

Darbo su studentų duomenimis programa

Sugeneruota Doxygen 1.13.2

1 Hierarchijos Indeksas	1
1.1 Klasių hierarchija	1
2 Duomenų Struktūros Indeksas	3
2.1 Duomenų Struktūros	3
3 Failo Indeksas	5
3.1 Failai	5
4 Duomenų Struktūros Dokumentacija	7
4.1 Stud Klasė	7
4.1.1 Konstruktoriaus ir Destruktoriaus Dokumentacija	8
4.1.1.1 Stud() [1/4]	8
4.1.1.2 Stud() [2/4]	8
4.1.1.3 ~Stud()	8
4.1.1.4 Stud() [3/4]	8
4.1.1.5 Stud() [4/4]	8
4.1.2 Metodų Dokumentacija	9
4.1.2.1 getGalutinisMed()	9
4.1.2.2 getGalutinisVid()	9
4.1.2.3 getPavarde()	9
4.1.2.4 getVardas()	9
4.1.2.5 operator=() [1/2]	9
4.1.2.6 operator=() [2/2]	9
4.1.2.7 readStudent()	9
4.1.2.8 skaiciuotiGalutini()	9
4.1.3 Draugiškų Ir Susijusių Funkcijų Dokumentacija	10
4.1.3.1 operator<<	10
4.1.3.2 operator>>	10
4.1.4 Lauko aprašymas	10
4.1.4.1 egzaminas	10
4.1.4.2 galutinis_med	10
4.1.4.3 galutinis_vid	10
4.1.4.4 pazymiai	10
4.2 Vector< T > Klasė Šablonas	10
4.2.1 Konstruktoriaus ir Destruktoriaus Dokumentacija	11
4.2.1.1 Vector() [1/3]	11
4.2.1.2 ~Vector()	12
4.2.1.3 Vector() [2/3]	12
4.2.1.4 Vector() [3/3]	12
4.2.2 Metodų Dokumentacija	12
4.2.2.1 append_range()	12
4.2.2.2 assign() [1/3]	12

4.2.2.3 assign() [2/3]	12
4.2.2.4 assign() [3/3]	12
4.2.2.5 at()	13
4.2.2.6 back()	13
4.2.2.7 begin()	13
4.2.2.8 clear()	13
4.2.2.9 empty()	13
4.2.2.10 end()	13
4.2.2.11 erase()	13
4.2.2.12 front()	13
4.2.2.13 getCapacity()	14
4.2.2.14 getData()	14
4.2.2.15 getSize()	14
4.2.2.16 insert()	14
4.2.2.17 max_size()	14
4.2.2.18 operator!=()	14
4.2.2.19 operator<()	14
4.2.2.20 operator<=()	14
4.2.2.21 operator=() [1/2]	15
4.2.2.22 operator=() [2/2]	15
4.2.2.23 operator==()	15
4.2.2.24 operator>()	15
4.2.2.25 operator>=()	15
4.2.2.26 operator[]()	15
4.2.2.27 pop_back()	15
4.2.2.28 push_back()	15
4.2.2.29 rbegin()	16
4.2.2.30 rend()	16
4.2.2.31 reserve()	16
4.2.2.32 resize()	16
4.2.2.33 shrink_to_fit()	16
4.2.2.34 swap()	16
4.2.3 Lauko aprašymas	16
4.2.3.1 capacity	16
4.2.3.2 data	16
4.2.3.3 size	17
4.3 Zmogus Klasė	17
4.3.1 Konstruktoriaus ir Destruktoriaus Dokumentacija	17
4.3.1.1 Zmogus() [1/4]	17
4.3.1.2 Zmogus() [2/4]	18
4.3.1.3 Zmogus() [3/4]	18
4.3.1.4 Zmogus() [4/4]	18

4.3.1.5 ~Zmogus()	18
4.3.2 Metodų Dokumentacija	18
4.3.2.1 getPavarde()	18
4.3.2.2 getVardas()	18
4.3.2.3 operator=() [1/2]	18
4.3.2.4 operator=() [2/2]	18
4.3.2.5 setPavarde()	19
4.3.2.6 setVardas()	19
4.3.3 Lauko aprašymas	19
4.3.3.1 pavarde	19
4.3.3.2 vardas	19
5 Failo Dokumentacija	21
5.1 include/headers_main.h Failo Nuoroda	21
5.2 headers_main.h	21
5.3 include/isvesties_rezimai.h Failo Nuoroda	22
5.3.1 Funkcijos Dokumentacija	22
5.3.1.1 isvestis()	22
5.4 isvesties_rezimai.h	22
5.5 include/ivesties_rezimai.h Failo Nuoroda	22
5.5.1 Funkcijos Dokumentacija	23
5.5.1.1 failu_generavimas()	23
5.5.1.2 pazymiu_generavimas()	23
5.5.1.3 ranka()	23
5.5.1.4 skaitymas_is_failo()	23
5.5.1.5 strategija_1()	23
5.5.1.6 studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius()	23
5.5.1.7 visko_generavimas()	23
5.6 ivesties_rezimai.h	24
5.7 include/ivesties_tikrinimas.h Failo Nuoroda	24
5.7.1 Funkcijos Dokumentacija	24
5.7.1.1 ivesties_tikrinimas()	24
5.8 ivesties_tikrinimas.h	24
5.9 include/rikiavimo_funckija.h Failo Nuoroda	25
5.9.1 Funkcijos Dokumentacija	25
5.9.1.1 raktas_gm()	25
5.9.1.2 raktas_gv()	25
5.9.1.3 raktas_p()	25
5.9.1.4 raktas_v()	25
5.9.1.5 rikiavimas()	25
5.10 rikiavimo_funckija.h	26
5.11 include/skaiciavimo_funkcijos.h Failo Nuoroda	26

5.11.1 Funkcijos Dokumentacija	26
5.11.1.1 mediana()	26
5.11.1.2 vidurkis()	26
5.12 skaiciavimo_funkcijos.h	26
5.13 include/studentas.h Failo Nuoroda	27
5.14 studentas.h	27
5.15 include/studentas_testai.h Failo Nuoroda	28
5.15.1 Funkcijos Dokumentacija	28
5.15.1.1 testAll()	28
5.15.1.2 testAssignmentOperator()	29
5.15.1.3 testCopyConstructor()	29
5.15.1.4 testDestructor()	29
5.15.1.5 testEmptyConstructor()	29
5.15.1.6 testInputOperator()	29
5.15.1.7 testInputStreamConstructor()	29
5.15.1.8 testMoveAssignmentOperator()	29
5.15.1.9 testMoveConstructor()	29
5.15.1.10 testOutputOperator()	29
5.16 studentas_testai.h	30
5.17 include/testavimo_funkcijos.h Failo Nuoroda	30
5.17.1 Funkcijos Dokumentacija	30
5.17.1.1 duomenu_apdorojimo_testavimas()	30
5.17.1.2 failo_kurimo_testavimas()	30
5.17.1.3 konteinerio_testavimas()	30
5.17.1.4 perskirstymu_testas()	31
5.17.1.5 testavimas()	31
5.17.1.6 testinis_skaitymas_is_failo()	31
5.18 testavimo_funkcijos.h	31
5.19 include/vartotojo_pasirinkimo_funkcijos.h Failo Nuoroda	31
5.19.1 Funkcijos Dokumentacija	31
5.19.1.1 failo_pasirinkimas()	31
5.19.1.2 isvesties_pasirinkimas()	32
5.19.1.3 rezimo_pasirinkimas()	32
5.20 vartotojo_pasirinkimo_funkcijos.h	32
5.21 include/vector.h Failo Nuoroda	32
5.22 vector.h	32
5.23 include/zmogus.h Failo Nuoroda	33
5.24 zmogus.h	33
5.25 source/isvesties_rezimai.cpp Failo Nuoroda	34
5.25.1 Funkcijos Dokumentacija	34
5.25.1.1 isvestis()	34
5.26 source/ivesties_rezimai.cpp Failo Nuoroda	34

