

V2.0

Alpha

Generated by Doxygen 1.10.0

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Human Class Reference	7
4.1.1 Detailed Description	8
4.1.2 Member Function Documentation	8
4.1.2.1 addNamuDarbas()	8
4.1.2.2 atsitiktiniai()	8
4.1.2.3 atsitiktiniaiStudentai()	8
4.1.2.4 getEgzaminas()	9
4.1.2.5 getNamuDarbai()	9
4.1.2.6 getPavarde()	9
4.1.2.7 getVardas()	9
4.1.2.8 setEgzaminas()	9
4.1.2.9 setNamuDarbai()	10
4.1.2.10 setPavarde()	10
4.1.2.11 setVardas()	10
4.1.2.12 skaiciuotiGalutini()	11
4.1.2.13 skaiciuotiMediana()	11
4.1.2.14 skaiciuotiVidurki()	11
4.2 Studentas Class Reference	12
4.2.1 Detailed Description	13
4.2.2 Constructor & Destructor Documentation	13
4.2.2.1 Studentas() [1/3]	13
4.2.2.2 Studentas() [2/3]	13
4.2.2.3 Studentas() [3/3]	14
4.2.3 Member Function Documentation	14
4.2.3.1 addNamuDarbas()	14
4.2.3.2 atsitiktiniai()	14
4.2.3.3 atsitiktiniaiStudentai()	14
4.2.3.4 getEgzaminas()	15
4.2.3.5 getNamuDarbai()	15
4.2.3.6 getPavarde()	15
4.2.3.7 getVardas()	15
4.2.3.8 operator=() [1/2]	15
4.2.3.9 operator=() [2/2]	16

4.2.3.10 setEgzaminas()	16
4.2.3.11 setNamuDarbai()	16
4.2.3.12 setPavarde()	17
4.2.3.13 setVardas()	17
4.2.3.14 skaiciuotiGalutini()	17
4.2.3.15 skaiciuotiMediana()	18
4.2.3.16 skaiciuotiVidurki()	18
4.2.4 Friends And Related Symbol Documentation	18
4.2.4.1 operator<<	18
4.2.4.2 operator>>	18
5 File Documentation	21
5.1 functions.h	21
5.2 functions_old.h	21
5.3 functions_vector.h	22
5.4 studentas.h	22
5.5 zmogus.h	23
Index	25

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Human	7
Studentas	12

Chapter 2

Class Index

2.1 Class List

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

Human	Klasė Žmogus atstovauja žmogų	7
Studentas	Klasė Studentas atstovauja studentą, paveldintį iš klasės Žmogus	12

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

functions.h	21
functions_old.h	21
functions_vector.h	22
studentas.h	22
zmogus.h	23

Chapter 4

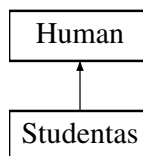
Class Documentation

4.1 Human Class Reference

Klasė Žmogus atstovauja žmogų.

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

Inheritance diagram for Human:



Public Member Functions

- virtual `~Human()`=default
Žmogaus klasės destruktorius.
- virtual void `setVardas` (const std::string &vardas)=0
Nustato žmogaus vardą.
- virtual std::string `getVardas` () const =0
Gražina žmogaus vardą.
- virtual void `setPavarde` (const std::string &pavarde)=0
Nustato žmogaus pavardę.
- virtual std::string `getPavarde` () const =0
Gražina žmogaus pavardę.
- virtual void `setNamuDarbai` (const std::vector< int > &nd)=0
Nustato žmogaus namų darbų rezultatus.
- virtual std::vector< int > `getNamuDarbai` () const =0
Gražina žmogaus namų darbų rezultatus.
- virtual void `addNamuDarbas` (int pazymys)=0
Prideda naują namų darbą prie žmogaus namų darbų sąrašo.
- virtual void `setEgzaminas` (int egzaminas)=0
Nustato žmogaus egzamino rezultatą.
- virtual int `getEgzaminas` () const =0

- *Grąžina žmogaus egzamino rezultatą.*
virtual double [skaiciuotiVidurki](#) () const =0
Skaičiuoja žmogaus vidurkį.
- virtual double [skaiciuotiMediana](#) () const =0
Skaičiuoja žmogaus medianą.
- virtual double [skaiciuotiGalutini](#) (bool naudotiVidurki) const =0
Skaičiuoja galutinį žmogaus rezultatą.
- virtual void [atsitiktiniai](#) ()=0
Generuoja atsitiktinius žmogaus rezultatus.
- virtual void [atsitiktiniaiStudentai](#) ()=0
Generuoja atsitiktinius rezultatus žmonių sąrašė.

