

ObjektinisProg2

Sugeneruota Doxygen 1.13.2



<b>1 objektinisProg2</b>	<b>1</b>
1.1 Programų spartos palyginimas <code>struct</code> vs. <code>class</code>	1
1.1.1 Testavimo metodika	1
1.1.2 Rezultatai	1
1.1.2.1 Struktūros ( <code>struct</code> ) rezultatai	1
1.1.2.2 Klasės ( <code>class</code> ) rezultatai	1
1.1.3 Išvados	1
1.2 Programos spartos analizė su O1, O2, O3 flag'ais	2
1.2.1 Testavimo metodika	2
1.2.2 Rezultatai	2
1.2.2.1 .exe failų dydžiai	2
1.2.3 Analizė	2
1.2.4 Išvados	3
1.3 objektinisProg	3
1.4 RULE OF 5 IR IO operatoriai	3
1.4.1 Uždavinio paaiškinimas	3
1.4.2 Įvesties galimybės	3
1.4.3 Išvesties galimybės	4
1.4.4 Testavimo informacija	4
1.4.5 Ekrano vaizdai	4
1.4.5.1 Programos paleidimas	4
1.4.5.2 Rule of Five testavimas	4
1.4.5.3 Įvesties ir išvesties operatorių testavimas	4
1.4.5.4 Išvedimas į ekraną ir failą	4
1.4.5.5 Destruktoriaus kvietimai	4
1.4.6 Failai	4
1.5 BAZINĖ IR IŠVESTINĖ klasės	4
1.5.1 Versijos santrauka	4
1.5.2 Klasės struktūra	5
1.5.2.1 <code>Zmogus</code> (abstrakti klasė)	5
1.5.2.2 <code>Studentas : public Zmogus</code>	5
1.5.3 Testavimo santrauka	5
1.5.4 Ekrano vaizdai	6
1.5.4.1 Programos paleidimas	6
1.5.4.2 Rule of Five testavimas	6
1.5.4.3 Įvesties / išvesties operatoriai	6
1.5.4.4 Išvedimas į ekraną ir failą	6
1.5.4.5 Destruktoriaus kvietimai	6
<b>2 Hierarchijos Indeksas</b>	<b>7</b>
2.1 Klasių hierarchija	7
<b>3 Klasės Indeksas</b>	<b>9</b>

3.1 Klasės	9
<b>4 Failo Indeksas</b>	<b>11</b>
4.1 Failai	11
<b>5 Klasės Dokumentacija</b>	<b>13</b>
5.1 LaikoMatavimas Klasė	13
5.1.1 Konstruktoriaus ir Destruktoriaus Dokumentacija	13
5.1.1.1 LaikoMatavimas()	13
5.1.2 Metodų Dokumentacija	13
5.1.2.1 baigti()	13
5.1.2.2 gautiLaikoSkirtuma()	13
5.1.2.3 pradeti()	13
5.2 Studentas Klasė	13
5.2.1 Konstruktoriaus ir Destruktoriaus Dokumentacija	15
5.2.1.1 Studentas() [1/4]	15
5.2.1.2 Studentas() [2/4]	15
5.2.1.3 Studentas() [3/4]	15
5.2.1.4 Studentas() [4/4]	15
5.2.1.5 ~Studentas()	15
5.2.2 Metodų Dokumentacija	15
5.2.2.1 egzaminas()	15
5.2.2.2 galBalas()	15
5.2.2.3 galutinisBalsas()	15
5.2.2.4 nd()	15
5.2.2.5 nuskaitymasFile()	15
5.2.2.6 operator=() [1/2]	15
5.2.2.7 operator=() [2/2]	15
5.2.2.8 pavarde()	16
5.2.2.9 readStudent()	16
5.2.2.10 rusiuotiStud()	16
5.2.2.11 setEgzaminas()	16
5.2.2.12 setNamudarbai()	16
5.2.2.13 setPavarde()	16
5.2.2.14 setVardas()	16
5.2.2.15 spausdintiRez()	16
5.2.2.16 vardas()	16
5.2.3 Draugiškų Ir Susijusių Funkcijų Dokumentacija	16
5.2.3.1 operator<<	16
5.2.3.2 operator>>	17
5.2.4 Atributų Dokumentacija	17
5.2.4.1 destruktoriausSk	17
5.3 studentas< T > Struktūra Šablonas	17

5.3.1 Tipo Aprašymo Dokumentacija	17
5.3.1.1 nd_type	17
5.3.2 Atributų Dokumentacija	17
5.3.2.1 egz	17
5.3.2.2 Gal	17
5.3.2.3 nd	17
5.3.2.4 Pav	17
5.3.2.5 Vard	18
5.4 Zmogus Klasė	18
5.4.1 Konstruktoriaus ir Destruktoriaus Dokumentacija	18
5.4.1.1 Zmogus() [1/2]	18
5.4.1.2 Zmogus() [2/2]	18
5.4.1.3 ~Zmogus()	18
5.4.2 Metodų Dokumentacija	18
5.4.2.1 pavarde()	18
5.4.2.2 setPavarde()	19
5.4.2.3 setVardas()	19
5.4.2.4 vardas()	19
5.4.3 Atributų Dokumentacija	19
5.4.3.1 pavarde_	19
5.4.3.2 vardas_	19
<b>6 Failo Dokumentacija</b>	<b>21</b>
6.1 include/LaikoMatavimas.h Failo Nuoroda	21
6.2 LaikoMatavimas.h	21
6.3 include/strategija3.h Failo Nuoroda	21
6.3.1 Funkcijos Dokumentacija	22
6.3.1.1 isskirtiVargsusIrKietiakusOpt()	22
6.3.1.2 spausdintiKietiakusIrVargsius()	22
6.3.1.3 testuotiSkaidymoStrategija3()	22
6.4 strategija3.h	22
6.5 include/studentas.h Failo Nuoroda	22
6.5.1 Funkcijos Dokumentacija	23
6.5.1.1 generuotiGalmed()	23
6.5.1.2 generuotiGalvid()	23
6.5.2 Kintamojo Dokumentacija	23
6.5.2.1 buffer	23
6.6 studentas.h	23
6.7 include/testRuleOf5.h Failo Nuoroda	25
6.7.1 Funkcijos Dokumentacija	25
6.7.1.1 testuotiRuleOfFive()	25
6.8 testRuleOf5.h	25

6.9 include/versijos1Funkc.h Failo Nuoroda . . . . .	25
6.9.1 Funkcijos Dokumentacija . . . . .	26
6.9.1.1 generuotiGalmed() . . . . .	26
6.9.1.2 generuotiGalvid() . . . . .	26
6.9.1.3 generuotiVardPav() . . . . .	26
6.9.1.4 nuskaitymasFile() . . . . .	26
6.9.1.5 pasirinktasGal() . . . . .	26
6.9.1.6 randomPaz() . . . . .	26
6.9.1.7 rusiuotiStud() . . . . .	26
6.9.1.8 spausdintiRez() . . . . .	27
6.9.1.9 spausdintiRez2() . . . . .	27
6.9.2 Kintamojo Dokumentacija . . . . .	27
6.9.2.1 buffer . . . . .	27
6.10 versijos1Funkc.h . . . . .	27
6.11 include/zmogus.h Failo Nuoroda . . . . .	28
6.12 zmogus.h . . . . .	28
6.13 objektinisProg/README.md Failo Nuoroda . . . . .	29
6.14 README.md Failo Nuoroda . . . . .	29
6.15 Src/LaikoMatavimas.cpp Failo Nuoroda . . . . .	29
6.16 Src/strategija3.cpp Failo Nuoroda . . . . .	29
6.16.1 Funkcijos Dokumentacija . . . . .	29
6.16.1.1 isskirtiVargsusIrKietiakusOpt() . . . . .	29
6.16.1.2 spausdintiKietiakusIrVargsius() . . . . .	29
6.16.1.3 testuotiSkaidymoStrategija3() . . . . .	29
6.17 Src/studentas.cpp Failo Nuoroda . . . . .	29
6.17.1 Funkcijos Dokumentacija . . . . .	30
6.17.1.1 generuotiGalmed() . . . . .	30
6.17.1.2 generuotiGalvid() . . . . .	30
6.17.1.3 operator<<() . . . . .	30
6.17.1.4 operator>>() . . . . .	30
6.18 Src/testRuleOf5.cpp Failo Nuoroda . . . . .	30
6.18.1 Funkcijos Dokumentacija . . . . .	30
6.18.1.1 testuotiRuleOfFive() . . . . .	30
6.19 Src/v1.2.cpp Failo Nuoroda . . . . .	30
6.19.1 Funkcijos Dokumentacija . . . . .	31
6.19.1.1 main() . . . . .	31
6.19.1.2 meniu() . . . . .	31
6.19.1.3 pasirinktiFaila() . . . . .	31
6.19.1.4 vykdytiPrograma() . . . . .	31
6.20 Src/versijos1Funkc.cpp Failo Nuoroda . . . . .	31
6.20.1 Funkcijos Dokumentacija . . . . .	32
6.20.1.1 generuotiGalmed() . . . . .	32