5.26.1 Funkcijos Dokumentacija	35
5.26.1.1 failu_generavimas()	35
5.26.1.2 pazymiu_generavimas()	35
5.26.1.3 ranka()	35
5.26.1.4 skaitymas_is_failo()	35
5.26.1.5 strategija_1()	35
5.26.1.6 studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius()	36
5.26.1.7 visko_generavimas()	36
5.27 source/ivesties_tikrinimas.cpp Failo Nuoroda	36
5.27.1 Funkcijos Dokumentacija	36
5.27.1.1 ivesties_tikrinimas()	36
5.27.1.2 ivesties_tikrinimas< int >()	36
5.27.1.3 ivesties_tikrinimas< string >()	36
5.28 source/main.cpp Failo Nuoroda	37
5.28.1 Funkcijos Dokumentacija	37
5.28.1.1 main()	37
5.29 source/rikiavimo_funkcija.cpp Failo Nuoroda	37
5.29.1 Funkcijos Dokumentacija	37
5.29.1.1 raktas_gm()	37
5.29.1.2 raktas_gv()	37
5.29.1.3 raktas_p()	38
5.29.1.4 raktas_v()	38
5.29.1.5 rikiavimas()	38
5.30 source/skaiciavimo_funkcijos.cpp Failo Nuoroda	38
5.30.1 Funkcijos Dokumentacija	38
5.30.1.1 mediana()	38
5.30.1.2 vidurkis()	38
5.31 source/studentas.cpp Failo Nuoroda	39
5.32 source/studentas_testai.cpp Failo Nuoroda	39
5.32.1 Funkcijos Dokumentacija	39
5.32.1.1 testAll()	39
5.32.1.2 testAssignmentOperator()	39
5.32.1.3 testCopyConstructor()	39
5.32.1.4 testDestructor()	39
5.32.1.5 testEmptyConstructor()	40
5.32.1.6 testInputOperator()	40
5.32.1.7 testInputStreamConstructor()	40
5.32.1.8 testMoveAssignmentOperator()	40
5.32.1.9 testMoveConstructor()	40
5.32.1.10 testOutputOperator()	40
5.33 source/testavimo_funkcijos.cpp Failo Nuoroda	40
5.33.1 Funkcijos Dokumentacija	41

5.33.1.1 duomenų_apdorojimo_testavimas()	41
5.33.1.2 failo_kurimo_testavimas()	41
5.33.1.3 konteinerio_testavimas()	41
5.33.1.4 perskirstymo_testas()	41
5.33.1.5 testavimas()	41
5.33.1.6 testinis_skaitymas_is_failo()	41
5.34 source/vartotojo_pasirinkimo_funkcijos.cpp Failo Nuoroda	41
5.34.1 Funkcijos Dokumentacija	42
5.34.1.1 failo_pasirinkimas()	42
5.34.1.2 isvesties_pasirinkimas()	42
5.34.1.3 rezimo_pasirinkimas()	42
Rodyklė	43

skyrius 1

Hierarchijos Indeksas

1.1 Klasių hierarchija

Šis paveldėjimo sąrašas yra beveik surikiuotas abėcėlės tvarka:

Vector< T >	10
Zmogus	17
Stud	7

skyrius 2

Duomenų Struktūros Indeksas

2.1 Duomenų Struktūros

Duomenų struktūros su trumpais aprašymais:

Stud	7
Vector< T >	10
Zmogus	17

skyrius 3

Failo Indeksas

3.1 Failai

Visų failų sąrašas su trumpais aprašymais:

include/headers_main.h	21
include/isvesties_rezimai.h	22
include/ivesties_rezimai.h	22
include/ivesties_tikrinimas.h	24
include/rikiavimo_funkcija.h	25
include/skaiciavimo_funkcijos.h	26
include/studentas.h	27
include/studentas_testai.h	28
include/testavimo_funkcijos.h	30
include/vartotojo_pasirinkimo_funkcijos.h	31
include/vector.h	32
include/zmogus.h	33
source/isvesties_rezimai.cpp	34
source/ivesties_rezimai.cpp	34
source/ivesties_tikrinimas.cpp	36
source/main.cpp	37
source/rikiavimo_funkcija.cpp	37
source/skaiciavimo_funkcijos.cpp	38
source/studentas.cpp	39
source/studentas_testai.cpp	39
source/testavimo_funkcijos.cpp	40
source/vartotojo_pasirinkimo_funkcijos.cpp	41

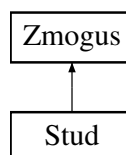
skyrius 4

Duomenų Struktūros Dokumentacija

4.1 Stud Klasė

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

Paveldimumo diagrama Stud:



Vieši Metodai

- `Stud ()`
- `Stud (std::istream &is, string rezimas)`
- string `getVardas ()` const override
- string `getPavarde ()` const override
- double `getGalutinisVid ()` const
- double `getGalutinisMed ()` const
- `std::istream & readStudent (std::istream &, string rezimas)`
- void `skaiciuotiGalutini ()`
- `~Stud ()`
- `Stud (const Stud &s)`
- `Stud & operator= (const Stud &s)`
- `Stud (Stud &&s)` noexcept
- `Stud & operator= (Stud &&s)` noexcept

Vieši Metodai inherited from `Zmogus`

- `Zmogus (string v, string p)`
- `Zmogus ()`
- `Zmogus (const Zmogus &z)`
- `Zmogus & operator= (const Zmogus &z)`
- `Zmogus (Zmogus &&z)` noexcept
- `Zmogus & operator= (Zmogus &&z)` noexcept
- virtual `~Zmogus ()`
- void `setVardas (const string &v)`
- void `setPavarde (const string &p)`

Privatūs Atributai

- `Vector< int > pazymiai`
- `int egzaminas`
- `double galutinis_vid == -1`
- `double galutinis_med == -1`

Draugai

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

Additional Inherited Members**Apsaugoti Atributai inherited from Zmogus**

- `string vardas`
- `string pavarde`

4.1.1 Konstruktorius ir Destruktorius Dokumentacija**4.1.1.1 Stud() [1/4]**

```
Stud::Stud () [inline]
```

4.1.1.2 Stud() [2/4]

```
Stud::Stud (
    std::istream & is,
    string rezimas)
```

4.1.1.3 ~Stud()

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

4.1.1.4 Stud() [3/4]

```
Stud::Stud (
    const Stud & s) [inline]
```

4.1.1.5 Stud() [4/4]

```
Stud::Stud (
    Stud && s) [inline], [noexcept]
```


4.1.2 Metodų Dokumentacija

4.1.2.1 getGalutinisMed()

```
double Stud::getGalutinisMed () const [inline]
```

4.1.2.2 getGalutinisVid()

```
double Stud::getGalutinisVid () const [inline]
```

4.1.2.3 getPavarde()

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

Realizuoja [Zmogus](#).

4.1.2.4 getVardas()

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

Realizuoja [Zmogus](#).

4.1.2.5 operator=() [1/2]

```
Stud & Stud::operator= (  
    const Stud & s) [inline]
```

4.1.2.6 operator=() [2/2]

```
Stud & Stud::operator= (  
    Stud && s) [inline], [noexcept]
```

4.1.2.7 readStudent()

```
std::istream & Stud::readStudent (  
    std::istream & is,  
    string rezimas)
```

4.1.2.8 skaiciuotiGalutini()

```
void Stud::skaiciuotiGalutini ()
```

4.1.3 Draugiškų Ir Susijusių Funkcijų Dokumentacija

4.1.3.1 operator<<

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

4.1.3.2 operator>>

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

4.1.4 Lauko aprašymas

4.1.4.1 egzaminas

```
int Stud::egzaminas [private]
```

4.1.4.2 galutinis_med

```
double Stud::galutinis_med = -1 [private]
```

4.1.4.3 galutinis_vid

```
double Stud::galutinis_vid = -1 [private]
```

4.1.4.4 pazymiai

```
Vector<int> Stud::pazymiai [private]
```

Dokumentacija šiai klasei sugeneruota iš šių failų:

- include/[studentas.h](#)
- source/[studentas.cpp](#)

4.2 Vector< T > Klasė Šablonas

```
#include <vector.h>
```

Vieši Metodai

- `Vector` (size_t init_capacity=10)
- `~Vector` ()
- `Vector` (const `Vector` &other)
- `Vector & operator=` (const `Vector` &other)
- `Vector` (`Vector` &&other) noexcept
- `Vector & operator=` (`Vector` &&other) noexcept
- void `assign` (size_t count, const T &value)
- void `assign` (const T *first, const T *last)
- void `assign` (const `Vector` &other)
- T & `at` (size_t index)
- T & `operator[]` (size_t index)
- T & `front` ()
- T & `back` ()
- T * `getData` ()
- T * `begin` ()
- T * `end` ()
- T * `rbegin` ()
- T * `rend` ()
- bool `empty` () const
- size_t `getSize` () const
- size_t `max_size` () const
- void `reserve` (size_t new_capacity)
- size_t `getCapacity` () const
- void `shrink_to_fit` ()
- void `clear` ()
- void `insert` (size_t index, const T &value)
- void `erase` (size_t index)
- void `push_back` (const T &value)
- template<typename Range>
void `append_range` (const Range &other)
- void `pop_back` ()
- void `resize` (size_t new_size)
- void `swap` (`Vector` &other)
- bool `operator==` (const `Vector` &other) const
- bool `operator!=` (const `Vector` &other) const
- bool `operator<` (const `Vector` &other) const
- bool `operator<=` (const `Vector` &other) const
- bool `operator>` (const `Vector` &other) const
- bool `operator>=` (const `Vector` &other) const