4.1.1 Detailed Description

Klasė Žmogus atstovauja žmogų.

4.1.2 Member Function Documentation

4.1.2.1 addNamuDarbas()

```
virtual void Human::addNamuDarbas (
    int pazymys ) [pure virtual]
```

Prideda naują namų darbą prie žmogaus namų darbų sąrašo.

Parameters

<i>pazymys</i>	Pridedamas namų darbo rezultatas.
----------------	-----------------------------------

Implemented in [Studentas](#).

4.1.2.2 atsitiktiniai()

```
virtual void Human::atsitiktiniai ( ) [pure virtual]
```

Generuoja atsitiktinius žmogaus rezultatus.

Implemented in [Studentas](#).

4.1.2.3 atsitiktiniaiStudentai()

```
virtual void Human::atsitiktiniaiStudentai ( ) [pure virtual]
```

Generuoja atsitiktinius rezultatus žmonių sąrašė.

Implemented in [Studentas](#).

4.1.2.4 getEgzaminas()

```
virtual int Human::getEgzaminas ( ) const [pure virtual]
```

Grąžina žmogaus egzamino rezultatą.

Returns

Žmogaus egzamino rezultatas.

Implemented in [Studentas](#).

4.1.2.5 getNamuDarbai()

```
virtual std::vector< int > Human::getNamuDarbai ( ) const [pure virtual]
```

Grąžina žmogaus namų darbų rezultatus.

Returns

Žmogaus namų darbų rezultatų sąrašas.

Implemented in [Studentas](#).

4.1.2.6 getPavarde()

```
virtual std::string Human::getPavarde ( ) const [pure virtual]
```

Grąžina žmogaus pavardę.

Returns

Žmogaus pavardė.

Implemented in [Studentas](#).

4.1.2.7 getVardas()

```
virtual std::string Human::getVardas ( ) const [pure virtual]
```

Grąžina žmogaus vardą.

Returns

Žmogaus vardas.

Implemented in [Studentas](#).

4.1.2.8 setEgzaminas()

```
virtual void Human::setEgzaminas (
    int egzaminas ) [pure virtual]
```

Nustato žmogaus egzamino rezultatą.

Parameters

<i>egzaminas</i>	Žmogaus egzamino rezultatas.
------------------	------------------------------

Implemented in [Studentas](#).

4.1.2.9 setNamuDarbai()

```
virtual void Human::setNamuDarbai (
    const std::vector< int > & nd ) [pure virtual]
```

Nustato žmogaus namų darbų rezultatus.

Parameters

<i>nd</i>	Žmogaus namų darbų rezultatų sąrašas.
-----------	---------------------------------------

Implemented in [Studentas](#).

4.1.2.10 setPavarde()

```
virtual void Human::setPavarde (
    const std::string & pavarde ) [pure virtual]
```

Nustato žmogaus pavardę.

Parameters

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

Implemented in [Studentas](#).

4.1.2.11 setVardas()

```
virtual void Human::setVardas (
    const std::string & vardas ) [pure virtual]
```

Nustato žmogaus vardą.

Parameters

<i>vardas</i>	Žmogaus vardas.
---------------	-----------------

Implemented in [Studentas](#).

4.1.2.12 skaiciuotiGalutini()

```
virtual double Human::skaiciuotiGalutini (
    bool naudotiVidurki ) const [pure virtual]
```

Skaičiuoja galutinį žmogaus rezultatą.

Parameters

<i>naudotiVidurki</i>	Ar naudoti vidurkj.
-----------------------	---------------------

Returns

Galutinis rezultatas.

Implemented in [Studentas](#).

4.1.2.13 skaiciuotiMediana()

```
virtual double Human::skaiciuotiMediana ( ) const [pure virtual]
```

Skaičiuoja žmogaus medianą.

