasirinkimas);
00074
00075 // Funkcija skirta rezultatams atspausdinti į terminalą (vartotojui pasirinkus 'T') arba įrašyti į
    failą (vartotojui pasirinkus 'F').
00086 template <typename Container>
00087 void SpausdinimasRez(Container &stud, string isvedimo_pasirinkimas, string rez_pasirinkimas, string
    rusiavimo_p, string ivedimo_skaitymo_p);
00088
00089 // Funkcija, skirta sukurti du naujus kontainerius vargšiukams ir kietiakams, taip studentai yra
    surūšiuojami į dvi grupes.
00105 template <typename Container>
00106 void Kategorijos_Priskirimas1(Container &stud, Container &stud_Vargsiukai, Container &stud_Kietiakai);
00107
00108 // Funkcija, kuri surūšiuoja studentus į dvi grupes, jei studento įvertinimas < 5.0, priskiriamas
    "Vargšiukų" kontaineriui
00109 // ir studentas ištrinamas iš bendro. Taip bendrame liks tik tie studentai, kurių įvertinimas >= 5.0.
00125 template <typename Container>
00126 void Kategorijos_Priskirimas2(Container &stud, Container &stud_Vargsiukai);
00127
00128 // Funkcija, kuri surūšiuoja studentus į dvi grupes, padaryta remiantis 2 strategija
    (Kategorijos_Priskirimas2). Pritaikyta std::stable_partition() funkcija.
00144 template <typename Container>
00145 void Kategorijos_Priskirimas3(Container &stud, Container &stud_Vargsiukai);
00146
00147 // Funkcija, kuri įrašo į failą pateiktą kontainerį.
00162 template <typename Container>
00163 void FailasPgalkategorija(Container &studentai, string pasirinkimas, string isvedimo_pasirinkimas,
    string pav);
00164
00165 // Funkcija, skirta vartotojui pasirinkti programos vykdymą (Įvesti - I, Nuskaityti - N, Sugeneruoti -
    S, Testuoti - T).
00175 string pasirinkimas_del_programos_vykdyto();
00176
00177 // Funkcija, skirta vartotojui pasirinkti duomenų struktūrą (V - vektorius, S - sąrašas).
00186 string pasirinkimas_del_duom_strukturos();
00187
00188 // Funkcija, skirta vartotojui pasirinkti galutinį įvertinimą, pagal vidurkį arba pagal medianą.
00197 string pasirinkimas_del_galutinio();
00198
00199 // Funkcija, skirta vartotojui pasirinkti pagal ką reikia surūšiuoti studentus.
00208 string pasirinkimas_del_rusiavimo();
00209
00210 // Funkcija, skirta vartotojui pasirinkti kur nori matyti rezultata, ar terminale, ar faile.
00220 string pasirinkimas_isvedimo();
00221
00222 // Funkcija, skirta vartotojui pasirinkti pagal kurią strategiją norimą skirstyti studentus į dvi
    grupes (1, 2, 3).
00234 int pasirinkimas_del_strategijos();
00235
00236 // Funkcija skirta duomenų tvarkymui, tai rezultatų įrašymui, kategorijos priskirimui, naujų failų
    sukūrimui.
00261 template <typename Container>
```

```

00262 void Duom_tvarkymas(Container &stud, Container &stud_Vargsiukai, Container &stud_Kietiakai, string
      rez_pasirinkimas, string rusiavimo_p, string isvedimo_pasirinkimas, string ivedimo_skaitymo_p, int
      kategorijos_strategija, int kiekis);
00263
00264 #endif

```

5.5 include/Studentas.h Failo Nuoroda

```

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

```

Klasės

- class [Studentas](#)

Žmogaus išvestinė klasė, skirta studentų duomenims saugoti tvarkyti.

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
00006 #include <iostream>
00007 #include <iomanip>
00008 #include <string>
00009 #include <vector>
00010
00011 using std::endl;
00012 using std::cout;
00013 using std::cin;
00014
00015 using std::string;
00016 using std::vector;
00017
00025 class Studentas : public Zmogus{
00026 private:
00027     vector<double> nd_;
00028     double egz_;
00029     double galutinis_;
00031 public:
00032     // Konstruktoriai
00033
00038     Studentas(){};
00039
00047     Studentas(const string& vardas, const string& pavarde, const vector<double>& nd, double egz);
00048
00057     Studentas(const string& vardas, const string& pavarde, int kiekis);
00058
00059     // Kopijavimo Konstruktoius
00065     Studentas(const Studentas& saltinis);
00066
00067     // Destruktorius
00073     ~Studentas();
00074
00075     // Overloaded Assignment Operator - Kopijavimo priskirymo operatorius
00083     Studentas& operator=(const Studentas& saltinis);
00084
00095     friend std::ostream& operator<<(std::ostream& os, const Studentas& s);
00096
00109     friend std::istream& operator>>(std::istream& is, Studentas& s);
00110
00111     // set'eriai