6.20.1.2 generuotiGalvid()	32
6.20.1.3 generuotiVardPav()	32
6.20.1.4 nuskaitymasFile()	32
6.20.1.5 nuskaitymasFile< std::deque< studentas< std::deque< float > > > >()	32
6.20.1.6 nuskaitymasFile< std::list< studentas< std::list< float > > > >()	33
6.20.1.7 nuskaitymasFile< std::vector< studentas< std::vector< float > > > >()	33
6.20.1.8 pasirinktasGal()	33
6.20.1.9 pasirinktasGal< std::deque< float > >()	33
6.20.1.10 pasirinktasGal< std::list< float > >()	33
6.20.1.11 pasirinktasGal< std::vector< float > >()	33
6.20.1.12 randomPaz()	33
6.20.1.13 randomPaz< std::deque< float > >()	33
6.20.1.14 randomPaz< std::list< float > >()	33
6.20.1.15 randomPaz< std::vector< float > >()	34
6.20.1.16 rusiuotiStud()	34
6.20.1.17 rusiuotiStud< std::deque< studentas< std::deque< float > > > >()	34
6.20.1.18 rusiuotiStud< std::list< studentas< std::list< float > > > >()	34
6.20.1.19 rusiuotiStud< std::vector< studentas< std::vector< float > > > >()	34
6.20.1.20 spausdintiRez()	34
6.20.1.21 spausdintiRez2()	34
6.20.1.22 spausdintiRez< std::deque< studentas< std::deque< float > > > >()	34
6.20.1.23 spausdintiRez< std::list< studentas< std::list< float > > > >()	35
6.20.1.24 spausdintiRez< std::vector< studentas< std::vector< float > > > >()	35

## Rodyklė

37





# skyrius 1

## objektinisProg2

### 1.1 Programų spartos palyginimas `struct` vs. `class`

Šiame dokumente pateikiamas programų, naudojančių `struct` ir `class` tipo studentus, spartos palyginimas.

#### 1.1.1 Testavimo metodika

- Konteineris: `std::vector`
- Failų dydžiai: 100000 ir 1000000 studentų
- Dalijimo strategija: Greičiausia (3 strategija)

#### 1.1.2 Rezultatai

##### 1.1.2.1 Struktūros (`struct`) rezultatai

Studentų skaičius	Failo nuskaitymas (s)	Studentų rūšiavimas (s)	Skirstymas į grupes (s)
100000	0.20392	0.0007994	0.01625
1000000	1.86321	0.00835099	0.19125

##### 1.1.2.2 Klasės (`class`) rezultatai

Studentų skaičius	Failo nuskaitymas (s)	Studentų rūšiavimas (s)	Skirstymas į grupes (s)
100000	0.196728	0.000793	0.013829
1000000	1.824976	0.009382	0.214729

#### 1.1.3 Išvados

- Abiejų implementacijų veikimo laikas yra panašus.
- Nėra didelio skirtumo tarp `struct` ir `class` naudojimo spartos.
- Nedideli skirtumai gali atsirasti dėl įvairių faktorių.

## 1.2 Programos spartos analizė su O1, O2, O3 flag'ais

Šiame dokumente pateikiama programos spartos analizė, lyginant "Struktūros" ir "Klasės" variantus su skirtingais optimizavimo flagais.

### 1.2.1 Testavimo metodika

1. Programa buvo paleista su skirtingais studentų skaičiais: 1000, 10000, 100000, 1000000 ir 10000000.
2. Programa buvo kompiliuota su optimizavimo flagais: O1, O2, O3.
3. Buvo matuojamos šios operacijos:
  - Failo nuskaitymas
  - Studentų rūšiavimas
  - Studentų skaidymas į grupes
4. Testai buvo pakartoti kelis kartus ir apskaičiuoti vidurkiai.

### 1.2.2 Rezultatai

#### 1.2.2.1 .exe failų dydžiai

Tipas	Optimizavimo flagas	.exe failo dydis (KB)
Struktūra	O1	438
Struktūra	O2	267
Struktūra	O3	282
Klasė	O1	327
Klasė	O2	209
Klasė	O3	224

- .exe failų dydžiai mažėja su didesniais optimizavimo flagais.

Tipas	Optimizavimo flagas	Failo dydis	Failo nuskaitymas (s)	Studentų rūšiavimas (s)	Skirstymas į grupes (s)
Struktūra	O1	100000	0.3506804	1.022409	0.0343105
Struktūra	O1	1000000	3.199772	10.15857	0.3709248
Struktūra	O2	100000	0.2065624	0.8267488	0.0143346
Struktūra	O2	1000000	1.837956	8.00394	0.2062462
Struktūra	O3	100000	0.2042266	0.7903494	0.0147426
Struktūra	O3	1000000	1.81989	8.5064	0.2086486
Klasė	O1	100000	0.337025	1.518416	0.0369319
Klasė	O1	1000000	3.049766	15.45106	0.4165822
Klasė	O2	100000	0.217939	0.811108	0.01640278
Klasė	O2	1000000	2.086458	9.742368	0.2256188
Klasė	O3	100000	0.198677	1.077024	0.01438404
Klasė	O3	1000000	1.84187	8.099636	0.202793

### 1.2.3 Analizė

- Optimizavimo flagų įtaka:
  - Matome, kad optimizavimo flagai turi didelę įtaką programos veikimo laikui. Su didesniu optimizavimo flagu (O3) programos veikimo laikas sumažėja.
  - Optimizavimo flagai ypač veikia failo nuskaitymo ir skaidymo į grupes operacijas.

- **Failo nuskaitymas:**
  - Failo nuskaitymo laikas didėja tiesiškai didėjant studentų skaičiui.
  - Optimizavimo flagai akivaizdžiai turi įtakos failo nuskaitymo laikui.
- **Studentų rūšiavimas:**
  - Rūšiavimo laikas yra labai mažas abiem variantams.
  - Optimizavimo flagai turi minimalią įtaką rūšiavimo laikui.
- **Studentų skaidymas į grupes:**
  - Skaidymo laikas didėja didėjant studentų skaičiui.
  - Optimizavimo flagai turi didelę įtaką skaidymo laikui.
- **Struktūra vs. Klasė:**
  - Abiejų variantų veikimo laikas yra panašus, tačiau yra nedidelių skirtumų, priklausomai nuo optimizavimo flagų.
  - Klasės variantas su O2 ir O3 optimizavimo flagais rodo šiek tiek didesnį nuskaitymo laiką didesniame faile.

#### 1.2.4 Išvados

- Optimizavimo flagai (O1, O2, O3) reikšmingai pagerina programos veikimo laiką, ypač didesniems duomenų rinkiniams.
- Failo nuskaitymo ir studentų skaidymo į grupes operacijos labiausiai pasinaudoja optimizavimo flagais.
- Studentų rūšiavimo operacija yra palyginti greita ir mažai priklauso nuo optimizavimo flagų.
- Rekomenduojama naudoti didesnius optimizavimo flagus (O2 ar O3) didesniems duomenų rinkiniams, kad programa veiktų greičiau.

### 1.3 objektinisProg

bfee5a45c0feec4d2f885eedda9dd39cab515343

## 1.4 RULE OF 5 IR IO operatoriai

### 1.4.1 Uždavinio paaiškinimas

Šioje užduotyje reikėjo:

- **Studentas** klasėje pilnai realizuoti „Rule of Five“ metodus:
  - Konstruktorius (default ir su parametrais)
  - Kopijavimo konstruktorius
  - Kopijavimo priskyrimo operatorius
  - Perkėlimo konstruktorius (move)
  - Perkėlimo priskyrimo operatorius
  - Destruktorius
- Perdengti >> ir << operatorius:
  - Kad palaikytų įvestį iš vartotojo (cin), iš stringo (testavimui), iš failo
  - Kad palaikytų išvestį į ekraną ir į failą
- Sukurti testą (`testuotiRuleOfFive()`), kuris patikrina visus metodus.