Returns

Mediana.

Implemented in [Studentas](#).

4.1.2.14 skaiciuotiVidurki()

```
virtual double Human::skaiciuotiVidurki ( ) const [pure virtual]
```

Skaičiuoja žmogaus vidurkj.

Returns

Vidurkis.

Implemented in [Studentas](#).

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

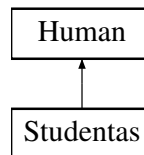
- zmogus.h

4.2 Studentas Class Reference

Klasė [Studentas](#) atstovauja studentą, paveldintį iš klasės Žmogus.

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

Inheritance diagram for Studentas:



Public Member Functions

- **Studentas ()**
Numatytasis [Studentas](#) klasės konstruktorius.
- **Studentas (const std::string &vardas, const std::string &pavarde)**
Konstruktorius su parametrais [Studentas](#) klasėje.
- **Studentas (const [Studentas](#) &other)**
[Studentas](#) klasės kopijavimo konstruktorius.
- **Studentas & operator= (const [Studentas](#) &other)**
[Studentas](#) klasės kopijavimo priskyrimo operatorius.
- **Studentas ([Studentas](#) &&other) noexcept**
[Studentas](#) klasės perkėlimo konstruktorius.
- **Studentas & operator= ([Studentas](#) &&other) noexcept**
[Studentas](#) klasės perkėlimo priskyrimo operatorius.
- **~Studentas () override**
[Studentas](#) klasės destruktorius.
- void **setVardas (const std::string &vardas) override**
Nustato studento vardą.
- std::string **getVardas () const override**
Gražina studento vardą.
- void **setPavarde (const std::string &pavarde) override**
Nustato studento pavardę.
- std::string **getPavarde () const override**
Gražina studento pavardę.
- void **setNamuDarbai (const std::vector< int > &nd) override**
Nustato studento namų darbų rezultatus.
- std::vector< int > **getNamuDarbai () const override**
Gražina studento namų darbų rezultatus.
- void **addNamuDarbas (int pazymys) override**
Prideda naują namų darbą prie studento namų darbų sąrašo.
- void **setEgzaminas (int egzaminas) override**
Nustato studento egzamino rezultatą.
- int **getEgzaminas () const override**
Gražina studento egzamino rezultatą.
- double **skaiciuotiVidurki () const override**
Skaičiuoja studento vidurkį.

- double `skaiciuotiMediana ()` const override
Skaičiuoja studento medianą.
- double `skaiciuotiGalutini (bool naudotiVidurki)` const override
Skaičiuoja galutinį studento rezultatą.
- void `atsitiktiniai ()` override
Generuoja atsitiktinius studento rezultatus.
- void `atsitiktiniaiStudentai ()` override
Generuoja atsitiktinius rezultatus studentams sąrašė.
- void `testRuleOfFive ()`
Testuoja penkių taisyklę `Studentas` klasėje.

Public Member Functions inherited from `Human`

- virtual `~Human ()`=default
Žmogaus klasės destruktorius.

Friends

- `std::ostream & operator<< (std::ostream &os, const Studentas &student)`
Perkrovimas << operatoriaus `Studentas` klasėje.
- `std::istream & operator>> (std::istream &is, Studentas &student)`
Perkrovimas >> operatoriaus `Studentas` klasėje.

4.2.1 Detailed Description

Klasė `Studentas` atstovauja studentą, paveldintį iš klasės `Žmogus`.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 `Studentas()` [1/3]

```
Studentas::Studentas (
    const std::string & vardas,
    const std::string & pavarde )
```

Konstruktorius su parametrais `Studentas` klasėje.

Parameters

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

4.2.2.2 `Studentas()` [2/3]

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

[Studentas](#) klasės kopijavimo konstruktorius.

Parameters

<i>other</i>	Kita Studentas klasės instancija.
--------------	---

4.2.2.3 Studentas() [3/3]

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

[Studentas](#) klasės perkėlimo konstruktorius.