Privatūs Atributai

- T * `data`
- size_t `size`
- size_t `capacity`

4.2.1 Konstruktorius ir Destruktorius Dokumentacija**4.2.1.1 Vector() [1/3]**

```
template<typename T>
Vector< T >::Vector (
    size_t init_capacity = 10)
```

4.2.1.2 ~Vector()

```
template<typename T>
Vector< T >::~~Vector ()
```

4.2.1.3 Vector() [2/3]

```
template<typename T>
Vector< T >::Vector (
    const Vector< T > & other)
```

4.2.1.4 Vector() [3/3]

```
template<typename T>
Vector< T >::Vector (
    Vector< T > && other) [noexcept]
```

4.2.2 Metodų Dokumentacija

4.2.2.1 append_range()

```
template<typename T>
template<typename Range>
void Vector< T >::append_range (
    const Range & other)
```

4.2.2.2 assign() [1/3]

```
template<typename T>
void Vector< T >::assign (
    const T * first,
    const T * last)
```

4.2.2.3 assign() [2/3]

```
template<typename T>
void Vector< T >::assign (
    const Vector< T > & other)
```

4.2.2.4 assign() [3/3]

```
template<typename T>
void Vector< T >::assign (
    size_t count,
    const T & value)
```

4.2.2.5 at()

```
template<typename T>
T & Vector< T >::at (
    size_t index)
```

4.2.2.6 back()

```
template<typename T>
T & Vector< T >::back ()
```

4.2.2.7 begin()

```
template<typename T>
T * Vector< T >::begin ()
```

4.2.2.8 clear()

```
template<typename T>
void Vector< T >::clear ()
```

4.2.2.9 empty()

```
template<typename T>
bool Vector< T >::empty () const
```

4.2.2.10 end()

```
template<typename T>
T * Vector< T >::end ()
```

4.2.2.11 erase()

```
template<typename T>
void Vector< T >::erase (
    size_t index)
```

4.2.2.12 front()

```
template<typename T>
T & Vector< T >::front ()
```

4.2.2.13 getCapacity()

```
template<typename T>
size_t Vector< T >::getCapacity () const
```

4.2.2.14 getData()

```
template<typename T>
T * Vector< T >::getData ()
```

4.2.2.15 getSize()

```
template<typename T>
size_t Vector< T >::getSize () const
```

4.2.2.16 insert()

```
template<typename T>
void Vector< T >::insert (
    size_t index,
    const T & value)
```

4.2.2.17 max_size()

```
template<typename T>
size_t Vector< T >::max_size () const
```

4.2.2.18 operator"!="()

```
template<typename T>
bool Vector< T >::operator!= (
    const Vector< T > & other) const
```

4.2.2.19 operator<()

```
template<typename T>
bool Vector< T >::operator< (
    const Vector< T > & other) const
```

4.2.2.20 operator<=()

```
template<typename T>
bool Vector< T >::operator<= (
    const Vector< T > & other) const
```

4.2.2.21 operator=() [1/2]

```
template<typename T>
Vector & Vector< T >::operator= (
    const Vector< T > & other)
```

4.2.2.22 operator=() [2/2]

```
template<typename T>
Vector & Vector< T >::operator= (
    Vector< T > && other) [noexcept]
```

4.2.2.23 operator==()

```
template<typename T>
bool Vector< T >::operator== (
    const Vector< T > & other) const
```

4.2.2.24 operator>()

```
template<typename T>
bool Vector< T >::operator> (
    const Vector< T > & other) const
```

4.2.2.25 operator>=()

```
template<typename T>
bool Vector< T >::operator>= (
    const Vector< T > & other) const
```

4.2.2.26 operator[]()

```
template<typename T>
T & Vector< T >::operator[] (
    size_t index)
```

4.2.2.27 pop_back()

```
template<typename T>
void Vector< T >::pop_back ()
```

4.2.2.28 push_back()

```
template<typename T>
void Vector< T >::push_back (
    const T & value)
```

4.2.2.29 rbegin()

```
template<typename T>
T * Vector< T >::rbegin ()
```

4.2.2.30 rend()

```
template<typename T>
T * Vector< T >::rend ()
```

4.2.2.31 reserve()

```
template<typename T>
void Vector< T >::reserve (
    size_t new_capacity)
```

4.2.2.32 resize()

```
template<typename T>
void Vector< T >::resize (
    size_t new_size)
```

4.2.2.33 shrink_to_fit()

```
template<typename T>
void Vector< T >::shrink_to_fit ()
```

4.2.2.34 swap()

```
template<typename T>
void Vector< T >::swap (
    Vector< T > & other)
```

4.2.3 Lauko aprašymas

4.2.3.1 capacity

```
template<typename T>
size_t Vector< T >::capacity [private]
```

4.2.3.2 data

```
template<typename T>
T* Vector< T >::data [private]
```


4.2.3.3 size

```
template<typename T>
size_t Vector< T >::size [private]
```

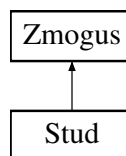
Dokumentacija šiai klasei sugeneruota iš šio failo:

- include/[vector.h](#)

4.3 Zmogus Klasė

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

Paveldimumo diagrama Zmogus:



Vieši Metodai

- [Zmogus](#) (string v, string p)
- [Zmogus](#) ()
- [Zmogus](#) (const [Zmogus](#) &z)
- [Zmogus](#) & operator= (const [Zmogus](#) &z)
- [Zmogus](#) ([Zmogus](#) &&z) noexcept
- [Zmogus](#) & operator= ([Zmogus](#) &&z) noexcept
- virtual [~Zmogus](#) ()
- virtual string [getVardas](#) () const =0
- virtual string [getPavarde](#) () const =0
- void [setVardas](#) (const string &v)
- void [setPavarde](#) (const string &p)

Apsaugoti Atributai

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

4.3.1 Konstruktoriaus ir Destruktoriaus Dokumentacija

4.3.1.1 Zmogus() [1/4]

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

4.3.1.2 Zmogus() [2/4]

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

4.3.1.3 Zmogus() [3/4]

```
Zmogus::Zmogus (  
    const Zmogus & z) [inline]
```

4.3.1.4 Zmogus() [4/4]

```
Zmogus::Zmogus (  
    Zmogus && z) [inline], [noexcept]
```

4.3.1.5 ~Zmogus()

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

4.3.2 Metodų Dokumentacija

4.3.2.1 getPavarde()

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

Realizuota [Stud.](#)

4.3.2.2 getVardas()

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

Realizuota [Stud.](#)

4.3.2.3 operator=() [1/2]

```
Zmogus & Zmogus::operator= (  
    const Zmogus & z) [inline]
```

4.3.2.4 operator=() [2/2]

```
Zmogus & Zmogus::operator= (  
    Zmogus && z) [inline], [noexcept]
```

4.3.2.5 setPavarde()

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

4.3.2.6 setVardas()

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

4.3.3 Lauko aprašymas

4.3.3.1 pavarde

```
string Zmogus::pavarde [protected]
```

4.3.3.2 vardas

```
string Zmogus::vardas [protected]
```

Dokumentacija šiai klasei sugeneruota iš šio failo:

- [include/zmogus.h](#)

skyrius 5

Failo Dokumentacija

5.1 include/headers_main.h Failo Nuoroda

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <vector>
#include <iomanip>
#include <algorithm>
#include <chrono>
#include <type_traits>
#include <functional>
#include "../include/vector.h"
```

5.2 headers_main.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef HEADERS_MAIN_H
00002 #define HEADERS_MAIN_H
00003
00004 #include <iostream>
00005 #include <fstream>
00006 #include <sstream>
00007 #include <string>
00008 #include <vector>
00009 #include <iomanip>
00010 #include <algorithm>
00011 #include <chrono>
00012 #include <type_traits>
00013 #include <functional>
00014 #include "../include/vector.h"
00015
00016 using std::cin;
00017 using std::cout;
00018 using std::endl;
00019 using std::string;
00020 using std::vector;
00021
00022 #endif
```

5.3 include/isvesties_rezimai.h Failo Nuoroda

```
#include "headers_main.h"
#include "studentas.h"
```

Funkcijos

- void `isvestis` (std::ostream &out, `Vector< Stud >` &studentai)