### 1.4.2 Įvesties galimybės

Tipas	Aprašymas
Rankinė	Vartotojas įveda vardą, pavardę, pažymius ir egzaminą naudodamas <code>cin</code>
Automatinė	Įvestis testavimo metu per <code>istream</code>
Iš failo	Įvestis iš failo per <code>ifstream</code> , formatas: Vardas Pavarde nd1 nd2 ... egz

#### Pavyzdys faile:

`Jonas Jonaitis 10 9 8 7 6 8`

### 1.4.3 Išvesties galimybės

Tipas	Aprašymas
Į ekraną	Naudojamas <code>cout &lt;&lt; studentas</code>
Į failą	Studentai išvedami į <code>rezultataiT.txt</code> failą

### 1.4.4 Testavimo informacija

Funkcija `testuotiRuleOfFive()` tikrina viską:

- Originalaus objekto kūrimą
- Kopijavimą ir perkėlimą (visos 5 taisyklės)
- Stream operatorius (`<<` ir `>>`) su:
  - `cin`
  - `istream`
  - `ifstream`
- Failo skaitymą su `nuskaitymasFile`
- Failo išrašymą su `ofstream`
- Destruktoriaus iškvietimus

### 1.4.5 Ekranu vaizdai

#### 1.4.5.1 Programos paleidimas

#### 1.4.5.2 Rule of Five testavimas

#### 1.4.5.3 Įvesties ir išvesties operatorių testavimas

#### 1.4.5.4 Išvedimas į ekraną ir failą

#### 1.4.5.5 Destruktoriaus kvietimai

### 1.4.6 Failai

- `studentai.txt`, `failinis.txt` – įvesties failai testams
- `rezultataiT.txt` – išvedimo rezultatai

## 1.5 BAZINĖ IR IŠVESTINĖ klasės

### 1.5.1 Versijos santrauka

- Sukurta abstrakti bazinė klasė `Zmogus`
- `Studentas` dabar paveldi iš `Zmogus`
- Įvykdyta "Rule of Five" taisyklė ir visi metodai perkelti
- Perdengti įvesties / išvesties operatoriai
- Programa vis dar palaiko ankstesnę v1.2 logiką

## 1.5.2 Klasės struktūra

### 1.5.2.1 Zmogus (abstrakti klasė)

Atributai	Tipas
vardas↔ —	string
pavarde↔ —	string

#### Funkcijos (grynai virtualios):

- `string vardas() const = 0;`
- `string pavarde() const = 0;`
- `void setVardas(const string&) = 0;`
- `void setPavarde(const string&) = 0;`

#### Testavimas:

```
// Zmogus z; // Nesikompiluoja, nes Zmogus yra abstrakti klasė
```

### 1.5.2.2 Studentas : `public Zmogus`

Paveldi `vardas_`, `pavarde_`, perrašo virtualias funkcijas ir turi:

Atributai	Tipas
<code>egzaminas_</code>	<code>int</code>
<code>nd_</code>	<code>vector&lt;float&gt;</code>
<code>galutinis_↔</code> <code>balas_</code>	<code>float</code>

#### Implementuoti metodai:

- 5 Rule of Five: konstruktoriai, operatoriai, destruktorius
- `istream& operator>>(istream&, Studentas&)`
- `ostream& operator<<(ostream&, const Studentas&)`
- `float galBalas(...), readStudent(...), nuskaitymasFile(...)` ir kt.

## 1.5.3 Testavimo santrauka

Testas	Veikia
Konstruktoriai (default, copy)	+
Priskyrimo operatoriai	+
Perkelimo metodai	+
Destruktorius	+
<code>operator&lt;&lt;, operator&gt;&gt;</code>	+
Nuskaitymas iš failo	+
Išvedimas į failą / ekraną	+
Paveldėjimo loginis testas	+

---

## **1.5.4 Ekranų vaizdai**

**1.5.4.1 Programos paleidimas**

**1.5.4.2 Rule of Five testavimas**

**1.5.4.3 Įvesties / išvesties operatoriai**

**1.5.4.4 Išvedimas į ekraną ir failą**

**1.5.4.5 Destruktoriaus kvietimai**

---

## skyrius 2

# Hierarchijos Indeksas

### 2.1 Klasių hierarchija

Šis paveldėjimo sąrašas yra beveik surikiuotas abėcėlės tvarka:

LaikoMatavimas . . . . .	13
studentas< T > . . . . .	17
Zmogus . . . . .	18
Studentas . . . . .	13





## skyrius 3

# Klasės Indeksas

### 3.1 Klasės

Klasės, struktūros, sąjungos ir sąsajos su trumpais aprašymais:

<a href="#">LaikoMatavimas</a>	13
<a href="#">Studentas</a>	13
<a href="#">studentas&lt; T &gt;</a>	17
<a href="#">Zmogus</a>	18



## skyrius 4

# Failo Indeksas

### 4.1 Failai

Visų failų sąrašas su trumpais aprašymais:

include/ <a href="#">LaikoMatavimas.h</a> . . . . .	21
include/ <a href="#">strategija3.h</a> . . . . .	21
include/ <a href="#">studentas.h</a> . . . . .	22
include/ <a href="#">testRuleOf5.h</a> . . . . .	25
include/ <a href="#">versijos1Funkc.h</a> . . . . .	25
include/ <a href="#">zmogus.h</a> . . . . .	28
Src/ <a href="#">LaikoMatavimas.cpp</a> . . . . .	29
Src/ <a href="#">strategija3.cpp</a> . . . . .	29
Src/ <a href="#">studentas.cpp</a> . . . . .	29
Src/ <a href="#">testRuleOf5.cpp</a> . . . . .	30
Src/ <a href="#">v1.2.cpp</a> . . . . .	30
Src/ <a href="#">versijos1Funkc.cpp</a> . . . . .	31



## skyrius 5

# Klasės Dokumentacija

### 5.1 LaikoMatavimas Klasė

```
#include <LaikoMatavimas.h>
```

#### Vieši Metodai

- [LaikoMatavimas](#) (const std::string &pavadinimas)
- void [pradeti](#) ()
- void [baigti](#) ()
- double [gautiLaikoSkirtuma](#) ()

#### 5.1.1 Konstruktoriaus ir Destruktoriaus Dokumentacija

##### 5.1.1.1 LaikoMatavimas()

```
LaikoMatavimas::LaikoMatavimas (  
    const std::string & pavadinimas)
```

#### 5.1.2 Metodų Dokumentacija

##### 5.1.2.1 baigti()

```
void LaikoMatavimas::baigti ()
```

##### 5.1.2.2 gautiLaikoSkirtuma()

```
double LaikoMatavimas::gautiLaikoSkirtuma ()
```

##### 5.1.2.3 pradeti()

```
void LaikoMatavimas::pradeti ()
```

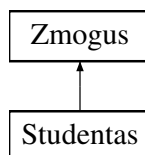
Dokumentacija šiai klasei sugeneruota iš šių failų:

- include/[LaikoMatavimas.h](#)
- Src/[LaikoMatavimas.cpp](#)

### 5.2 Studentas Klasė

```
#include <studentas.h>
```

Paveldimumo diagrama Studentas:



### Vieši Metodai

- `Studentas ()`
- `Studentas (istream &is)`
- `Studentas (const Studentas &other)`
- `Studentas & operator= (const Studentas &other)`
- `Studentas (Studentas &&other) noexcept`
- `Studentas & operator= (Studentas &&other) noexcept`
- `~Studentas ()`
- `string vardas () const` override
- `string pavarde () const` override
- `int egzaminas () const`
- `const vector< float > & nd () const`
- `float galutinisBalsas () const`
- `void setVardas (const string &v)` override
- `void setPavarde (const string &p)` override
- `void setEgzaminas (int e)`
- `void setNamudarbai (const vector< float > &nd)`
- `float galBalas (float(*balasFunkcija)(const vector< float > &, int))`
- `istream & readStudent (istream &)`

### Vieši Metodai inherited from Zmogus

- `Zmogus ()=default`
- `Zmogus (const string &v, const string &p)`
- `virtual ~Zmogus ()=default`

### Statiniai Vieši Metodai

- `static void rusiuotiStud (vector< Studentas > &grupe, char rusiavimoPas)`
- `static void nuskaitymasFile (vector< Studentas > &grupe, const string &filename)`
- `static void spausdintiRez (vector< Studentas > &grupe, bool iFaila, char pasirinkimas, const std::string &failoPavadinimas)`

### Statiniai Vieši Atributai

- `static int destruktoriuSk = 0`

### Draugai

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

### Additional Inherited Members

### Apsaugoti Atributai inherited from Zmogus

- `string vardas_`
- `string pavarde_`

## 5.2.1 Konstruktoriaus ir Destruktoriaus Dokumentacija

### 5.2.1.1 Studentas() [1/4]

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

### 5.2.1.2 Studentas() [2/4]

```
Studentas::Studentas (
    istream & is)
```

### 5.2.1.3 Studentas() [3/4]

```
Studentas::Studentas (
    const Studentas & other)
```

### 5.2.1.4 Studentas() [4/4]

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

### 5.2.1.5 ~Studentas()

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

## 5.2.2 Metodų Dokumentacija

### 5.2.2.1 egzaminas()

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

### 5.2.2.2 galBalas()

```
float Studentas::galBalas (
    float(* balasFunkcija ) (const vector< float > &, int))
```

### 5.2.2.3 galutinisBalsas()

```
float Studentas::galutinisBalsas () const [inline]
```

### 5.2.2.4 nd()

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

### 5.2.2.5 nuskaitymasFile()

```
void Studentas::nuskaitymasFile (
    vector< Studentas > & grupe,
    const string & filename) [static]
```

### 5.2.2.6 operator=() [1/2]

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

### 5.2.2.7 operator=() [2/2]

```
Studentas & Studentas::operator= (
    Studentas && other) [noexcept]
```

#### 5.2.2.8 pavarde()

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

Realizuoja [Zmogus](#).