```

```

00112
00117 void setVardas(const string& vardas) {vardas_ = vardas; }
00118
00123 void setPavarde(const string& pavarde) {pavarde_ = pavarde; }
00124
00129 void setNd(const vector<double>& nd) {nd_ = nd; }
00130
00135 void setEgz(const double& egz) {egz_ = egz; }
00136
00141 void setGalutinis(const double& ivertinimas){galutinis_ = ivertinimas; }
00142
00143 // get'eriai
00144
00149 string getVardas() const override { return vardas_; }
00150
00155 string getPavarde() const override { return pavarde_; }
00156
00161 inline double getEgz() const {return egz_; }
00162
00167 inline vector<double> getNd() const { return nd_; }
00168
00173 inline double getGalutinis() const { return galutinis_; }
00174
00175
00182 void Ivertinimas_vid();
00183
00190 void Ivertinimas_med();
00191
00192
00193
00194 };
00195
00196 #endif

```

5.7 include/timer.h Failo Nuoroda

```
#include <chrono>
```

Klasės

- class [Timer](#)

Laiko matavimo klasė, skirta skaičiuoti programos veikimo laiką.

5.8 timer.h

[Eiti į šio failo dokumentaciją.](#)

```

00001 #include <chrono>
00008 class Timer {
00009     private:
00010         // panaudojame using
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 <string>
```

Klasės

- class [Zmogus](#)

Abstrakti bazinė klasė, skirta žmonių duomenims valdyti.

5.10 Zmogus.h

Eiti į šio failo dokumentaciją.

```
00001 #ifndef Zmogus_H_INCLUDED
00002 #define Zmogus_H_INCLUDED
00003
00004
00005 #include <string>
00006 using std::string;
00016 class Zmogus{
00017     protected:
00018         string vardas_;
00019         string pavarde_;
00026         Zmogus(){};
00027
00033         Zmogus(const string& vardas, const string& pavarde)
00034             : vardas_(vardas), pavarde_(pavarde) {};
00035
00036     public:
00037
00045         virtual string getVardas() const = 0;
00046
00055         virtual string getPavarde() const = 0;
00056
00062         virtual ~Zmogus(){};
00063
00064 };
00065
00066 #endif
```

5.11 src/Stud.cpp Failo Nuoroda

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

Funkcijos

- template<typename Container >
void [Duom_ivedimas](#) (Container &stud, string vardas, string pavarde)
Funkcija, skirta studento duomenų įvedimui rankiniu būdu.
- template void [Duom_ivedimas](#)< vector< [Studentas](#) > > (vector< [Studentas](#) > &, string, string)
- template void [Duom_ivedimas](#)< list< [Studentas](#) > > (list< [Studentas](#) > &, string, string)
- template<typename Container >
void [Info_ivedimas_ranka](#) (Container &stud, int n)
Surenkama bendra informacija apie studentų duomenis. Klausiama studentų vardų bei pavardžių, koks įvertinimų įvedimas (rankinis ar generavimas) ir pagal tai įvykdoma.
- template void [Info_ivedimas_ranka](#)< vector< [Studentas](#) > > (vector< [Studentas](#) > &, int)
- template void [Info_ivedimas_ranka](#)< list< [Studentas](#) > > (list< [Studentas](#) > &, int)
- template<typename Container >
void [Duom_is_failo](#) (Container &stud)
Funkcija skirta nuskaityti studento įvertinimus iš failo.
- template void [Duom_is_failo](#)< vector< [Studentas](#) > > (vector< [Studentas](#) > &)