5.3.1 Funkcijos Dokumentacija

5.3.1.1 isvestis()

```
void isvestis (
    std::ostream & out,
    Vector< Stud > & studentai)
```

5.4 isvesties_rezimai.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef ISVESTIES_REZIMAI_H
00002 #define ISVESTIES_REZIMAI_H
00003
00004 #include "headers_main.h"
00005 #include "studentas.h"
00006
00007 void isvestis (std::ostream &out, Vector<Stud> &studentai);
00008
00009 #endif
```

5.5 include/ivesties_rezimai.h Failo Nuoroda

```
#include "headers_main.h"
#include "studentas.h"
```

Funkcijos

- void `strategija_1` (`Vector< Stud >` &studentai, `Vector< Stud >` &vargsiukai, `Vector< Stud >` &galvociai)
- void `ranka` (`Vector< Stud >` &studentai)
- void `pazymiu_generavimas` (`Vector< Stud >` &studentai)
- void `visko_generavimas` (`Vector< Stud >` &studentai)
- void `skaitymas_is_failo` (`Vector< Stud >` &studentai, string failas, bool `testavimas`)
- void `failu_generavimas` (int n)
- void `studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius` (`Vector< Stud >` &studentai, string failas, std::chrono::duration< double > &ivesties_suma, std::chrono::duration< double > &rusiavimo_suma, std::chrono::duration< double > &skirstymo_suma, int strategija)

5.5.1 Funkcijos Dokumentacija

5.5.1.1 failu_generavimas()

```
void failu_generavimas (  
    int n)
```

5.5.1.2 pazymiu_generavimas()

```
void pazymiu_generavimas (  
    Vector< Stud > & studentai)
```

5.5.1.3 ranka()

```
void ranka (  
    Vector< Stud > & studentai)
```

5.5.1.4 skaitymas_is_failo()

```
void skaitymas_is_failo (  
    Vector< Stud > & studentai,  
    string failas,  
    bool testavimas)
```

5.5.1.5 strategija_1()

```
void strategija_1 (  
    Vector< Stud > & studentai,  
    Vector< Stud > & vargsiukai,  
    Vector< Stud > & galvociai)
```

5.5.1.6 studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius()

```
void studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius (  
    Vector< Stud > & studentai,  
    string failas,  
    std::chrono::duration< double > & ivesties_suma,  
    std::chrono::duration< double > & rusiavimo_suma,  
    std::chrono::duration< double > & skirstymo_suma,  
    int strategija)
```

5.5.1.7 visko_generavimas()

```
void visko_generavimas (  
    Vector< Stud > & studentai)
```

5.6 ivesties_rezimai.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef IVESTIES_REZIMAI_H
00002 #define IVESTIES_REZIMAI_H
00003
00004 #include "headers_main.h"
00005 #include "studentas.h"
00006
00007 void strategija_1(Vector<Stud> &studentai, Vector<Stud> &vargsiukai, Vector<Stud> &galvociai);
00008 void ranka (Vector<Stud> &studentai);
00009 void pazymiu_generavimas (Vector<Stud> &studentai);
00010 void visko_generavimas (Vector<Stud> &studentai);
00011 void skaitymas_is_failo (Vector<Stud> &studentai, string failas, bool testavimas);
00012 void failu_generavimas(int n);
00013 void studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius (Vector<Stud> &studentai, string
failas, std::chrono::duration<double> &ivesties_suma, std::chrono::duration<double> &rusiavimo_suma,
std::chrono::duration<double> &skirstymo_suma, int strategija/*, int pasirinkimas1, int pasirinkimas2,
int pasirinkimas3, int pasirinkimas4*/);
00014
00015 #endif
```

5.7 include/ivesties_tikrinimas.h Failo Nuoroda

```
#include "headers_main.h"
#include "studentas.h"
```

Funkcijos

- `template<typename T>`
`void ivesties_tikrinimas (T &reiksme, string &tekstas)`

5.7.1 Funkcijos Dokumentacija

5.7.1.1 ivesties_tikrinimas()

```
template<typename T>
void ivesties_tikrinimas (
    T & reiksme,
    string & tekstas)
```

5.8 ivesties_tikrinimas.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef IVESTIES_TIKRINIMAS_H
00002 #define IVESTIES_TIKRINIMAS_H
00003
00004 #include "headers_main.h"
00005 #include "studentas.h"
00006
00007 template <typename T>
00008 void ivesties_tikrinimas(T &reiksme, string &tekstas);
00009
00010 #endif
```


5.9 include/rikiavimo_funckija.h Failo Nuoroda

```
#include "headers_main.h"  
#include "studentas.h"
```

Funkcijos

- bool `raktas_v` (`Stud a`, `Stud b`)
- bool `raktas_p` (`Stud a`, `Stud b`)
- bool `raktas_gv` (`Stud a`, `Stud b`)
- bool `raktas_gm` (`Stud a`, `Stud b`)
- void `rikiavimas` (`Vector< Stud > &studentai`)

5.9.1 Funkcijos Dokumentacija

5.9.1.1 `raktas_gm()`

```
bool raktas_gm (  
    Stud a,  
    Stud b)
```

5.9.1.2 `raktas_gv()`

```
bool raktas_gv (  
    Stud a,  
    Stud b)
```

5.9.1.3 `raktas_p()`

```
bool raktas_p (  
    Stud a,  
    Stud b)
```

5.9.1.4 `raktas_v()`

```
bool raktas_v (  
    Stud a,  
    Stud b)
```

5.9.1.5 `rikiavimas()`

```
void rikiavimas (  
    Vector< Stud > & studentai)
```

5.10 rikiavimo_funckija.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef RIKIAVIMO_FUNKCIJA_H
00002 #define RIKIAVIMO_FUNKCIJA_H
00003
00004 #include "headers_main.h"
00005 #include "studentas.h"
00006
00007 bool raktas_v(Stud a, Stud b);
00008 bool raktas_p(Stud a, Stud b);
00009 bool raktas_gv(Stud a, Stud b);
00010 bool raktas_gm(Stud a, Stud b);
00011
00012 void rikiavimas (Vector<Stud> &studentai);
00013
00014 #endif
```

5.11 include/skaiciavimo_funkcijos.h Failo Nuoroda

```
#include "headers_main.h"
#include "studentas.h"
```

Funkcijos

- double [vidurkis](#) (Vector< int > &pazymiai, int &egzaminas)
- double [mediana](#) (Vector< int > &pazymiai, int &egzaminas)

5.11.1 Funkcijos Dokumentacija

5.11.1.1 mediana()

```
double mediana (
    Vector< int > & pazymiai,
    int & egzaminas)
```

5.11.1.2 vidurkis()

```
double vidurkis (
    Vector< int > & pazymiai,
    int & egzaminas)
```

5.12 skaiciavimo_funkcijos.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef SKAICIAVIMU_FUNKCIJOS_H
00002 #define SKAICIAVIMU_FUNKCIJOS_H
00003
00004 #include "headers_main.h"
00005 #include "studentas.h"
00006
00007 double vidurkis(Vector<int> &pazymiai, int &egzaminas);
00008 double mediana(Vector<int> &pazymiai, int &egzaminas);
00009
00010 #endif
```

5.13 include/studentas.h Failo Nuoroda