#### 5.2.2.9 readStudent()

```
istream & Studentas::readStudent (
    istream & is)
```

#### 5.2.2.10 rusiuotiStud()

```
void Studentas::rusiuotiStud (
    vector< Studentas > & grupe,
    char rusiavimoPas) [static]
```

#### 5.2.2.11 setEgzaminas()

```
void Studentas::setEgzaminas (
    int e)
```

#### 5.2.2.12 setNamudarbai()

```
void Studentas::setNamudarbai (
    const vector< float > & nd)
```

#### 5.2.2.13 setPavarde()

```
void Studentas::setPavarde (
    const string & p) [inline], [override], [virtual]
```

Realizuoja [Zmogus](#).

#### 5.2.2.14 setVardas()

```
void Studentas::setVardas (
    const string & v) [inline], [override], [virtual]
```

Realizuoja [Zmogus](#).

#### 5.2.2.15 spausdintiRez()

```
static void Studentas::spausdintiRez (
    vector< Studentas > & grupe,
    bool iFaila,
    char pasirinkimas,
    const std::string & failoPavadinimas) [static]
```

#### 5.2.2.16 vardas()

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

Realizuoja [Zmogus](#).

### 5.2.3 Draugiškų Ir Susijusių Funkcijų Dokumentacija

#### 5.2.3.1 operator<<

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



### 5.2.3.2 operator>>

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

## 5.2.4 Atributų Dokumentacija

### 5.2.4.1 destruktoriuSk

```
int Studentas::destruktoriuSk = 0 [static]
```

Dokumentacija šiai klasei sugeneruota iš šių failų:

- include/studentas.h
- Src/studentas.cpp

## 5.3 studentas< T > Struktūra Šablonas

```
#include <versijos1Funkc.h>
```

### Vieši Tipai

- using nd\_type = T

### Vieši Atributai

- string Vard
- string Pav
- nd\_type nd
- int egz
- double Gal

### 5.3.1 Tipo Aprašymo Dokumentacija

#### 5.3.1.1 nd\_type

```
template<typename T = vector<float>>
using studentas< T >::nd_type = T
```

### 5.3.2 Atributų Dokumentacija

#### 5.3.2.1 egz

```
template<typename T = vector<float>>
int studentas< T >::egz
```

#### 5.3.2.2 Gal

```
template<typename T = vector<float>>
double studentas< T >::Gal
```

#### 5.3.2.3 nd

```
template<typename T = vector<float>>
nd_type studentas< T >::nd
```

#### 5.3.2.4 Pav

```
template<typename T = vector<float>>
string studentas< T >::Pav
```

### 5.3.2.5 Vard

```
template<typename T = vector<float>>>
string studentas< T >::Vard
```

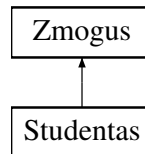
Dokumentacija šiai struktūrai sugeneruota iš šio failo:

- include/[versijos1Funkc.h](#)

## 5.4 Zmogus Klasė

```
#include <zmogus.h>
```

Paveldimumo diagrama Zmogus:



### Vieši Metodai

- [Zmogus](#) ()=default
- [Zmogus](#) (const string &v, const string &p)
- virtual [~Zmogus](#) ()=default
- virtual string [vardas](#) () const =0
- virtual string [pavarde](#) () const =0
- virtual void [setVardas](#) (const string &)=0
- virtual void [setPavarde](#) (const string &)=0

### Apsaugoti Atributai

- string [vardas\\_](#)
- string [pavarde\\_](#)

## 5.4.1 Konstruktorius ir Destruktorius Dokumentacija

### 5.4.1.1 Zmogus() [1/2]

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

### 5.4.1.2 Zmogus() [2/2]

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

### 5.4.1.3 ~Zmogus()

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

## 5.4.2 Metodų Dokumentacija

### 5.4.2.1 pavarde()

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

Realizuota [Studentas](#).

#### 5.4.2.2 setPavarde()

```
virtual void Zmogus::setPavarde (
    const string & ) [pure virtual]
```

Realizuota [Studentas](#).

#### 5.4.2.3 setVardas()

```
virtual void Zmogus::setVardas (
    const string & ) [pure virtual]
```

Realizuota [Studentas](#).

#### 5.4.2.4 vardas()

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

Realizuota [Studentas](#).

### 5.4.3 Atributų Dokumentacija

#### 5.4.3.1 pavarde\_

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

#### 5.4.3.2 vardas\_

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

Dokumentacija šiai klasei sugeneruota iš šio failo:

- include/[zmogus.h](#)



## skyrius 6

# Failo Dokumentacija

### 6.1 include/LaikoMatavimas.h Failo Nuoroda

```
#include <chrono>
#include <iostream>
```

#### Klasės

- class [LaikoMatavimas](#)

### 6.2 LaikoMatavimas.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef LAIKO_MATAVIMAS_H
00002 #define LAIKO_MATAVIMAS_H
00003
00004 #include <chrono>
00005 #include <iostream>
00006
00007 class LaikoMatavimas {
00008 private:
00009     std::chrono::high_resolution_clock::time_point start;
00010     std::chrono::high_resolution_clock::time_point end;
00011     std::string veiksmoPavadinimas;
00012
00013 public:
00014     // Konstruktorius su veiksmo pavadinimu
00015     LaikoMatavimas(const std::string& pavadinimas);
00016
00017     // Pradėti matavimą
00018     void pradeti();
00019
00020     // Baigti matavimą ir parodyti laiką
00021     void baigti();
00022
00023     // Gauti laiko skirtumą (ms)
00024     double gautiLaikoSkirtuma();
00025 };
00026
00027 #endif
```

### 6.3 include/strategija3.h Failo Nuoroda

```
#include <vector>
#include "studentas.h"
#include "LaikoMatavimas.h"
```

## Funkcijos

- void `spausdintiKietiakusIrVargsius` (vector< `Studentas` > &vargsiai, vector< `Studentas` > &kietiakai, const string &vargsiuFailas, const string &kietiakuFailas)
- void `isskirtiVargsusIrKietiakusOpt` (vector< `Studentas` > &grupe, vector< `Studentas` > &vargsiai, vector< `Studentas` > &kietiakai)
- void `testuotiSkaidymoStrategija3` (const string &failoPavadinimas, const string &rezultataiAplankas)

### 6.3.1 Funkcijos Dokumentacija

#### 6.3.1.1 isskirtiVargsusIrKietiakusOpt()

```
void isskirtiVargsusIrKietiakusOpt (
    vector< Studentas > & grupe,
    vector< Studentas > & vargsiai,
    vector< Studentas > & kietiakai)
```

#### 6.3.1.2 spausdintiKietiakusIrVargsius()

```
void spausdintiKietiakusIrVargsius (
    vector< Studentas > & vargsiai,
    vector< Studentas > & kietiakai,
    const string & vargsiuFailas,
    const string & kietiakuFailas)
```

#### 6.3.1.3 testuotiSkaidymoStrategija3()

```
void testuotiSkaidymoStrategija3 (
    const string & failoPavadinimas,
    const string & rezultataiAplankas)
```

## 6.4 strategija3.h

Eiti į šio failo dokumentaciją.