- template void `Duom_is_failo< list< Studentas > >` (list< `Studentas` > &)
- void `Stud_failu_generavimas` (int kiekis)
Funkcija, skirta sugeneruoti studentų duomenų failą su nurodytu įrašų kiekiu.
- void `Rez_antraste` (string pasirinkimas, ostream &out, string isvedimo_pasirinkimas)
Funkcija, skirta atspausdinti antraštei pagal vartotojo įvertinimo pasirinkimą.
- template<typename Container >
void `Studentu_rusiavimas` (Container &stud, string pasirinkimas)
Funkcija skirta studentų rūšiavimui.
- template<typename Container >
void `SpausdinimasRez` (Container &stud, string isvedimo_pasirinkimas, string rez_pasirinkimas, string rusiavimo_p, string ivedimo_skaitymo_p)
Funkcija skirta rezultatams atspausdinti į terminalą (vartotojui pasirinkus 'T') arba įrašyti į failą (vartotojui pasirinkus 'F').
- template void `SpausdinimasRez< vector< Studentas > >` (vector< `Studentas` > &, string, string, string, string)
- template void `SpausdinimasRez< list< Studentas > >` (list< `Studentas` > &, string, string, string, string)
- template<typename Container >
void `Kategorijos_Priskirimas1` (Container &stud, Container &stud_Vargsiukai, Container &stud_Kietiakai)
Funkcija skirsto studentus į dvi grupes: "vargsiukus" ir "kietiakus".
- template<typename Container >
void `Kategorijos_Priskirimas2` (Container &stud, Container &stud_Vargsiukai)
Funkcija skirsto studentus į dvi grupes: "vargsiukus" ir likusius, pašalindama "vargsiukus" iš pradinio konteinerio.
- template<typename Container >
void `Kategorijos_Priskirimas3` (Container &stud, Container &stud_Vargsiukai)
Skirsto studentus į dvi grupes: "vargsiukus" ir "kietiakus", naudojant std::stable_partition.
- template<typename Container >
void `FailasPgaiKategorija` (Container &studentai, string pasirinkimas, string isvedimo_pasirinkimas, string pav)
Įrašo pateiktą studentų konteinerį į nurodytą failą.
- string `pasirinkimas_del_programos_vykdymo` ()
Leidžia vartotojui pasirinkti programos vykdymo režimą.
- string `pasirinkimas_del_duom_strukturos` ()
Leidžia vartotojui pasirinkti duomenų struktūrą.
- string `pasirinkimas_del_galutinio` ()
Leidžia vartotojui pasirinkti galutinio įvertinimo skaičiavimo metodą.
- string `pasirinkimas_del_rusiavimo` ()
Leidžia vartotojui pasirinkti rūšiavimo būdą.
- string `pasirinkimas_isvedimo` ()
Leidžia vartotojui pasirinkti, kur nori matyti rezultatą: terminale ar faile.
- int `pasirinkimas_del_strategijos` ()
Leidžia vartotojui pasirinkti strategiją studentų skirstymui į dvi grupes.
- template<typename Container >
void `Duom_tvarkymas` (Container &stud, Container &stud_Vargsiukai, Container &stud_Kietiakai, string rez_↵
_pasirinkimas, string rusiavimo_p, string isvedimo_pasirinkimas, string ivedimo_skaitymo_p, int kategorijos_↵
_strategija, int kiekis)
Apdoroja studentų duomenis, skirsto į grupes ir išsaugo į failus.
- template void `Duom_tvarkymas< vector< Studentas > >` (vector< `Studentas` > &stud, vector< `Studentas` > &stud_Vargsiukai, vector< `Studentas` > &stud_Kietiakai, string rez_pasirinkimas, string rusiavimo_↵
p, string isvedimo_pasirinkimas, string ivedimo_skaitymo_p, int kategorijos_strategija, int kiekis)
- template void `Duom_tvarkymas< list< Studentas > >` (list< `Studentas` > &stud, list< `Studentas` > &stud_Vargsiukai, list< `Studentas` > &stud_Kietiakai, string rez_pasirinkimas, string rusiavimo_p, string isvedimo_pasirinkimas, string ivedimo_skaitymo_p, int kategorijos_strategija, int kiekis)

5.11.1 Funkcijos Dokumentacija

5.11.1.1 Duom_is_failo()

```
template<typename Container >
void Duom_is_failo (
    Container & stud)
```

Funkcija skirta nuskaityti studento įvertinimus iš failo.

Template Parameters

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

Parametrai

<i>stud</i>	Konteineris, į kurį bus įvedami studentų duomenys.
-------------	----------------------------------------------------

Pastaba

Funkcija veikia tiek su vector<Studentas>, tiek su list<Studentas>.

5.11.1.2 Duom_is_failo< list< Studentas > >()

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

5.11.1.3 Duom_is_failo< vector< Studentas > >()

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

5.11.1.4 Duom_ivedimas()

```
template<typename Container >
void Duom_ivedimas (
    Container & stud,
    string vardas,
    string pavarde)
```

Funkcija, skirta studento duomenų įvedimui rankiniu būdu.

Ši funkcija leidžia vartotojui ranka įvesti studento namų darbų įvertinimus ir egzamino rezultatą.

Template Parameters

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

Parametrai

<i>stud</i>	Konteineris, į kurį bus įvedami studentų duomenys.
<i>vardas</i>	Studentų vardas.
<i>pavarde</i>	Studentų pavardė.