```
#include "headers_main.h"
#include "skaiciavimo_funkcijos.h"
#include "zmogus.h"
```

Duomenų struktūros

- class [Stud](#)

5.14 studentas.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef STUDENTAS_H
00002 #define STUDENTAS_H
00003
00004 #include "headers_main.h"
00005 #include "skaiciavimo_funkcijos.h"
00006 #include "zmogus.h"
00007
00008 class Stud : public Zmogus {
00009 private:
00010     Vector<int> pazymiai;
00011     int egzaminas;
00012     double galutinis_vid=-1;
00013     double galutinis_med=-1;
00014 public:
00015     Stud() : Zmogus("", "", egzaminas(0)) {}
00016     Stud(std::istream& is, string rezimas);
00017     inline string getVardas() const override { return vardas; }
00018     inline string getPavarde() const override { return pavarde; }
00019     inline double getGalutinisVid() const { return galutinis_vid; }
00020     inline double getGalutinisMed() const { return galutinis_med; }
00021     std::istream& readStudent(std::istream&, string rezimas);
00022     void skaiciuotiGalutini();
00023     ~Stud() {
00024         pazymiai.clear();
00025         //cout<<"Destruktorius\n";
00026     }
00027     Stud(const Stud &s) : Zmogus(s) {
00028         pazymiai = s.pazymiai;
00029         egzaminas = s.egzaminas;
00030         galutinis_vid = s.galutinis_vid;
00031         galutinis_med = s.galutinis_med;
00032     }
00033     Stud& operator=(const Stud &s) {
00034         if (this != &s) {
00035             vardas = s.vardas;
00036             pavarde = s.pavarde;
00037             pazymiai = s.pazymiai;
00038             egzaminas = s.egzaminas;
00039             galutinis_vid = s.galutinis_vid;
00040             galutinis_med = s.galutinis_med;
00041         }
00042         return *this;
00043     }
00044     Stud(Stud &&s) noexcept : Zmogus(std::move(s)) {
00045         pazymiai = std::move(s.pazymiai);
00046         egzaminas = s.egzaminas;
00047         galutinis_vid = s.galutinis_vid;
00048         galutinis_med = s.galutinis_med;
00049         s.egzaminas = 0;
00050         s.galutinis_vid = 0;
00051         s.galutinis_med = 0;
00052     }
00053     Stud& operator=(Stud &&s) noexcept {
00054         if (this != &s) {
00055             vardas = std::move(s.vardas);
00056             pavarde = std::move(s.pavarde);
00057             pazymiai = std::move(s.pazymiai);
00058             egzaminas = s.egzaminas;
00059             galutinis_vid = s.galutinis_vid;
```

```

00060         galutinis_med = s.galutinis_med;
00061         s.egzaminas = 0;
00062         s.galutinis_vid = 0;
00063         s.galutinis_med = 0;
00064     }
00065     return *this;
00066 }
00067
00068 friend std::ostream& operator<<(std::ostream& os, const Stud& s) {
00069     os << std::left<<std::setw(25)<< s.getVardas() <<std::setw(25)<< s.getPavarde() <<std::setw(25)<<
std::fixed << std::setprecision(2) << s.galutinis_vid << std::setw(25)<< s.galutinis_med;
00070     return os;
00071 }
00072
00073 friend std::istream& operator>>(std::istream& is, Stud& s) {
00074     s.pazymiai.clear();
00075     string v, p;
00076     is >> v >> p;
00077     s.setVardas(v);
00078     s.setPavarde(p);
00079
00080     int pazymys;
00081     bool isCin = (&is == &std::cin);
00082
00083     while (is >> pazymys) {
00084         if (isCin && pazymys == -1) break;
00085         s.pazymiai.push_back(pazymys);
00086     }
00087
00088     if (!isCin && !s.pazymiai.empty()) {
00089         s.egzaminas = s.pazymiai.back();
00090         s.pazymiai.pop_back();
00091     } else if (isCin) {
00092         is >> s.egzaminas;
00093     }
00094
00095     s.skaiciuotiGalutini();
00096     return is;
00097 }
00098
00099 };
00100
00101 #endif

```

5.15 include/studentas_testai.h Failo Nuoroda

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

Funkcijos

- bool testAll ()
- bool testEmptyConstructor ()
- bool testInputStreamConstructor ()
- bool testCopyConstructor ()
- bool testAssignmentOperator ()
- bool testMoveConstructor ()
- bool testMoveAssignmentOperator ()
- bool testInputOperator ()
- bool testOutputOperator ()
- bool testDestructor ()

5.15.1 Funkcijos Dokumentacija

5.15.1.1 testAll()

```
bool testAll ()
```

5.15.1.2 testAssignmentOperator()

```
bool testAssignmentOperator ()
```

5.15.1.3 testCopyConstructor()

```
bool testCopyConstructor ()
```

5.15.1.4 testDestructor()

```
bool testDestructor ()
```

5.15.1.5 testEmptyConstructor()

```
bool testEmptyConstructor ()
```

5.15.1.6 testInputOperator()

```
bool testInputOperator ()
```

5.15.1.7 testInputStreamConstructor()

```
bool testInputStreamConstructor ()
```

5.15.1.8 testMoveAssignmentOperator()

```
bool testMoveAssignmentOperator ()
```

5.15.1.9 testMoveConstructor()

```
bool testMoveConstructor ()
```

5.15.1.10 testOutputOperator()

```
bool testOutputOperator ()
```

5.16 studentas_testai.h

Eiti į šio failo dokumentaciją.

```
00001 #ifndef STUDENTAS_TESTAI_H
00002 #define STUDENTAS_TESTAI_H
00003
00004 #include "studentas.h"
00005
00006 bool testAll();
00007 bool testEmptyConstructor();
00008 bool testInputStreamConstructor();
00009 bool testCopyConstructor();
00010 bool testAssignmentOperator();
00011 bool testMoveConstructor();
00012 bool testMoveAssignmentOperator();
00013 bool testInputOperator();
00014 bool testOutputOperator();
00015 bool testDestructor();
00016
00017 #endif
```

5.17 include/testavimo_funkcijos.h Failo Nuoroda

```
#include "headers_main.h"
#include "studentas.h"
```

Funkcijos

- void [testinis_skaitymas_is_failo](#) (Vector< [Stud](#) > &studentai, string failas)
- void [testavimas](#) (Vector< [Stud](#) > &studentai)
- void [failo_kurimo_testavimas](#) ()
- void [duomenu_apdorojimo_testavimas](#) ()
- void [konteinerio_testavimas](#) ()
- void [perskirstymu_testas](#) ()

5.17.1 Funkcijos Dokumentacija

5.17.1.1 duomenu_apdorojimo_testavimas()

```
void duomenu_apdorojimo_testavimas ()
```

5.17.1.2 failo_kurimo_testavimas()

```
void failo_kurimo_testavimas ()
```

5.17.1.3 konteinerio_testavimas()

```
void konteinerio_testavimas ()
```

5.17.1.4 perskirstymu_testas()

```
void perskirstymu_testas ()
```

5.17.1.5 testavimas()

```
void testavimas (  
    Vector< Stud > & studentai)
```

5.17.1.6 testinis_skaitymas_is_failo()

```
void testinis_skaitymas_is_failo (  
    Vector< Stud > & studentai,  
    string failas)
```

5.18 testavimo_funkcijos.h

Eiti į šio failo dokumentaciją.

```
00001 #ifndef TESTAVIMO_FUNKCIJOS_H  
00002 #define TESTAVIMO_FUNKCIJOS_H  
00003  
00004 #include "headers_main.h"  
00005 #include "studentas.h"  
00006  
00007 void testinis_skaitymas_is_failo (Vector<Stud> &studentai, string failas);  
00008 void testavimas(Vector<Stud> &studentai);  
00009 void failo_kurimo_testavimas ();  
00010 void duomenu_apdorojimo_testavimas();  
00011 void konteinerio_testavimas();  
00012 void perskirstymu_testas();  
00013  
00014 #endif
```

5.19 include/vartotojo_pasirinkimo_funkcijos.h Failo Nuoroda

```
#include "headers_main.h"  
#include "studentas.h"
```

Funkcijos

- void [rezimo_pasirinkimas](#) (Vector< Stud > &studentai)
- void [isvesties_pasirinkimas](#) (Vector< Stud > &studentai)
- void [failo_pasirinkimas](#) (Vector< Stud > &studentai)

5.19.1 Funkcijos Dokumentacija

5.19.1.1 failo_pasirinkimas()

```
void failo_pasirinkimas (  
    Vector< Stud > & studentai)
```

5.19.1.2 isvesties_pasirinkimas()

```
void isvesties_pasirinkimas (
    Vector< Stud > & studentai)
```

5.19.1.3 rezimo_pasirinkimas()

```
void rezimo_pasirinkimas (
    Vector< Stud > & studentai)
```

5.20 vartotojo_pasirinkimo_funkcijos.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef VARTOTOJO_PASIRINKIMO_FUNKCIJOS_H
00002 #define VARTOTOJO_PASIRINKIMO_FUNKCIJOS_H
00003
00004 #include "headers_main.h"
00005 #include "studentas.h"
00006
00007 void rezimo_pasirinkimas (Vector<Stud> &studentai);
00008 void isvesties_pasirinkimas (Vector<Stud> &studentai);
00009 void failo_pasirinkimas (Vector<Stud> &studentai);
00010
00011 #endif
```

5.21 include/vector.h Failo Nuoroda

```
#include <iostream>
#include <stdexcept>
#include <limits>
#include "../source/vector.tpp"
```

Duomenų struktūros

- class `Vector< T >`

5.22 vector.h

[Eiti į šio failo dokumentaciją.](#)

```
00001 #ifndef VECTOR_H
00002 #define VECTOR_H
00003
00004 #include <iostream>
00005 #include <stdexcept>
00006 #include <limits>
00007
00008 template <typename T>
00009 class Vector {
00010 private:
00011     T* data;
00012     size_t size;
00013     size_t capacity;
00014 public:
00015     Vector(size_t init_capacity=10);
```