```
00001 #ifndef STRATEGIJA3_H
00002 #define STRATEGIJA3_H
00003
00004 #include <vector>
00005 #include "studentas.h"
00006 #include "LaikoMatavimas.h"
00007
00008 void spausdintiKietiakusIrVargsius(
00009     vector<Studentas>& vargsiai,
00010     vector<Studentas>& kietiakai,
00011     const string& vargsiuFailas,
00012     const string& kietiakuFailas);
00013
00014 //3 strategija - Optimizuotas studentų skirstymas i vargsus ir kietiakus
00015 void isskirtiVargsusIrKietiakusOpt(
00016     vector<Studentas>& grupe,
00017     vector<Studentas>& vargsiai,
00018     vector<Studentas>& kietiakai);
00019
00020 //testavimo funkcija su laiko matavimais
00021 void testuotiSkaidymoStrategija3(const string& failoPavadinimas, const string& rezultataiAplankas);
00022
00023 #endif // STRATEGIJA3_H
```

## 6.5 include/studentas.h Failo Nuoroda

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

```
#include <limits>
#include <algorithm>
#include <cstdlib>
#include <ctime>
#include <sstream>
#include <filesystem>
#include <chrono>
#include <random>
#include <list>
#include <deque>
#include "zmogus.h"
```

## Klasės

- class [Studentas](#)

## Funkcijos

- float [generuotiGalvid](#) (const vector< float > &nd, int egz)
- float [generuotiGalmed](#) (const vector< float > &nd, int egz)

## Kintamieji

- std::stringstream [buffer](#)

## 6.5.1 Funkcijos Dokumentacija

### 6.5.1.1 generuotiGalmed()

```
float generuotiGalmed (
    const vector< float > & nd,
    int egz)
```

### 6.5.1.2 generuotiGalvid()

```
float generuotiGalvid (
    const vector< float > & nd,
    int egz)
```

## 6.5.2 Kintamojo Dokumentacija

### 6.5.2.1 buffer

```
std::stringstream buffer [extern]
```

## 6.6 studentas.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef STUDENTAS_H
00002 #define STUDENTAS_H
00003
00004 #include <iostream>
00005 #include <vector>
00006 #include <iomanip>
00007 #include <string>
00008 #include <fstream>
00009 #include <limits>
00010 #include <algorithm>
00011 #include <cstdlib>
00012 #include <ctime>
00013 #include <sstream>
00014 #include <filesystem>
```

```

00015 #include <chrono>
00016 #include <random>
00017 #include <list>
00018 #include <deque>
00019
00020 #include "zmogus.h" // paveldėjimas
00021
00022 using std::cout;
00023 using std::cin;
00024 using std::vector;
00025 using std::setw;
00026 using std::endl;
00027 using std::string;
00028 using std::fstream;
00029 using std::fixed;
00030 using std::setprecision;
00031 using std::cerr;
00032 using std::getline;
00033 using std::ifstream;
00034 using std::ofstream;
00035 using std::ostream;
00036 using std::stringstream;
00037 using std::numeric_limits;
00038 using std::streamsize;
00039 using namespace std::chrono;
00040 using std::ostringstream;
00041 using std::istringstream;
00042 using std::move;
00043 extern std::stringstream buffer;
00044 using std::list;
00045 using std::deque;
00046 using std::istream;
00047 using std::ostream;
00048
00049 // Galutinio balo apskaičiavimo funkcijos
00050 float generuotiGalvid(const vector<float>& nd, int egz);
00051 float generuotiGalmed(const vector<float>& nd, int egz);
00052
00053 class Studentas : public Zmogus {
00054 private:
00055     int egzaminas_;
00056     vector<float> nd_;
00057     float galutinis_balas_;
00058
00059 public:
00060     Studentas(); // default konstruktorius
00061     Studentas(istream& is); // konstruktorius su stream
00062
00063     // Rule of Five
00064     Studentas(const Studentas& other); // Copy konstruktorius
00065     Studentas& operator=(const Studentas& other); // Copy priskyrimo operatorius
00066     Studentas(Studentas&& other) noexcept; // Move konstruktorius
00067     Studentas& operator=(Studentas&& other) noexcept; // Move priskyrimo operatorius
00068     ~Studentas(); // Destruktorius
00069
00070     // Stream operatoriai
00071     friend ostream& operator<<(ostream& os, const Studentas& s);
00072     friend istream& operator>>(istream& is, Studentas& s);
00073
00074     // Getteriai (igyvendina Zmogus)
00075     inline string vardas() const override { return vardas_; }
00076     inline string pavarde() const override { return pavarde_; }
00077     inline int egzaminas() const { return egzaminas_; }
00078     inline const vector<float>& nd() const { return nd_; }
00079     inline float galutinisBalsas() const { return galutinis_balas_; }
00080
00081     // Setteriai (igyvendina Zmogus)
00082     void setVardas(const string& v) override { vardas_ = v; }
00083     void setPavarde(const string& p) override { pavarde_ = p; }
00084     void setEgzaminas(int e);
00085     void setNamudarbai(const vector<float>& nd);
00086
00087     float galBalas(float (*balasFunkcija)(const vector<float>&, int));
00088     istream& readStudent(istream&);
00089     static int destruktoriausk;
00090
00091     static void rusiuotiStud(vector<Studentas>& grupe, char rusiavimoPas);
00092     static void nuskaitymasFile(vector<Studentas>& grupe, const string& filename);
00093     static void spausdintiRez(vector<Studentas>& grupe, bool iFaila, char pasirinkimas, const
std::string& failoPavadinimas);
00094 };
00095
00096 #endif // STUDENTAS_H

```



## 6.7 include/testRuleOf5.h Failo Nuoroda

### Funkcijos

- void [testuotiRuleOfFive](#) ()

### 6.7.1 Funkcijos Dokumentacija

#### 6.7.1.1 testuotiRuleOfFive()

```
void testuotiRuleOfFive ()
```

## 6.8 testRuleOf5.h

Eiti į šio failo dokumentaciją.

```
00001 #ifndef TEST_RULE_OF_5_H
00002 #define TEST_RULE_OF_5_H
00003
00004 void testuotiRuleOfFive();
00005
00006 #endif
```

## 6.9 include/versijos1Funkc.h Failo Nuoroda

```
#include <iostream>
#include <vector>
#include <iomanip>
#include <string>
#include <fstream>
#include <limits>
#include <algorithm>
#include <cstdlib>
#include <ctime>
#include <sstream>
#include <filesystem>
#include <chrono>
#include <random>
#include <list>
#include <deque>
```

### Klasės

- struct [studentas< T >](#)

### Funkcijos

- template<typename Container>  
double [generuotiGalvid](#) (Container &nd, int egz)
- template<typename Container>  
double [generuotiGalmed](#) (Container nd)
- template<typename Container>  
double [pasirinktasGal](#) (Container &nd, int egz, char kaip)
- template<typename Container>  
void [randomPaz](#) (Container &nd, int &egz, int kiek\_nd)
- void [generuotiVardPav](#) (std::string &vardas, std::string &pavarde)
- template<typename Container>  
void [rusiuotiStud](#) (Container &grupe, char rusiavimoPas)

- `template<typename Container>`  
`void nuskaitymasFile (Container &grupe, const string &filename)`
- `void spausdintiRez2 (vector< studentas< vector< float > > > grupė, bool iFaila, char pasirinkimas, char rusiavimoPas)`
- `template<typename Container>`  
`void spausdintiRez (Container &grupe, bool iFaila, char pasirinkimas, const string &failoPavadinimas)`

## Kintamieji

- `std::stringstream buffer`

## 6.9.1 Funkcijos Dokumentacija

### 6.9.1.1 generuotiGalmed()

```
template<typename Container>
double generuotiGalmed (
    Container nd)
```

### 6.9.1.2 generuotiGalvid()

```
template<typename Container>
double generuotiGalvid (
    Container & nd,
    int egz)
```

### 6.9.1.3 generuotiVardPav()

```
void generuotiVardPav (
    std::string & vardas,
    std::string & pavarde)
```

### 6.9.1.4 nuskaitymasFile()

```
template<typename Container>
void nuskaitymasFile (
    Container & grupė,
    const string & filename)
```

### 6.9.1.5 pasirinktasGal()

```
template<typename Container>
double pasirinktasGal (
    Container & nd,
    int egz,
    char kaip)
```

### 6.9.1.6 randomPaz()

```
template<typename Container>
void randomPaz (
    Container & nd,
    int & egz,
    int kiek_nd)
```

### 6.9.1.7 rusiuotiStud()

```
template<typename Container>
void rusiuotiStud (
```

```
Container & grupe,
char rusiavimoPas)
```

### 6.9.1.8 spausdintiRez()

```
template<typename Container>
void spausdintiRez (
    Container & grupe,
    bool iFaila,
    char pasirinkimas,
    const string & failoPavadinimas)
```

### 6.9.1.9 spausdintiRez2()

```
void spausdintiRez2 (
    vector< studentas< vector< float > > > grupe,
    bool iFaila,
    char pasirinkimas,
    char rusiavimoPas)
```