Parameters

<i>other</i>	Kita Studentas klasės instancija.
--------------	---

4.2.3 Member Function Documentation

4.2.3.1 addNamuDarbas()

```
void Studentas::addNamuDarbas (  
    int pazymys ) [override], [virtual]
```

Prideda naują namų darbą prie studento namų darbų sąrašo.

Parameters

<i>pazymys</i>	Pridedamas namų darbo rezultatas.
----------------	-----------------------------------

Implements [Human](#).

4.2.3.2 atsitiktiniai()

```
void Studentas::atsitiktiniai ( ) [override], [virtual]
```

Generuoja atsitiktinius studento rezultatus.

Implements [Human](#).

4.2.3.3 atsitiktiniaiStudentai()

```
void Studentas::atsitiktiniaiStudentai ( ) [override], [virtual]
```

Generuoja atsitiktinius rezultatus studentams sąrašė.

Implements [Human](#).

4.2.3.4 getEgzaminas()

```
int Studentas::getEgzaminas ( ) const [override], [virtual]
```

Gražina studento egzamino rezultatą.

Returns

Studento egzamino rezultatas.

Implements [Human](#).

4.2.3.5 getNamuDarbai()

```
std::vector< int > Studentas::getNamuDarbai ( ) const [override], [virtual]
```

Gražina studento namų darbų rezultatus.

Returns

Studento namų darbų rezultatų sąrašas.

Implements [Human](#).

4.2.3.6 getPavarde()

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

Gražina studento pavardę.

Returns

Studento pavardė.

Implements [Human](#).

4.2.3.7 getVardas()

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

Gražina studento vardą.

Returns

Studento vardas.

Implements [Human](#).

4.2.3.8 operator=() [1/2]

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

[Studentas](#) klasės kopijavimo priskyrimo operatorius.

Parameters

<i>other</i>	Kita Studentas klasės instancija.
--------------	---

Returns

Nuoroda į priskirtą [Studentas](#) instanciją.

4.2.3.9 operator=() [2/2]

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

[Studentas](#) klasės perkėlimo priskyrimo operatorius.

Parameters

<i>other</i>	Kita Studentas klasės instancija.
--------------	---

Returns

Nuoroda į perkeltą [Studentas](#) instanciją.

4.2.3.10 setEgzaminas()

```
void Studentas::setEgzaminas (
    int egzaminas ) [override], [virtual]
```

Nustato studento egzamino rezultatą.

Parameters

<i>egzaminas</i>	Studento egzamino rezultatas.
------------------	-------------------------------

Implements [Human](#).

4.2.3.11 setNamuDarbai()

```
void Studentas::setNamuDarbai (
    const std::vector< int > & nd ) [override], [virtual]
```

Nustato studento namų darbų rezultatus.

Parameters

<i>nd</i>	Studento namų darbų rezultatų sąrašas.
-----------	--

Implements [Human](#).

4.2.3.12 setPavarde()

```
void Studentas::setPavarde (
    const std::string & pavarde ) [override], [virtual]
```

Nustato studento pavardę.

Parameters

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

Implements [Human](#).

4.2.3.13 setVardas()

```
void Studentas::setVardas (
    const std::string & vardas ) [override], [virtual]
```

Nustato studento vardą.

Parameters

<i>vardas</i>	Studento vardas.
---------------	------------------

Implements [Human](#).

4.2.3.14 skaiciuotiGalutini()

```
double Studentas::skaiciuotiGalutini (
    bool naudotiVidurki ) const [override], [virtual]
```

Skaičiuoja galutinį studento rezultatą.

Parameters

<i>naudotiVidurki</i>	Ar naudoti vidurkį.
-----------------------	---------------------

Returns

Galutinis rezultatas.

Implements [Human](#).

4.2.3.15 skaiciuotiMediana()

```
double Studentas::skaiciuotiMediana ( ) const [override], [virtual]
```

Skaičiuoja studento medianą.

Returns

Mediana.

Implements [Human](#).

4.2.3.16 skaiciuotiVidurki()

```
double Studentas::skaiciuotiVidurki ( ) const [override], [virtual]
```

Skaičiuoja studento vidurkį.

Returns

Vidurkis.

Implements [Human](#).

4.2.4 Friends And Related Symbol Documentation

4.2.4.1 operator<<

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

Perkrovimas << operatoriaus [Studentas](#) klasėje.