Pastaba

Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.11.1.5 Duom_ivedimas< list< Studentas > >()

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

5.11.1.6 Duom_ivedimas< vector< Studentas > >()

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

5.11.1.7 Duom_tvarkymas()

```
template<typename Container >
void Duom_tvarkymas (
    Container & stud,
    Container & stud_Vargsiukai,
    Container & stud_Kietiakai,
    string rez_pasirinkimas,
    string rusiavimo_p,
    string isvedimo_pasirinkimas,
    string ivedimo_skaitymo_p,
    int kategorijos_strategija,
    int kiekis)
```

Apdoroja studentų duomenis, skirsto į grupes ir išsaugo į failus.

Ši funkcija vykdo šiuos veiksmus:

1. Spausdina pradinis studentų duomenis.
2. Skirsto studentus į dvi grupes (vargsiukai ir kietiakai) pagal pasirinktą strategiją.
3. Rūšiuoja studentų duomenis pagal vartotojo pasirinktą kriterijų.
4. Įrašo rezultatus į atskirus failus (Vargsiukai.txt ir Kietiakai.txt).

Template Parameters

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

Parametrai

<i>stud</i>	Pagrindinis konteineris, kuriame yra visi studentų duomenys.
<i>stud_Vargsiukai</i>	Konteineris, kuriame bus saugomi studentai, nepasiekę nustatyto rezultato (vargsiukai).
<i>stud_Kietiakai</i>	Konteineris, kuriame bus saugomi geriausi studentai (kietiakai).
<i>rez_pasirinkimas</i>	Rezultatų išvedimo būdas (pvz., į failą ar terminalą).
<i>rusiavimo_p</i>	Pasirinkimas nurodo, pagal ką studentų konteineris bus rūšiuojamas (pvz.: pagal vardą ir pavardę, pavardę ir vardą, galutinį įvertinimą mažėjančia tvarka, pagal galutinį įvertinimą didėjančia tvarka).
<i>isvedimo_pasirinkimas</i>	Rezultatų failo pavadinimas (pvz., Rezultatai.txt).
<i>ivedimo_skaitymo_p</i>	Įvedimo failo pavadinimas.
<i>kategorijos_strategija</i>	Pasirinkta strategija studentų skirstymui į grupes (1, 2 arba 3).
<i>kiekis</i>	Studentų įrašų skaičius (naudojamas rezultatų laikui spausdinti).

Pastaba

Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.11.1.8 Duom_tvarkymas< list< Studentas > >()

```
template void Duom_tvarkymas< list< Studentas > > (
    list< Studentas > & stud,
    list< Studentas > & stud_Vargsiukai,
    list< Studentas > & stud_Kietiakai,
    string rez_pasirinkimas,
    string rusiavimo_p,
    string isvedimo_pasirinkimas,
    string ivedimo_skaitymo_p,
    int kategorijos_strategija,
    int kiekis)
```

5.11.1.9 Duom_tvarkymas< vector< Studentas > >()

```
template void Duom_tvarkymas< vector< Studentas > > (
    vector< Studentas > & stud,
    vector< Studentas > & stud_Vargsiukai,
    vector< Studentas > & stud_Kietiakai,
    string rez_pasirinkimas,
    string rusiavimo_p,
    string isvedimo_pasirinkimas,
    string ivedimo_skaitymo_p,
    int kategorijos_strategija,
    int kiekis)
```

5.11.1.10 FailasPgalKategorija()

```
template<typename Container >
void FailasPgalKategorija (
    Container & studentai,
    string pasirinkimas,
    string isvedimo_pasirinkimas,
    string pav)
```

Įrašo pateiktą studentų konteinerį į nurodytą failą.

Ši funkcija atidaro failą pagal nurodytą pavadinimą ir įrašo studentų duomenis iš pateikto konteinerio. Taip pat antraštė generuojama naudojant funkciją Rez_antraste.

Template Parameters

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

Parametrai

<i>studentai</i>	Studentų konteineris, kuris bus įrašytas į failą.
<i>pasirinkimas</i>	Pasirinkimas nurodantis pagal ką buvo skaičiuotas galutinis rezultatas (vidurkis ar mediana). (Skirta antraštei)
<i>isvedimo_pasirinkimas</i>	Rezultatų išvedimo pasirinkimas (T - terminalas, F - failas). (Skirta antraštei)
<i>pav</i>	Failo pavadinimas, į kurį bus įrašomi duomenys.

Pastaba

Funkcija veikia tiek su vector<Studentas>, tiek su list<Studentas>.

5.11.1.11 Info_ivedimas_ranka()

```
template<typename Container >
void Info_ivedimas_ranka (
    Container & stud,
    int n)
```

Surenkama bendra informacija apie studentų duomenis. Klausama studentų vardų bei pavardžių, koks įvertinimų įvedimas (rankinis ar generavimas) ir pagal tai įvykdoma.