## 6.9.2 Kintamojo Dokumentacija

### 6.9.2.1 buffer

```
std::stringstream buffer [extern]
```

## 6.10 versijos1Funkc.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef VERSIJOSFUNKC_H
00002 #define VERSIJOSFUNKC_H
00003
00004 #include <iostream>
00005 #include <vector>
00006 #include <iomanip>
00007 #include <string>
00008 #include <fstream>
00009 #include <limits>
00010 #include <algorithm>
00011 #include <cstdlib> // srand(), rand()
00012 #include <ctime>
00013 #include <sstream>
00014 #include <fstream>
00015 #include <filesystem>
00016 #include <chrono>
00017 #include <random>
00018 #include <list>
00019 #include <deque>
00020
00021 using std::cout;
00022 using std::cin;
00023 using std::vector;
00024 using std::setw;
00025 using std::endl;
00026 using std::string;
00027 using std::fstream;
00028 using std::fixed;
00029 using std::setprecision;
00030 using std::cerr;
00031 using std::getline;
00032 using std::ifstream;
00033 using std::ofstream;
00034 using std::ostream;
00035 using std::stringstream;
00036 using std::numeric_limits;
00037 using std::streamsize;
00038 using namespace std::chrono;
00039 using std::ostringstream;
00040 using std::istringstream;
00041 using std::move;
00042 extern std::stringstream buffer;
00043 using std::list;
00044 using std::deque;
```

```

00045 using std::string;
00046
00047 template <typename T=vector<float> > //jei T nepateikta, bus vector<float>
00048 struct studentas {
00049     using nd_type=T; //konteineris automatiškai priskiriamas is T
00050
00051     string Vard;
00052     string Pav;
00053     nd_type nd; //cia gali buti vector<float>, list<float> arba deque<float>
00054     int egz;
00055     double Gal;
00056 };
00057
00058
00059 template <typename Container>
00060 double generuotiGalvid(Container& nd, int egz) ;
00061
00062 template <typename Container>
00063 double generuotiGalmed(Container nd) ;
00064
00065 template <typename Container>
00066 double pasirinktasGal(Container& nd, int egz, char kaip);
00067
00068 template <typename Container>
00069 void randomPaz(Container& nd, int& egz, int kiek_nd) ;
00070
00071 void generuotiVardPav(std::string& vardas, std::string& pavarde);
00072
00073 template <typename Container>
00074 void rusiuotiStud(Container & grupe, char rusiavimoPas);
00075
00076 template <typename Container>
00077 void nuskaitymasFile(Container& grupe, const string& filename);
00078
00079 void spausdintiRez2( vector<studentas<vector<float>> grupe, bool iFaila, char pasirinkimas, char
    rusiavimoPas);
00080
00081 template<typename Container>
00082 void spausdintiRez(Container& grupe, bool iFaila, char pasirinkimas, const string& failoPavadinimas);
00083
00084
00085 #endif

```

## 6.11 include/zmogus.h Failo Nuoroda

```
#include <string>
```

### Klasės

- class [Zmogus](#)

## 6.12 zmogus.h

Eiti į šio failo dokumentaciją.

```

00001 #ifndef ZMOGUS_H
00002 #define ZMOGUS_H
00003
00004 #include <string>
00005 using std::string;
00006
00007 class Zmogus {
00008 protected:
00009     string vardas_;
00010     string pavarde_;
00011
00012 public:
00013     Zmogus()=default;
00014     Zmogus(const string& v, const string& p) : vardas_(v), pavarde_(p) {}
00015
00016     virtual ~Zmogus()= default;
00017
00018     virtual string vardas() const= 0; //0 - ABSTRAKTI FUNKC
00019     virtual string pavarde() const= 0;
00020
00021     virtual void setVardas(const string&)= 0;
00022     virtual void setPavarde(const string&)= 0;
00023 };

```

```
00024
00025 #endif
```

## 6.13 objektinisProg/README.md Failo Nuoroda

## 6.14 README.md Failo Nuoroda

## 6.15 Src/LaikoMatavimas.cpp Failo Nuoroda

```
#include "LaikoMatavimas.h"
```

## 6.16 Src/strategija3.cpp Failo Nuoroda

```
#include "strategija3.h"
```

### Funkcijos

- void `spausdintiKietiakusIrVargsius` (vector< `Studentas` > &vargsiai, vector< `Studentas` > &kietiakai, const string &vargsiuFailas, const string &kietiakuFailas)
- void `isskirtiVargsusIrKietiakusOpt` (vector< `Studentas` > &grupe, vector< `Studentas` > &vargsiai, vector< `Studentas` > &kietiakai)
- void `testuotiSkaidymoStrategija3` (const string &failoPavadinimas, const string &rezultataiAplankas)

### 6.16.1 Funkcijos Dokumentacija

#### 6.16.1.1 isskirtiVargsusIrKietiakusOpt()

```
void isskirtiVargsusIrKietiakusOpt (
    vector< Studentas > & grupe,
    vector< Studentas > & vargsiai,
    vector< Studentas > & kietiakai)
```

#### 6.16.1.2 spausdintiKietiakusIrVargsius()

```
void spausdintiKietiakusIrVargsius (
    vector< Studentas > & vargsiai,
    vector< Studentas > & kietiakai,
    const string & vargsiuFailas,
    const string & kietiakuFailas)
```

#### 6.16.1.3 testuotiSkaidymoStrategija3()

```
void testuotiSkaidymoStrategija3 (
    const string & failoPavadinimas,
    const string & rezultataiAplankas)
```

## 6.17 Src/studentas.cpp Failo Nuoroda

```
#include "studentas.h"
#include "zmogus.h"
```

## Funkcijos

- ostream & operator<< (ostream &os, const [Studentas](#) &s)
- istream & operator>> (istream &is, [Studentas](#) &s)
- float [generuotiGalvid](#) (const vector< float > &nd, int egz)
- float [generuotiGalmed](#) (const vector< float > &nd, int egz)

## 6.17.1 Funkcijos Dokumentacija

### 6.17.1.1 generuotiGalmed()

```
float generuotiGalmed (
    const vector< float > & nd,
    int egz)
```

### 6.17.1.2 generuotiGalvid()

```
float generuotiGalvid (
    const vector< float > & nd,
    int egz)
```

### 6.17.1.3 operator<<()

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

### 6.17.1.4 operator>>()

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

## 6.18 Src/testRuleOf5.cpp Failo Nuoroda

```
#include "studentas.h"
#include "testRuleOf5.h"
#include <cassert>
```

## Funkcijos

- void [testuotiRuleOfFive](#) ()

## 6.18.1 Funkcijos Dokumentacija

### 6.18.1.1 testuotiRuleOfFive()

```
void testuotiRuleOfFive ()
```

## 6.19 Src/v1.2.cpp Failo Nuoroda

```
#include "studentas.h"
#include "LaikoMatavimas.h"
#include "strategija3.h"
#include "versijos1Funkc.h"
#include "testRuleOf5.h"
```

## Funkcijos

- void vykdytiPrograma ()
- void meniu ()
- string pasirinktiFaila ()
- int main ()

### 6.19.1 Funkcijos Dokumentacija

#### 6.19.1.1 main()

```
int main ()
```

#### 6.19.1.2 meniu()

```
void meniu ()
```

#### 6.19.1.3 pasirinktiFaila()

```
string pasirinktiFaila ()
```

#### 6.19.1.4 vykdytiPrograma()