```

00016     ~Vector();
00017     Vector(const Vector& other);
00018     Vector& operator= (const Vector& other);
00019     Vector(Vector&& other) noexcept;
00020     Vector& operator= (Vector&& other) noexcept;
00021     void assign(size_t count, const T& value);
00022     void assign(const T* first, const T* last);
00023     void assign(const Vector& other);
00024     T& at(size_t index);
00025     T& operator[] (size_t index);
00026     T& front();
00027     T& back();
00028     T* getData();
00029     T* begin();
00030     T* end();
00031     T* rbegin();
00032     T* rend();
00033     bool empty() const;
00034     size_t getSize() const;
00035     size_t max_size() const;
00036     void reserve(size_t new_capacity);
00037     size_t getCapacity() const;
00038     void shrink_to_fit();
00039     void clear();
00040     void insert(size_t index, const T& value);
00041     void erase(size_t index);
00042     void push_back(const T& value);
00043     template <typename Range>
00044     void append_range(const Range& other);
00045     void pop_back();
00046     void resize(size_t new_size);
00047     void swap(Vector& other);
00048     bool operator== (const Vector& other) const;
00049     bool operator!= (const Vector& other) const;
00050     bool operator< (const Vector& other) const;
00051     bool operator<= (const Vector& other) const;
00052     bool operator> (const Vector& other) const;
00053     bool operator>= (const Vector& other) const;
00054
00055
00056 };
00057
00058 #include "../source/vector.tpp"
00059
00060 #endif

```

5.23 include/zmogus.h Failo Nuoroda

```
#include "headers_main.h"
```

Duomenų struktūros

- class [Zmogus](#)

5.24 zmogus.h

Eiti į šio failo dokumentaciją.

```

00001 #ifndef ZMOGUS_H
00002 #define ZMOGUS_H
00003
00004 #include "headers_main.h"
00005
00006 class Zmogus {
00007 protected:
00008     string vardas;
00009     string pavarde;
00010 public:
00011     Zmogus(string v, string p) : vardas(v), pavarde(p) {}
00012     Zmogus() : vardas(""), pavarde("") {}

```

```

00013
00014     Zmogus(const Zmogus &z) : vardas(z.vardas), pavarde(z.pavarde) {}
00015     Zmogus& operator=(const Zmogus &z) {
00016         if (this != &z) {
00017             vardas = z.vardas;
00018             pavarde = z.pavarde;
00019         }
00020         return *this;
00021     }
00022     Zmogus(Zmogus &&z) noexcept : vardas(std::move(z.vardas)), pavarde(std::move(z.pavarde)) {}
00023     Zmogus& operator=(Zmogus &&z) noexcept {
00024         if (this != &z) {
00025             vardas = std::move(z.vardas);
00026             pavarde = std::move(z.pavarde);
00027         }
00028         return *this;
00029     }
00030
00031     virtual ~Zmogus() {
00032         vardas.clear();
00033         pavarde.clear();
00034     }
00035     virtual string getVardas() const = 0;
00036     virtual string getPavarde() const = 0;
00037
00038     void setVardas(const string& v) { vardas = v; }
00039     void setPavarde(const string& p) { pavarde = p; }
00040 };
00041
00042 #endif

```

5.25 source/isvesties_rezimai.cpp Failo Nuoroda

```
#include "../include/isvesties_rezimai.h"
```

Funkcijos

- void `isvestis` (std::ostream &out, `Vector< Stud >` &studentai)

5.25.1 Funkcijos Dokumentacija

5.25.1.1 isvestis()

```

void isvestis (
    std::ostream & out,
    Vector< Stud > & studentai)

```

5.26 source/ivesties_rezimai.cpp Failo Nuoroda

```

#include "../include/ivesties_rezimai.h"
#include "../include/ivesties_tikrinimas.h"
#include "../include/skaiciavimo_funkcijos.h"
#include "../include/vartotojo_pasirinkimo_funkcijos.h"
#include "../include/isvesties_rezimai.h"
#include "../include/rikiavimo_funckija.h"

```

Funkcijos

- void `strategija_1` (`Vector< Stud > &studentai`, `Vector< Stud > &vargsiukai`, `Vector< Stud > &galvociai`)
- void `ranka` (`Vector< Stud > &studentai`)
- void `pazymiu_generavimas` (`Vector< Stud > &studentai`)
- void `visko_generavimas` (`Vector< Stud > &studentai`)
- void `skaitymas_is_failo` (`Vector< Stud > &studentai`, string `failas`, bool `testavimas`)
- void `failu_generavimas` (int `n`)
- void `studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius` (`Vector< Stud > &studentai`, string `failas`, `std::chrono::duration< double > &ivesties_suma`, `std::chrono::duration< double > &rusiavimo_suma`, `std::chrono::duration< double > &skirstymo_suma`, int `strategija`)

5.26.1 Funkcijos Dokumentacija

5.26.1.1 failu_generavimas()

```
void failu_generavimas (  
    int n)
```

5.26.1.2 pazymiu_generavimas()

```
void pazymiu_generavimas (  
    Vector< Stud > & studentai)
```

5.26.1.3 ranka()

```
void ranka (  
    Vector< Stud > & studentai)
```

5.26.1.4 skaitymas_is_failo()

```
void skaitymas_is_failo (  
    Vector< Stud > & studentai,  
    string failas,  
    bool testavimas)
```

5.26.1.5 strategija_1()

```
void strategija_1 (  
    Vector< Stud > & studentai,  
    Vector< Stud > & vargsiukai,  
    Vector< Stud > & galvociai)
```

5.26.1.6 studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius()

```
void studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius (
    Vector< Stud > & studentai,
    string failas,
    std::chrono::duration< double > & ivesties_suma,
    std::chrono::duration< double > & rusiavimo_suma,
    std::chrono::duration< double > & skirstymo_suma,
    int strategija)
```

5.26.1.7 visko_generavimas()

```
void visko_generavimas (
    Vector< Stud > & studentai)
```

5.27 source/ivesties_tikrinimas.cpp Failo Nuoroda

```
#include "../include/ivesties_tikrinimas.h"
```

Funkcijos

- template<typename T>
void [ivesties_tikrinimas](#) (T &reiksme, string &tekstas)
- template void [ivesties_tikrinimas< int >](#) (int &, string &)
- template void [ivesties_tikrinimas< string >](#) (string &, string &)

5.27.1 Funkcijos Dokumentacija

5.27.1.1 ivesties_tikrinimas()

```
template<typename T>
void ivesties_tikrinimas (
    T & reiksme,
    string & tekstas)
```

5.27.1.2 ivesties_tikrinimas< int >()

```
template void ivesties_tikrinimas< int > (
    int & ,
    string & )
```

5.27.1.3 ivesties_tikrinimas< string >()

```
template void ivesties_tikrinimas< string > (
    string & ,
    string & )
```

5.28 source/main.cpp Failo Nuoroda

```
#include "../include/headers_main.h"
#include "../include/ivesties_rezimai.h"
#include "../include/ivesties_tikrinimas.h"
#include "../include/isvesties_rezimai.h"
#include "../include/testavimo_funkcijos.h"
#include "../include/vartotojo_pasirinkimo_funkcijos.h"
```

Funkcijos

- int `main` ()

5.28.1 Funkcijos Dokumentacija

5.28.1.1 `main()`

```
int main ()
```

5.29 source/rikiavimo_funkcija.cpp Failo Nuoroda

```
#include "../include/rikiavimo_funckija.h"
#include "../include/ivesties_tikrinimas.h"
```

Funkcijos

- bool `raktas_v` (`Stud` a, `Stud` b)
- bool `raktas_p` (`Stud` a, `Stud` b)
- bool `raktas_gv` (`Stud` a, `Stud` b)
- bool `raktas_gm` (`Stud` a, `Stud` b)
- void `rikiavimas` (`Vector`< `Stud` > &studentai)

5.29.1 Funkcijos Dokumentacija

5.29.1.1 `raktas_gm()`

```
bool raktas_gm (
    Stud a,
    Stud b)
```

5.29.1.2 `raktas_gv()`

```
bool raktas_gv (
    Stud a,
    Stud b)
```

5.29.1.3 raktas_p()

```
bool raktas_p (
    Stud a,
    Stud b)
```

5.29.1.4 raktas_v()

```
bool raktas_v (
    Stud a,
    Stud b)
```

5.29.1.5 rikiavimas()

```
void rikiavimas (
    Vector< Stud > & studentai)
```

5.30 source/skaiciavimo_funkcijos.cpp Failo Nuoroda