Template Parameters

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

Parametrai

<i>stud</i>	Konteineris, į kurį bus įvedami studentų duomenys.
<i>n</i>	Studentų skaičius, kurį vartotojas nori įtraukti.

Pastaba

Funkcija veikia tiek su vector<Studentas>, tiek su list<Studentas>.

5.11.1.12 Info_ivedimas_ranka< list< Studentas > >()

```
template void Info_ivedimas_ranka< list< Studentas > > (
    list< Studentas > & ,
    int )
```

5.11.1.13 Info_ivedimas_ranka< vector< Studentas > >()

```
template void Info_ivedimas_ranka< vector< Studentas > > (
    vector< Studentas > & ,
    int )
```

5.11.1.14 Kategorijos_Priskirimas1()

```
template<typename Container >
void Kategorijos_Priskirimas1 (
    Container & stud,
    Container & stud_Vargsiukai,
    Container & stud_Kietiakai)
```

Funkcija skirsto studentus į dvi grupes: "vargsiukus" ir "kietiakus".

Funkcija patikrina kiekvieno studento galutinį pažymį (getGalutinis()) ir priskiria jį į vieną iš dviejų grupių:

- stud_Vargsiukai – jei galutinis pažymys mažesnis už 5.0.
- stud_Kietiakai – jei galutinis pažymys lygus arba didesnis už 5.0.

Template Parameters

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

Parametrai

<i>stud</i>	Pradinė studentų konteineris, kurį reikia suskirstyti.
<i>stud_Vargsiukai</i>	konteineris, į kurią bus priskirti "vargsiukai".
<i>stud_Kietiakai</i>	konteineris, į kurią bus priskirti "kietiakai".

Pastaba

Funkcija nekeičia pradinio stud konteinerio, o tik kopijuoja elementus į naujas grupes. Funkcija veikia tiek su vector<Studentas>, tiek su list<Studentas>.

5.11.1.15 Kategorijos_Priskirimas2()

```
template<typename Container >
void Kategorijos_Priskirimas2 (
    Container & stud,
    Container & stud_Vargsiukai)
```

Funkcija skirsto studentus į dvi grupes: "vargsiukus" ir likusius, pašalindama "vargsiukus" iš pradinio konteinerio.

Funkcija rūšiuoja studentų konteinerį mažėjančia tvarka pagal galutinį pažymį (getGalutinis()). Tada perkelia studentus, kurių galutinis pažymys mažesnis už 5.0, į "vargsiukų" konteinerį (stud_Vargsiukai) ir pašalina juos iš pradinio konteinerio (stud).

Template Parameters

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

Parametrai

<i>stud</i>	Pradinis studentų konteineris, kuri bus modifikuojama (iš jos pašalinami "vargsiukai").
<i>stud_Vargsiukai</i>	Konteineris, į kurį bus perkelti "vargsiukai".

Pastaba

Pradinis konteineris `stud` bus modifikuota – joje liks tik studentai, kurių galutinis pažymys ≥ 5.0 . Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.11.1.16 Kategorijos_Priskirimas3()

```
template<typename Container >
void Kategorijos_Priskirimas3 (
    Container & stud,
    Container & stud_Vargsiukai)
```

Skirsto studentus į dvi grupes: "vargsiukus" ir "kietiokus", naudojant `std::stable_partition`.

Ši funkcija naudoja `std::stable_partition` metodą, kad stabiliai atskirtų studentus su galutiniu švertinimu mažesniu nei 5.0 ("vargsiukai") nuo tų, kurių pažymys ≥ 5.0 ("kietiokus"). "Vargsiukai" perkelti į atskirą konteinerį, o pradinis konteineris modifikuojama, pašalinant "vargsiukus".

Template Parameters

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

Parametrai

<i>stud</i>	Pradinis studentų konteineris, iš kurios bus pašalinti "vargsiukai".
<i>stud_Vargsiukai</i>	Konteineris, į kurį bus perkelti "vargsiukai".

Pastaba

Pradinis konteineris `stud` modifikuojamas – iš jo pašalinami "vargsiukai". "Vargsiukai" yra perkelti į `stud_Vargsiukai`, išsaugant jų pradines eiles. Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.11.1.17 pasirinkimas_del_duom_strukturos()

```
string pasirinkimas_del_duom_strukturos ()
```

Leidžia vartotojui pasirinkti duomenų struktūrą.

Funkcija pateikia vartotojui klausimą, kokią duomenų struktūrą norima naudoti, ir užtikrina, kad įvestis būtų tinkama (V - vektorius, S - sąrašas).

Gražina

Pasirinktas duomenų struktūros tipas kaip string (V arba S).