```
void vykdytiPrograma ()
```

## 6.20 Src/versijos1Funkc.cpp Failo Nuoroda

```
#include "versijos1Funkc.h"
#include <fstream>
#include <sstream>
#include <iostream>
#include <algorithm>
#include <cstdlib>
#include <string>
#include <numeric>
```

## Funkcijos

- template<typename Container>  
double generuotiGalvid (Container &nd, int egz)
- template<typename Container>  
double generuotiGalmed (Container nd)
- template<typename Container>  
double pasirinktasGal (Container &nd, int egz, char kaip)
- template<typename Container>  
void randomPaz (Container &nd, int &egz, int kiek\_nd)
- void generuotiVardPav (std::string &vardas, std::string &pavarde)
- template<typename Container>  
void rusiuotiStud (Container &grupe, char rusiavimoPas)
- template<typename Container>  
void nuskaitymasFile (Container &grupe, const string &filename)
- void spausdintiRez2 (vector< studentas > > > grupe, bool iFaila, char pasirinkimas, char rusiavimoPas)
- template<typename Container>  
void spausdintiRez (Container &grupe, bool iFaila, char pasirinkimas, const string &failoPavadinimas)
- template double pasirinktasGal< std::vector< float > > (std::vector< float > &, int, char)
- template double pasirinktasGal< std::list< float > > (std::list< float > &, int, char)

- template double `pasirinktasGal< std::deque< float > >` (std::deque< float > &, int, char)
- template void `randomPaz< std::vector< float > >` (std::vector< float > &, int &, int)
- template void `randomPaz< std::list< float > >` (std::list< float > &, int &, int)
- template void `randomPaz< std::deque< float > >` (std::deque< float > &, int &, int)
- template void `rusiuotiStud< std::vector< studentas< std::vector< float > > > >` (std::vector< studentas< std::vector< float > > > &, char)
- template void `rusiuotiStud< std::list< studentas< std::list< float > > > >` (std::list< studentas< std::list< float > > > &, char)
- template void `rusiuotiStud< std::deque< studentas< std::deque< float > > > >` (std::deque< studentas< std::deque< float > > > &, char)
- template void `nuskaitymasFile< std::vector< studentas< std::vector< float > > > >` (std::vector< studentas< std::vector< float > > > &, const std::string &)
- template void `nuskaitymasFile< std::list< studentas< std::list< float > > > >` (std::list< studentas< std::list< float > > > &, const std::string &)
- template void `nuskaitymasFile< std::deque< studentas< std::deque< float > > > >` (std::deque< studentas< std::deque< float > > > &, const std::string &)
- template void `spausdintiRez< std::vector< studentas< std::vector< float > > > >` (std::vector< studentas< std::vector< float > > > &, bool, char, const std::string &)
- template void `spausdintiRez< std::list< studentas< std::list< float > > > >` (std::list< studentas< std::list< float > > > &, bool, char, const std::string &)
- template void `spausdintiRez< std::deque< studentas< std::deque< float > > > >` (std::deque< studentas< std::deque< float > > > &, bool, char, const std::string &)

## 6.20.1 Funkcijos Dokumentacija

### 6.20.1.1 generuotiGalmed()

```
template<typename Container>
double generuotiGalmed (
    Container nd)
```

### 6.20.1.2 generuotiGalvid()

```
template<typename Container>
double generuotiGalvid (
    Container & nd,
    int egz)
```

### 6.20.1.3 generuotiVardPav()

```
void generuotiVardPav (
    std::string & vardas,
    std::string & pavarde)
```

### 6.20.1.4 nuskaitymasFile()

```
template<typename Container>
void nuskaitymasFile (
    Container & grupe,
    const string & filename)
```

### 6.20.1.5 nuskaitymasFile< std::deque< studentas< std::deque< float > > > >()

```
template void nuskaitymasFile< std::deque< studentas< std::deque< float > > > > (
    std::deque< studentas< std::deque< float > > > & ,
    const std::string & )
```



**6.20.1.6 nuskaitymasFile< std::list< studentas< std::list< float > > > >()**

```
template void nuskaitymasFile< std::list< studentas< std::list< float > > > > (
    std::list< studentas< std::list< float > > > & ,
    const std::string & )
```

**6.20.1.7 nuskaitymasFile< std::vector< studentas< std::vector< float > > > >()**

```
template void nuskaitymasFile< std::vector< studentas< std::vector< float > > > > (
    std::vector< studentas< std::vector< float > > > & ,
    const std::string & )
```

**6.20.1.8 pasirinktasGal()**

```
template<typename Container>
double pasirinktasGal (
    Container & nd,
    int egz,
    char kaip)
```

**6.20.1.9 pasirinktasGal< std::deque< float > >()**

```
template double pasirinktasGal< std::deque< float > > (
    std::deque< float > & ,
    int ,
    char )
```

**6.20.1.10 pasirinktasGal< std::list< float > >()**

```
template double pasirinktasGal< std::list< float > > (
    std::list< float > & ,
    int ,
    char )
```

**6.20.1.11 pasirinktasGal< std::vector< float > >()**

```
template double pasirinktasGal< std::vector< float > > (
    std::vector< float > & ,
    int ,
    char )
```

**6.20.1.12 randomPaz()**

```
template<typename Container>
void randomPaz (
    Container & nd,
    int & egz,
    int kiek_nd)
```

**6.20.1.13 randomPaz< std::deque< float > >()**

```
template void randomPaz< std::deque< float > > (
    std::deque< float > & ,
    int & ,
    int )
```

**6.20.1.14 randomPaz< std::list< float > >()**

```
template void randomPaz< std::list< float > > (
    std::list< float > & ,
```

```
int & ,
int )
```

#### 6.20.1.15 randomPaz< std::vector< float > >()

```
template void randomPaz< std::vector< float > > (
    std::vector< float > & ,
    int & ,
    int )
```

#### 6.20.1.16 rusiuotiStud()

```
template<typename Container>
void rusiuotiStud (
    Container & grupe,
    char rusiavimoPas)
```

#### 6.20.1.17 rusiuotiStud< std::deque< studentas< std::deque< float > > > >()

```
template void rusiuotiStud< std::deque< studentas< std::deque< float > > > > (
    std::deque< studentas< std::deque< float > > > & ,
    char )
```

#### 6.20.1.18 rusiuotiStud< std::list< studentas< std::list< float > > > >()

```
template void rusiuotiStud< std::list< studentas< std::list< float > > > > (
    std::list< studentas< std::list< float > > > & ,
    char )
```

#### 6.20.1.19 rusiuotiStud< std::vector< studentas< std::vector< float > > > >()

```
template void rusiuotiStud< std::vector< studentas< std::vector< float > > > > (
    std::vector< studentas< std::vector< float > > > & ,
    char )
```

#### 6.20.1.20 spausdintiRez()

```
template<typename Container>
void spausdintiRez (
    Container & grupe,
    bool iFaila,
    char pasirinkimas,
    const string & failoPavadinimas)
```

#### 6.20.1.21 spausdintiRez2()

```
void spausdintiRez2 (
    vector< studentas< vector< float > > > grupe,
    bool iFaila,
    char pasirinkimas,
    char rusiavimoPas)
```

#### 6.20.1.22 spausdintiRez< std::deque< studentas< std::deque< float > > > >()

```
template void spausdintiRez< std::deque< studentas< std::deque< float > > > > (
    std::deque< studentas< std::deque< float > > > & ,
    bool ,
    char ,
    const std::string & )
```

**6.20.1.23 spausdintiRez< std::list< studentas< std::list< float > > > >()**

```
template void spausdintiRez< std::list< studentas< std::list< float > > > > (
    std::list< studentas< std::list< float > > > & ,
    bool ,
    char ,
    const std::string & )
```

**6.20.1.24 spausdintiRez< std::vector< studentas< std::vector< float > > > >()**