```
#include "../include/skaiciavimo_funkcijos.h"
```

Funkcijos

- double vidurkis (Vector< int > &pazymiai, int &egzaminas)
- double mediana (Vector< int > &pazymiai, int &egzaminas)

5.30.1 Funkcijos Dokumentacija

5.30.1.1 mediana()

```
double mediana (
    Vector< int > & pazymiai,
    int & egzaminas)
```

5.30.1.2 vidurkis()

```
double vidurkis (
    Vector< int > & pazymiai,
    int & egzaminas)
```

5.31 source/studentas.cpp Failo Nuoroda

```
#include "../include/studentas.h"  
#include "../include/ivesties_tikrinimas.h"  
#include "../include/skaiciavimo_funkcijos.h"
```

5.32 source/studentas_testai.cpp Failo Nuoroda

```
#include "../include/studentas_testai.h"
```

Funkcijos

- bool [testAll](#) ()
- bool [testEmptyConstructor](#) ()
- bool [testInputStreamConstructor](#) ()
- bool [testCopyConstructor](#) ()
- bool [testAssignmentOperator](#) ()
- bool [testMoveConstructor](#) ()
- bool [testMoveAssignmentOperator](#) ()
- bool [testInputOperator](#) ()
- bool [testOutputOperator](#) ()
- bool [testDestructor](#) ()

5.32.1 Funkcijos Dokumentacija

5.32.1.1 testAll()

```
bool testAll ()
```

5.32.1.2 testAssignmentOperator()

```
bool testAssignmentOperator ()
```

5.32.1.3 testCopyConstructor()

```
bool testCopyConstructor ()
```

5.32.1.4 testDestructor()

```
bool testDestructor ()
```

5.32.1.5 testEmptyConstructor()

```
bool testEmptyConstructor ()
```

5.32.1.6 testInputOperator()

```
bool testInputOperator ()
```

5.32.1.7 testInputStreamConstructor()

```
bool testInputStreamConstructor ()
```

5.32.1.8 testMoveAssignmentOperator()

```
bool testMoveAssignmentOperator ()
```

5.32.1.9 testMoveConstructor()

```
bool testMoveConstructor ()
```

5.32.1.10 testOutputOperator()

```
bool testOutputOperator ()
```

5.33 source/testavimo_funkcijos.cpp Failo Nuoroda

```
#include "../include/testavimo_funkcijos.h"  
#include "../include/ivesties_tikrinimas.h"  
#include "../include/skaiciavimo_funkcijos.h"  
#include "../include/ivesties_rezimai.h"
```

Funkcijos

- void [testinis_skaitymas_is_failo](#) (Vector< Stud > &studentai, string failas)
- void [testavimas](#) (Vector< Stud > &studentai)
- void [failo_kurimo_testavimas](#) ()
- void [duomenu_apdorojimo_testavimas](#) ()
- void [konteinerio_testavimas](#) ()
- void [perskirstymu_testas](#) ()

5.33.1 Funkcijos Dokumentacija

5.33.1.1 duomenu_apdorojimo_testavimas()

```
void duomenu_apdorojimo_testavimas ()
```

5.33.1.2 failo_kurimo_testavimas()

```
void failo_kurimo_testavimas ()
```

5.33.1.3 konteinerio_testavimas()

```
void konteinerio_testavimas ()
```

5.33.1.4 perskirstymu_testas()

```
void perskirstymu_testas ()
```

5.33.1.5 testavimas()

```
void testavimas (  
    Vector< Stud > & studentai)
```

5.33.1.6 testinis_skaitymas_is_failo()

```
void testinis_skaitymas_is_failo (  
    Vector< Stud > & studentai,  
    string failas)
```

5.34 source/vartotojo_pasirinkimo_funkcijos.cpp Failo Nuoroda

```
#include "../include/vartotojo_pasirinkimo_funkcijos.h"  
#include "../include/isvesties_rezimai.h"  
#include "../include/ivesties_rezimai.h"  
#include "../include/ivesties_tikrinimas.h"  
#include "../include/rikiavimo_funckija.h"  
#include "../include/testavimo_funkcijos.h"  
#include "../include/studentas_testai.h"
```

Funkcijos

- void [rezimo_pasirinkimas](#) (Vector< Stud > &studentai)
- void [isvesties_pasirinkimas](#) (Vector< Stud > &studentai)
- void [failo_pasirinkimas](#) (Vector< Stud > &studentai)

5.34.1 Funkcijos Dokumentacija

5.34.1.1 failo_pasirinkimas()

```
void failo_pasirinkimas (  
    Vector< Stud > & studentai)
```

5.34.1.2 isvesties_pasirinkimas()

```
void isvesties_pasirinkimas (  
    Vector< Stud > & studentai)
```

5.34.1.3 rezimo_pasirinkimas()