5.11.1.18 pasirinkimas_del_galutinio()

```
string pasirinkimas_del_galutinio ()
```

Leidžia vartotojui pasirinkti galutinio įvertinimo skaičiavimo metodą.

Funkcija pateikia klausimą vartotojui, ar galutinį įvertinimą skaičiuoti pagal vidurkį, ar pagal medianą. Užtikrina, kad įvestis būtų tinkama (V - vidurkis, M - mediana).

Gražina

Pasirinktas galutinio įvertinimo metodas kaip string (V arba M).

5.11.1.19 pasirinkimas_del_programos_vykdymo()

```
string pasirinkimas_del_programos_vykdymo ()
```

Leidžia vartotojui pasirinkti programos vykdymo režimą.

Funkcija pateikia vartotojui klausimą, kokį veikimo režimą pasirinkti, ir užtikrina, kad įvestis būtų viena iš galimų reikšmių: I, N, S, T, arba D. Jei įvestis netinkama, funkcija kartoja klausimą tol, kol gaunama tinkama reikšmė.

Gražina

Pasirinktas veikimo režimas kaip string (I, N, S, T, arba D).

5.11.1.20 pasirinkimas_del_rusivimo()

```
string pasirinkimas_del_rusivimo ()
```

Leidžia vartotojui pasirinkti rūšiavimo būdą.

Funkcija pateikia klausimą vartotojui, pagal ką reikia surūšiuoti studentus. Užtikrina, kad įvestis būtų tinkama (VP, PV, GM, GD).

Gražina

Pasirinktas rūšiavimo būdas kaip string (VP, PV, GM arba GD).

5.11.1.21 pasirinkimas_del_strategijos()

```
int pasirinkimas_del_strategijos ()
```

Leidžia vartotojui pasirinkti strategiją studentų skirstymui į dvi grupes.

Funkcija pateikia klausimą vartotojui apie pasirenkamą strategiją. Galimos strategijos:

- '1' - Pirmoji strategija,
- '2' - Antroji strategija,
- '3' - Trečioji strategija.

Gražina

Pasirinkta strategija kaip sveikasis skaičius (1, 2 arba 3).

5.11.1.22 pasirinkimas_isvedimo()

```
string pasirinkimas_isvedimo ()
```

Leidžia vartotojui pasirinkti, kur nori matyti rezultatą: terminale ar faile.

Funkcija pateikia klausimą vartotojui, kur norėtų jis matyti rezultatą. Vartotojas gali pasirinkti:

- 'T' - terminale,
- 'F' - faile.

Gražina

Pasirinktas rezultato pateikimo būdas kaip string ('T' arba 'F').

5.11.1.23 Rez_antraste()

```
void Rez_antraste (
    string pasirinkimas,
    ostream & out,
    string isvedimo_pasirinkimas)
```

Funkcija, skirta atspausdinti antraštei pagal vartotojo įvertinimo pasirinkimą.

Parametrai

<i>pasirinkimas</i>	Pasirinkimas nurodantis pagal ką buvo skaičiuotas galutinis rezultatas (vidurkis ar mediana).
<i>out</i>	Srautas, į kurį spausdinama (pvz., terminalas ar failas).
<i>isvedimo_pasirinkimas</i>	Rezultatų išvedimo pasirinkimas (T - terminalas, F - failas).

5.11.1.24 SpausdinimasRez()

```
template<typename Container >
void SpausdinimasRez (
    Container & stud,
    string isvedimo_pasirinkimas,
    string rez_pasirinkimas,
    string rusiavimo_p,
    string ivedimo_skaitymo_p)
```

Funkcija skirta rezultatams atspausdinti į terminalą (vartotojui pasirinkus 'T') arba įrašyti į failą (vartotojui pasirinkus 'F').

Template Parameters

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

Parametrai

<i>isvedimo_pasirinkimas</i>	Rezultatų išvedimo pasirinkimas (T - terminalas, F - failas).
<i>rez_pasirinkimas</i>	Pasirinkimas pagal ką skaičiuotas galutinis rezultatas (vidurkis ar mediana).
<i>rusiavimo_p</i>	Pasirinkimas nurodo, pagal ką studentų konteineris bus rūšiuojamas (pvz.: pagal vardą ir pavardę, pavardę ir vardą, galutinį įvertinimą mažėjančia tvarka, pagal galutinį įvertinimą didėjančia tvarka).
<i>ivedimo_skaitymo_p</i>	Pasirinkimas, kuris tikrinamas ar lygus T - Testavimui, tuo atveju bendras visų studentų konteineris yra atspausdinamas į failą Rez.txt.