```
template void spausdintiRez< std::vector< studentas< std::vector< float > > > > (
    std::vector< studentas< std::vector< float > > > & ,
    bool ,
    char ,
    const std::string & )
```



# Rodyklė

~Studentas  
Studentas, [15](#)

~Zmogus  
Zmogus, [18](#)

baigti  
LaikoMatavimas, [13](#)

buffer  
studentas.h, [23](#)  
versijos1Funkc.h, [27](#)

destruktoriuSk  
Studentas, [17](#)

egz  
studentas< T >, [17](#)

egzaminas  
Studentas, [15](#)

Gal  
studentas< T >, [17](#)

galBalas  
Studentas, [15](#)

galutinisBalsas  
Studentas, [15](#)

gautiLaikoSkirtuma  
LaikoMatavimas, [13](#)

generuotiGalmed  
studentas.cpp, [30](#)  
studentas.h, [23](#)  
versijos1Funkc.cpp, [32](#)  
versijos1Funkc.h, [26](#)

generuotiGalvid  
studentas.cpp, [30](#)  
studentas.h, [23](#)  
versijos1Funkc.cpp, [32](#)  
versijos1Funkc.h, [26](#)

generuotiVardPav  
versijos1Funkc.cpp, [32](#)  
versijos1Funkc.h, [26](#)

include/LaikoMatavimas.h, [21](#)  
include/strategija3.h, [21](#), [22](#)  
include/studentas.h, [22](#), [23](#)  
include/testRuleOf5.h, [25](#)  
include/versijos1Funkc.h, [25](#), [27](#)  
include/zmogus.h, [28](#)  
isskirtiVargsusIrKietiakusOpt  
strategija3.cpp, [29](#)  
strategija3.h, [22](#)

LaikoMatavimas, [13](#)  
baigti, [13](#)  
gautiLaikoSkirtuma, [13](#)  
LaikoMatavimas, [13](#)  
pradeti, [13](#)

main  
v1.2.cpp, [31](#)

menu  
v1.2.cpp, [31](#)

nd  
Studentas, [15](#)  
studentas< T >, [17](#)

nd\_type  
studentas< T >, [17](#)

nuskaitymasFile  
Studentas, [15](#)  
versijos1Funkc.cpp, [32](#)  
versijos1Funkc.h, [26](#)

nuskaitymasFile< std::deque< studentas< std::deque<  
float > > > >  
versijos1Funkc.cpp, [32](#)

nuskaitymasFile< std::list< studentas< std::list< float  
> > > >  
versijos1Funkc.cpp, [32](#)

nuskaitymasFile< std::vector< studentas< std::vector<  
float > > > >  
versijos1Funkc.cpp, [33](#)

objektinisProg/README.md, [29](#)

objektinisProg2, [1](#)

operator<<  
Studentas, [16](#)  
studentas.cpp, [30](#)

operator>>  
Studentas, [16](#)  
studentas.cpp, [30](#)

operator=  
Studentas, [15](#)

pasirinktasGal  
versijos1Funkc.cpp, [33](#)  
versijos1Funkc.h, [26](#)

pasirinktasGal< std::deque< float > >  
versijos1Funkc.cpp, [33](#)

pasirinktasGal< std::list< float > >  
versijos1Funkc.cpp, [33](#)

pasirinktasGal< std::vector< float > >  
versijos1Funkc.cpp, [33](#)

pasirinktiFaila  
     v1.2.cpp, [31](#)  
 Pav  
     studentas< T >, [17](#)  
 pavarde  
     Studentas, [15](#)  
     Zmogus, [18](#)  
 pavarde\_  
     Zmogus, [19](#)  
 pradeti  
     LaikoMatavimas, [13](#)  
  
 randomPaz  
     versijos1Funkc.cpp, [33](#)  
     versijos1Funkc.h, [26](#)  
 randomPaz< std::deque< float > >  
     versijos1Funkc.cpp, [33](#)  
 randomPaz< std::list< float > >  
     versijos1Funkc.cpp, [33](#)  
 randomPaz< std::vector< float > >  
     versijos1Funkc.cpp, [34](#)  
 README.md, [29](#)  
 readStudent  
     Studentas, [16](#)  
 rusiuotiStud  
     Studentas, [16](#)  
     versijos1Funkc.cpp, [34](#)  
     versijos1Funkc.h, [26](#)  
 rusiuotiStud< std::deque< studentas< std::deque< float > > >  
     versijos1Funkc.cpp, [34](#)  
 rusiuotiStud< std::list< studentas< std::list< float > >  
     > >  
     versijos1Funkc.cpp, [34](#)  
 rusiuotiStud< std::vector< studentas< std::vector< float > > >  
     versijos1Funkc.cpp, [34](#)  
  
 setEgzaminas  
     Studentas, [16](#)  
 setNamudarbai  
     Studentas, [16](#)  
 setPavarde  
     Studentas, [16](#)  
     Zmogus, [18](#)  
 setVardas  
     Studentas, [16](#)  
     Zmogus, [19](#)  
 spausdintiKietiakuslrVargsius  
     strategija3.cpp, [29](#)  
     strategija3.h, [22](#)  
 spausdintiRez  
     Studentas, [16](#)  
     versijos1Funkc.cpp, [34](#)  
     versijos1Funkc.h, [27](#)  
 spausdintiRez< std::deque< studentas< std::deque< float > > >  
     versijos1Funkc.cpp, [34](#)  
  
 spausdintiRez< std::list< studentas< std::list< float >  
     > > >  
     versijos1Funkc.cpp, [34](#)  
 spausdintiRez< std::vector< studentas< std::vector< float > > >  
     versijos1Funkc.cpp, [35](#)  
 spausdintiRez2  
     versijos1Funkc.cpp, [34](#)  
     versijos1Funkc.h, [27](#)  
 Src/LaikoMatavimas.cpp, [29](#)  
 Src/strategija3.cpp, [29](#)  
 Src/studentas.cpp, [29](#)  
 Src/testRuleOf5.cpp, [30](#)  
 Src/v1.2.cpp, [30](#)  
 Src/versijos1Funkc.cpp, [31](#)  
 strategija3.cpp  
     isskirtiVargsuslrKietiakusOpt, [29](#)  
     spausdintiKietiakuslrVargsius, [29](#)  
     testuotiSkaidymoStrategija3, [29](#)  
 strategija3.h  
     isskirtiVargsuslrKietiakusOpt, [22](#)  
     spausdintiKietiakuslrVargsius, [22](#)  
     testuotiSkaidymoStrategija3, [22](#)  
 Studentas, [13](#)  
     ~Studentas, [15](#)  
     destruktoriuSk, [17](#)  
     egzaminas, [15](#)  
     galBalas, [15](#)  
     galutinisBalsas, [15](#)  
     nd, [15](#)  
     nuskaitymasFile, [15](#)  
     operator<<, [16](#)  
     operator>>, [16](#)  
     operator=, [15](#)  
     pavarde, [15](#)  
     readStudent, [16](#)  
     rusiuotiStud, [16](#)  
     setEgzaminas, [16](#)  
     setNamudarbai, [16](#)  
     setPavarde, [16](#)  
     setVardas, [16](#)  
     spausdintiRez, [16](#)  
     Studentas, [15](#)  
     vardas, [16](#)  
 studentas< T >, [17](#)  
     egz, [17](#)  
     Gal, [17](#)  
     nd, [17](#)  
     nd\_type, [17](#)  
     Pav, [17](#)  
     Vard, [17](#)  
 studentas.cpp  
     generuotiGalmed, [30](#)  
     generuotiGalvid, [30](#)  
     operator<<, [30](#)  
     operator>>, [30](#)  
 studentas.h  
     buffer, [23](#)

generuotiGalmed, [23](#)  
 generuotiGalvid, [23](#)  
 testRuleOf5.cpp  
   testuotiRuleOfFive, [30](#)  
 testRuleOf5.h  
   testuotiRuleOfFive, [25](#)  
 testuotiRuleOfFive  
   testRuleOf5.cpp, [30](#)  
   testRuleOf5.h, [25](#)  
 testuotiSkaidymoStrategija3  
   strategija3.cpp, [29](#)  
   strategija3.h, [22](#)  
 v1.2.cpp  
   main, [31](#)  
   menu, [31](#)  
   pasirinktiFaila, [31](#)  
   vykdytiPrograma, [31](#)  
 Vard  
   studentas< T >, [17](#)  
 vardas  
   Studentas, [16](#)  
   Zmogus, [19](#)  
 vardas\_  
   Zmogus, [19](#)  
 versijos1Funkc.cpp  
   generuotiGalmed, [32](#)  
   generuotiGalvid, [32](#)  
   generuotiVardPav, [32](#)  
   nuskaitymasFile, [32](#)  
   nuskaitymasFile< std::deque< studentas<  
     std::deque< float > > >, [32](#)  
   nuskaitymasFile< std::list< studentas< std::list<  
     float > > >, [32](#)  
   nuskaitymasFile< std::vector< studentas<  
     std::vector< float > > >, [33](#)  
   pasirinktasGal, [33](#)  
   pasirinktasGal< std::deque< float > >, [33](#)  
   pasirinktasGal< std::list< float > >, [33](#)  
   pasirinktasGal< std::vector< float > >, [33](#)  
   randomPaz, [33](#)  
   randomPaz< std::deque< float > >, [33](#)  
   randomPaz< std::list< float > >, [33](#)  
   randomPaz< std::vector< float > >, [34](#)  
   rusiuotiStud, [34](#)  
   rusiuotiStud< std::deque< studentas< std::deque<  
     float > > >, [34](#)  
   rusiuotiStud< std::list< studentas< std::list< float  
     > > >, [34](#)  
   rusiuotiStud< std::vector< studentas< std::vector<  
     float > > >, [34](#)  
   spausdintiRez, [34](#)  
   spausdintiRez< std::deque< studentas< std::deque<  
     float > > >, [34](#)  
   spausdintiRez< std::list< studentas< std::list<  
     float > > >, [34](#)  
   spausdintiRez< std::vector< studentas< std::vector<  
     float > > >, [35](#)  
   spausdintiRez2, [34](#)  
 versijos1Funkc.h  
   buffer, [27](#)  
   generuotiGalmed, [26](#)  
   generuotiGalvid, [26](#)  
   generuotiVardPav, [26](#)  
   nuskaitymasFile, [26](#)  
   pasirinktasGal, [26](#)  
   randomPaz, [26](#)  
   rusiuotiStud, [26](#)  
   spausdintiRez, [27](#)  
   spausdintiRez2, [27](#)  
 vykdytiPrograma  
   v1.2.cpp, [31](#)  
 Zmogus, [18](#)  
   ~Zmogus, [18](#)  
   pavarde, [18](#)  
   pavarde\_, [19](#)  
   setPavarde, [18](#)  
   setVardas, [19](#)  
   vardas, [19](#)  
   vardas\_, [19](#)  
   Zmogus, [18](#)