```
void rezimo_pasirinkimas (  
    Vector< Stud > & studentai)
```

Rodyklė

~Stud
 Stud, [8](#)
~Vector
 Vector< T >, [11](#)
~Zmogus
 Zmogus, [18](#)

append_range
 Vector< T >, [12](#)
assign
 Vector< T >, [12](#)
at
 Vector< T >, [12](#)

back
 Vector< T >, [13](#)
begin
 Vector< T >, [13](#)

capacity
 Vector< T >, [16](#)
clear
 Vector< T >, [13](#)

data
 Vector< T >, [16](#)
duomenų_apdorojimo_testavimas
 testavimo_funkcijos.cpp, [41](#)
 testavimo_funkcijos.h, [30](#)

egzaminas
 Stud, [10](#)
empty
 Vector< T >, [13](#)
end
 Vector< T >, [13](#)
erase
 Vector< T >, [13](#)

failo_kurimo_testavimas
 testavimo_funkcijos.cpp, [41](#)
 testavimo_funkcijos.h, [30](#)
failo_pasirinkimas
 vartotojo_pasirinkimo_funkcijos.cpp, [42](#)
 vartotojo_pasirinkimo_funkcijos.h, [31](#)
failu_generavimas
 investis_rezimai.cpp, [35](#)
 investis_rezimai.h, [23](#)
front
 Vector< T >, [13](#)

galutinis_med
 Stud, [10](#)
galutinis_vid
 Stud, [10](#)
getCapacity
 Vector< T >, [13](#)
getData
 Vector< T >, [14](#)
getGalutinisMed
 Stud, [9](#)
getGalutinisVid
 Stud, [9](#)
getPavarde
 Stud, [9](#)
 Zmogus, [18](#)
getSize
 Vector< T >, [14](#)
getVardas
 Stud, [9](#)
 Zmogus, [18](#)

include/headers_main.h, [21](#)
include/isvesties_rezimai.h, [22](#)
include/ivesties_rezimai.h, [22](#), [24](#)
include/ivesties_tikrinimas.h, [24](#)
include/rikiavimo_funckija.h, [25](#), [26](#)
include/skaiciavimo_funkcijos.h, [26](#)
include/studentas.h, [27](#)
include/studentas_testai.h, [28](#), [30](#)
include/testavimo_funkcijos.h, [30](#), [31](#)
include/vartotojo_pasirinkimo_funkcijos.h, [31](#), [32](#)
include/vector.h, [32](#)
include/zmogus.h, [33](#)
insert
 Vector< T >, [14](#)
isvesties_pasirinkimas
 vartotojo_pasirinkimo_funkcijos.cpp, [42](#)
 vartotojo_pasirinkimo_funkcijos.h, [31](#)
isvesties_rezimai.cpp
 isvestis, [34](#)
isvesties_rezimai.h
 isvestis, [22](#)
isvestis
 investis_rezimai.cpp, [34](#)
 investis_rezimai.h, [22](#)
investis_rezimai.cpp
 failu_generavimas, [35](#)
 pazymiu_generavimas, [35](#)
 ranka, [35](#)

skaitymas_is_failo, 35
 strategija_1, 35
 studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius, 35
 visko_generavimas, 36
 ivesties_rezimai.h
 failu_generavimas, 23
 pazymiu_generavimas, 23
 ranka, 23
 skaitymas_is_failo, 23
 strategija_1, 23
 studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius, 23
 visko_generavimas, 23
 ivesties_tikrinimas
 ivesties_tikrinimas.cpp, 36
 ivesties_tikrinimas.h, 24
 ivesties_tikrinimas< int >
 ivesties_tikrinimas.cpp, 36
 ivesties_tikrinimas< string >
 ivesties_tikrinimas.cpp, 36
 ivesties_tikrinimas.cpp
 ivesties_tikrinimas, 36
 ivesties_tikrinimas< int >, 36
 ivesties_tikrinimas< string >, 36
 ivesties_tikrinimas.h
 ivesties_tikrinimas, 24
 konteinerio_testavimas
 testavimo_funkcijos.cpp, 41
 testavimo_funkcijos.h, 30
 main
 main.cpp, 37
 main.cpp
 main, 37
 max_size
 Vector< T >, 14
 mediana
 skaiciavimo_funkcijos.cpp, 38
 skaiciavimo_funkcijos.h, 26
 operator!=
 Vector< T >, 14
 operator<
 Vector< T >, 14
 operator<<
 Stud, 10
 operator<=
 Vector< T >, 14
 operator>
 Vector< T >, 15
 operator>>
 Stud, 10
 operator>=
 Vector< T >, 15
 operator=
 Stud, 9
 Vector< T >, 14, 15
 Zmogus, 18
 operator==
 Vector< T >, 15
 operator[]
 Vector< T >, 15
 pavarde
 Zmogus, 19
 pazymiai
 Stud, 10
 pazymiu_generavimas
 ivesties_rezimai.cpp, 35
 ivesties_rezimai.h, 23
 perskirstymu_testas
 testavimo_funkcijos.cpp, 41
 testavimo_funkcijos.h, 30
 pop_back
 Vector< T >, 15
 push_back
 Vector< T >, 15
 raktas_gm
 rikiavimo_funkcija.h, 25
 rikiavimo_funkcija.cpp, 37
 raktas_gv
 rikiavimo_funkcija.h, 25
 rikiavimo_funkcija.cpp, 37
 raktas_p
 rikiavimo_funkcija.h, 25
 rikiavimo_funkcija.cpp, 37
 raktas_v
 rikiavimo_funkcija.h, 25
 rikiavimo_funkcija.cpp, 38
 ranka
 ivesties_rezimai.cpp, 35
 ivesties_rezimai.h, 23
 rbegin
 Vector< T >, 15
 readStudent
 Stud, 9
 rend
 Vector< T >, 16
 reserve
 Vector< T >, 16
 resize
 Vector< T >, 16
 rezimo_pasirinkimas
 vartotojo_pasirinkimo_funkcijos.cpp, 42
 vartotojo_pasirinkimo_funkcijos.h, 32
 rikiavimas
 rikiavimo_funkcija.h, 25
 rikiavimo_funkcija.cpp, 38
 rikiavimo_funkcija.h
 raktas_gm, 25
 raktas_gv, 25
 raktas_p, 25
 raktas_v, 25
 rikiavimas, 25
 rikiavimo_funkcija.cpp

- raktas_gm, 37
- raktas_gv, 37
- raktas_p, 37
- raktas_v, 38
- rikiavimas, 38
- setPavarde
 - Zmogus, 18
- setVardas
 - Zmogus, 19
- shrink_to_fit
 - Vector< T >, 16
- size
 - Vector< T >, 16
- skaiciavimo_funkcijos.cpp
 - mediana, 38
 - vidurkis, 38
- skaiciavimo_funkcijos.h
 - mediana, 26
 - vidurkis, 26
- skaiciuotiGalutini
 - Stud, 9
- skaitymas_is_failo
 - investies_rezimai.cpp, 35
 - investies_rezimai.h, 23
- source/investies_rezimai.cpp, 34
- source/investies_rezimai.cpp, 34
- source/investies_tikrinimas.cpp, 36
- source/main.cpp, 37
- source/rikiavimo_funkcija.cpp, 37
- source/skaiciavimo_funkcijos.cpp, 38
- source/studentas.cpp, 39
- source/studentas_testai.cpp, 39
- source/testavimo_funkcijos.cpp, 40
- source/vartotojo_pasirinkimo_funkcijos.cpp, 41
- strategija_1
 - investies_rezimai.cpp, 35
 - investies_rezimai.h, 23
- Stud, 7
 - ~Stud, 8
 - egzaminas, 10
 - galutinis_med, 10
 - galutinis_vid, 10
 - getGalutinisMed, 9
 - getGalutinisVid, 9
 - getPavarde, 9
 - getVardas, 9
 - operator<<, 10
 - operator>>, 10
 - operator=, 9
 - pazymiai, 10
 - readStudent, 9
 - skaiciuotiGalutini, 9
 - Stud, 8
- studentas_testai.cpp
 - testAll, 39
 - testAssignmentOperator, 39
 - testCopyConstructor, 39
 - testDestructor, 39
- testEmptyConstructor, 39
- testInputOperator, 40
- testInputStreamConstructor, 40
- testMoveAssignmentOperator, 40
- testMoveConstructor, 40
- testOutputOperator, 40
- studentas_testai.h
 - testAll, 28
 - testAssignmentOperator, 28
 - testCopyConstructor, 29
 - testDestructor, 29
 - testEmptyConstructor, 29
 - testInputOperator, 29
 - testInputStreamConstructor, 29
 - testMoveAssignmentOperator, 29
 - testMoveConstructor, 29
 - testOutputOperator, 29
- studentu_skaitymas_ir_skirstymas_i_vargsiukus_ir_galvocius
 - investies_rezimai.cpp, 35
 - investies_rezimai.h, 23
- swap
 - Vector< T >, 16
- testAll
 - studentas_testai.cpp, 39
 - studentas_testai.h, 28
- testAssignmentOperator
 - studentas_testai.cpp, 39
 - studentas_testai.h, 28
- testavimas
 - testavimo_funkcijos.cpp, 41
 - testavimo_funkcijos.h, 31
- testavimo_funkcijos.cpp
 - duomenu_apdorojimo_testavimas, 41
 - failo_kurimo_testavimas, 41
 - konteinerio_testavimas, 41
 - perskirstymu_testas, 41
 - testavimas, 41
 - testinis_skaitymas_is_failo, 41
- testavimo_funkcijos.h
 - duomenu_apdorojimo_testavimas, 30
 - failo_kurimo_testavimas, 30
 - konteinerio_testavimas, 30
 - perskirstymu_testas, 30
 - testavimas, 31
 - testinis_skaitymas_is_failo, 31
- testCopyConstructor
 - studentas_testai.cpp, 39
 - studentas_testai.h, 29
- testDestructor
 - studentas_testai.cpp, 39
 - studentas_testai.h, 29
- testEmptyConstructor
 - studentas_testai.cpp, 39
 - studentas_testai.h, 29
- testinis_skaitymas_is_failo
 - testavimo_funkcijos.cpp, 41
 - testavimo_funkcijos.h, 31
- testInputOperator

- studentas_testai.cpp, 40
 - studentas_testai.h, 29
- testInputStreamConstructor
 - studentas_testai.cpp, 40
 - studentas_testai.h, 29
- testMoveAssignmentOperator
 - studentas_testai.cpp, 40
 - studentas_testai.h, 29
- testMoveConstructor
 - studentas_testai.cpp, 40
 - studentas_testai.h, 29
- testOutputOperator
 - studentas_testai.cpp, 40
 - studentas_testai.h, 29
- vardas
 - Zmogus, 19
- virtotojo_pasirinkimo_funkcijos.cpp
 - failo_pasirinkimas, 42
 - isvesties_pasirinkimas, 42
 - rezimo_pasirinkimas, 42
- virtotojo_pasirinkimo_funkcijos.h
 - failo_pasirinkimas, 31
 - isvesties_pasirinkimas, 31
 - rezimo_pasirinkimas, 32
- Vector
 - Vector< T >, 11, 12
- Vector< T >, 10
 - ~Vector, 11
 - append_range, 12
 - assign, 12
 - at, 12
 - back, 13
 - begin, 13
 - capacity, 16
 - clear, 13
 - data, 16
 - empty, 13
 - end, 13
 - erase, 13
 - front, 13
 - getCapacity, 13
 - getData, 14
 - getSize, 14
 - insert, 14
 - max_size, 14
 - operator!=, 14
 - operator<, 14
 - operator<=, 14
 - operator>, 15
 - operator>=, 15
 - operator=, 14, 15
 - operator==, 15
 - operator[], 15
 - pop_back, 15
 - push_back, 15
 - rbegin, 15
 - rend, 16
 - reserve, 16
 - resize, 16
 - shrink_to_fit, 16
 - size, 16
 - swap, 16
 - Vector, 11, 12
- vidurkis
 - skaiciavimo_funkcijos.cpp, 38
 - skaiciavimo_funkcijos.h, 26
- visko_generavimas
 - ivesties_rezimai.cpp, 36
 - ivesties_rezimai.h, 23
- Zmogus, 17
 - ~Zmogus, 18
 - getPavarde, 18
 - getVardas, 18
 - operator=, 18
 - pavarde, 19
 - setPavarde, 18
 - setVardas, 19
 - vardas, 19
 - Zmogus, 17, 18