Pastaba

Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.11.1.25 SpausdinimasRez< list< Studentas > >()

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

5.11.1.26 SpausdinimasRez< vector< Studentas > >()

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

5.11.1.27 Stud_failu_generavimas()

```
void Stud_failu_generavimas (
    int kiekis)
```

Funkcija, skirta sugeneruoti studentų duomenų failą su nurodytu įrašų kiekiu.

Parametrai

<i>kiekis</i>	Failo įrašų kiekis.
---------------	---------------------

5.11.1.28 Studentu_rusiavimas()

```
template<typename Container >
void Studentu_rusiavimas (
    Container & stud,
    string pasirinkimas)
```

Funkcija skirta studentų rūšiavimui.

Template Parameters

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

Parametrai

<i>pasirinkimas</i>	Pasirinkimas nurodo, pagal ką studentų konteineris bus rūšiuojamas (pvz.: pagal vardą ir pavardę, pavardę ir vardą, galutinį įvertinimą mažėjančia tvarka, pagal galutinį įvertinimą didėjančia tvarka).
---------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Pastaba

Funkcija veikia tiek su `vector<Studentas>`, tiek su `list<Studentas>`.

5.12 src/Studentai.cpp Failo Nuoroda

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

Funkcijos

- `int main ()`

5.12.1 Funkcijos Dokumentacija

5.12.1.1 main()

```
int main ()
```

5.13 src/Studentas.cpp Failo Nuoroda

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

Funkcijos

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

Naudojamas perdengtas operatorius `istream& operator>>(std::istream& is, Studentas& s)`, kuris leidžia įvesti duomenis dviem būdais:

1. Rankinis įvedimas iš terminalo, naudojant `std::cin`. Vartotojas turi pateikti studentų vardus bei pavardes, įvertinimai gali būti įvesti vartotojo rankiniu būdu arba sugeneruojami atsitiktinai pagal vartotojo pasirinkimą.
2. Duomenų nuskaitymas iš teksto srauto (pvz.: `std::stringstream`). Ši funkcija leidžia įvesti studento vardą, pavardę, namų darbų balus ir egzamino balą.

Parametrai

<i>is</i>	Įvesties srautas (<code>std::cin</code> arba failo srautas).
<i>s</i>	Studento klasės objektas, į kurį įrašomi ir saugomi duomenys: vardas, pavardė, namų darbai, egzamino įvertinimas.

Gražina

Modifikuotas įvesties srautas.

Rodyklė

- ~Studentas
 - Studentas, [10](#)
- ~Zmogus
 - Zmogus, [16](#)
- Duom_is_failo
 - Stud.cpp, [35](#)
 - Stud.h, [21](#)
- Duom_is_failo< list< Studentas > >
 - Stud.cpp, [35](#)
- Duom_is_failo< vector< Studentas > >
 - Stud.cpp, [35](#)
- Duom_ivedimas
 - Stud.cpp, [35](#)
 - Stud.h, [21](#)
- Duom_ivedimas< list< Studentas > >
 - Stud.cpp, [36](#)
- Duom_ivedimas< vector< Studentas > >
 - Stud.cpp, [36](#)
- Duom_tvarkymas
 - Stud.cpp, [36](#)
 - Stud.h, [22](#)
- Duom_tvarkymas< list< Studentas > >
 - Stud.cpp, [37](#)
- Duom_tvarkymas< vector< Studentas > >
 - Stud.cpp, [37](#)
- elapsed
 - Timer, [14](#)
- FailasPgalkategorija
 - Stud.cpp, [37](#)
 - Stud.h, [23](#)
- getEgz
 - Studentas, [10](#)
- getGalutinis
 - Studentas, [10](#)
- getNd
 - Studentas, [10](#)
- getPavarde
 - Studentas, [10](#)
 - Zmogus, [16](#)
- getVardas
 - Studentas, [11](#)
 - Zmogus, [16](#)
- include/Mylib.h, [19](#)
- include/Stud.h, [20](#), [30](#)
- include/Studentas.h, [31](#)
- include/timer.h, [32](#)
- include/Zmogus.h, [32](#), [33](#)
- Info_ivedimas_ranka
 - Stud.cpp, [38](#)
 - Stud.h, [23](#)
- Info_ivedimas_ranka< list< Studentas > >
 - Stud.cpp, [38](#)
- Info_ivedimas_ranka< vector< Studentas > >
 - Stud.cpp, [39](#)
- Ivertinimas_med
 - Studentas, [11](#)
- Ivertinimas_vid
 - Studentas, [11](#)
- Kategorijos_Priskirimas1
 - Stud.cpp, [39](#)
 - Stud.h, [24](#)
- Kategorijos_Priskirimas2
 - Stud.cpp, [39](#)
 - Stud.h, [24](#)
- Kategorijos_Priskirimas3
 - Stud.cpp, [40](#)
 - Stud.h, [25](#)
- main
 - Studentai.cpp, [44](#)
- operator<<
 - Studentas, [13](#)
 - Studentas.cpp, [45](#)
- operator>>
 - Studentas, [13](#)
 - Studentas.cpp, [45](#)
- operator=
 - Studentas, [11](#)
- pasirinkimas_del_duom_strukturos
 - Stud.cpp, [40](#)
 - Stud.h, [25](#)
- pasirinkimas_del_galutinio
 - Stud.cpp, [40](#)
 - Stud.h, [26](#)
- pasirinkimas_del_programos_vykdyimo
 - Stud.cpp, [41](#)
 - Stud.h, [26](#)
- pasirinkimas_del_rusiavimo
 - Stud.cpp, [41](#)
 - Stud.h, [26](#)
- pasirinkimas_del_strategijos
 - Stud.cpp, [41](#)

Stud.h, 26
 pasirinkimas_isvedimo
 Stud.cpp, 41
 Stud.h, 27
 pavarde_
 Zmogus, 17

 reset
 Timer, 14
 Rez_antraste
 Stud.cpp, 42
 Stud.h, 27

 setEgz
 Studentas, 12
 setGalutinis
 Studentas, 12
 setNd
 Studentas, 12
 setPavarde
 Studentas, 12
 setVardas
 Studentas, 13
 SpausdinimasRez
 Stud.cpp, 42
 Stud.h, 27
 SpausdinimasRez< list< Studentas > >
 Stud.cpp, 43
 SpausdinimasRez< vector< Studentas > >
 Stud.cpp, 43
 src/Stud.cpp, 33
 src/Studentai.cpp, 44
 src/Studentas.cpp, 44
 Stud.cpp
 Duum_is_failo, 35
 Duum_is_failo< list< Studentas > >, 35
 Duum_is_failo< vector< Studentas > >, 35
 Duum_ivedimas, 35
 Duum_ivedimas< list< Studentas > >, 36
 Duum_ivedimas< vector< Studentas > >, 36
 Duum_tvarkymas, 36
 Duum_tvarkymas< list< Studentas > >, 37
 Duum_tvarkymas< vector< Studentas > >, 37
 FailasPgalKategorija, 37
 Info_ivedimas_ranka, 38
 Info_ivedimas_ranka< list< Studentas > >, 38
 Info_ivedimas_ranka< vector< Studentas > >, 39
 Kategorijos_Priskirimas1, 39
 Kategorijos_Priskirimas2, 39
 Kategorijos_Priskirimas3, 40
 pasirinkimas_del_duom_strukturos, 40
 pasirinkimas_del_galutinio, 40
 pasirinkimas_del_programos_vykdyimo, 41
 pasirinkimas_del_rusiavimo, 41
 pasirinkimas_del_strategijos, 41
 pasirinkimas_isvedimo, 41
 Rez_antraste, 42
 SpausdinimasRez, 42
 SpausdinimasRez< list< Studentas > >, 43
 SpausdinimasRez< vector< Studentas > >, 43
 Stud_failu_generavimas, 43
 Studentu_rusiavimas, 43
 Stud.h
 Duum_is_failo, 21
 Duum_ivedimas, 21
 Duum_tvarkymas, 22
 FailasPgalKategorija, 23
 Info_ivedimas_ranka, 23
 Kategorijos_Priskirimas1, 24
 Kategorijos_Priskirimas2, 24
 Kategorijos_Priskirimas3, 25
 pasirinkimas_del_duom_strukturos, 25
 pasirinkimas_del_galutinio, 26
 pasirinkimas_del_programos_vykdyimo, 26
 pasirinkimas_del_rusiavimo, 26
 pasirinkimas_del_strategijos, 26
 pasirinkimas_isvedimo, 27
 Rez_antraste, 27
 SpausdinimasRez, 27
 Stud_failu_generavimas, 29
 Studentu_rusiavimas, 29
 Stud_failu_generavimas
 Stud.cpp, 43
 Stud.h, 29
 Studentai.cpp
 main, 44
 Studentas, 7
 ~Studentas, 10
 getEgz, 10
 getGalutinis, 10
 getNd, 10
 getPavarde, 10
 getVardas, 11
 Ivertinimas_med, 11
 Ivertinimas_vid, 11
 operator<<, 13
 operator>>, 13
 operator=, 11
 setEgz, 12
 setGalutinis, 12
 setNd, 12
 setPavarde, 12
 setVardas, 13
 Studentas, 9
 Studentas.cpp
 operator<<, 45
 operator>>, 45
 Studentu_rusiavimas
 Stud.cpp, 43
 Stud.h, 29

 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](#)