Parameters

<i>os</i>	Išvesties srautas.
<i>student</i>	Studentas objektas.

Returns

Išvesties srautas.

4.2.4.2 operator>>

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

Perkrovimas >> operatoriaus [Studentas](#) klasėje.

Parameters

<i>is</i>	Įvesties srautas.
<i>student</i>	Studentas objektas.

Returns

Įvesties srautas.

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

- studentas.h
- studentas.cpp

Chapter 5

File Documentation

5.1 functions.h

```
00001 #ifndef FUNCTIONS_H
00002 #define FUNCTIONS_H
00003
00004 #include <vector>
00005
00009 enum class ContainerType { None, Vector };
00010
00014 enum class Action { None, Generate, Sort };
00015
00020 Action getActionChoice();
00021
00028 void performAction(ContainerType containerChoice, Action actionChoice, const std::vector<int>& sizes);
00029
00036 void runApp();
00037
00038 #endif // FUNCTIONS_H
```

5.2 functions_old.h

```
00001 #ifndef FUNCTIONS_OLD_H
00002 #define FUNCTIONS_OLD_H
00003
00004 #include <vector>
00005 #include <string>
00006 #include "studentas.h"
00007
00016 void surusiuotiKategorijas(const std::vector<Studentas> &studentai, std::vector<Studentas>
&vargsiukai, std::vector<Studentas> &kietiakai, double &laikas);
00017
00025 void irasymasFaile(const std::vector<Studentas> &studentai, const std::string &failoPavadinimas,
double &laikas);
00026
00033 void rusiuotiStudentusIrIrasymas(const std::vector<Studentas> &studentai, double &laikas);
00034
00040 void generuotiStudentuFailus(const std::vector<int>& sizes);
00041
00049 void nuskaitymas(std::vector<Studentas>& studentai, const std::string& failoPavadinimas, double
&laikas);
00050
00058 bool rusiuotiPagalVarda(const Studentas& a, const Studentas& b);
00059
00067 bool rusiuotiPagalPavarde(const Studentas& a, const Studentas& b);
00068
00076 bool rusiuotiPagalGalutiniVidurki(const Studentas& a, const Studentas& b);
00077
00085 bool rusiuotiPagalGalutiniMediana(const Studentas& a, const Studentas& b);
00086
00093 void spausdinimas(const std::vector<Studentas>& studentai, const std::string& isvedimoFailas = "");
00094
00100 std::string pasirinktiFaila();
00101
00102 #endif // FUNCTIONS_OLD_H
```

5.3 functions_vector.h

```

00001 #ifndef FUNCTIONS_VECTOR_H
00002 #define FUNCTIONS_VECTOR_H
00003
00004 #include "studentas.h"
00005 #include <string>
00006 #include <vector>
00007
00018 void readDataVector(std::vector<Studentas>& studentai, const std::string& failoVardas);
00019
00025 void generateStudentFilesVector(int size);
00026
00032 void rusiuotStudentusVector1(const std::string& failoVardas);
00033
00034 #endif // FUNCTIONS_VECTOR_H

```

5.4 studentas.h

```

00001 #ifndef STUDENTAS_H
00002 #define STUDENTAS_H
00003
00004 #include "zmogus.h"
00005 #include <iostream>
00006 #include <vector>
00007
00011 class Studentas : public Human {
00012 public:
00016     Studentas();
00017
00023     Studentas(const std::string& vardas, const std::string& pavarde);
00024
00029     Studentas(const Studentas& other);
00030
00036     Studentas& operator=(const Studentas& other);
00037
00042     Studentas(Studentas&& other) noexcept;
00043
00049     Studentas& operator=(Studentas&& other) noexcept;
00050
00054     ~Studentas() override;
00055
00060     void setVardas(const std::string& vardas) override;
00061
00066     std::string getVardas() const override;
00067
00072     void setPavarde(const std::string& pavarde) override;
00073
00078     std::string getPavarde() const override;
00079
00084     void setNamuDarbai(const std::vector<int>& nd) override;
00085
00090     std::vector<int> getNamuDarbai() const override;
00091
00096     void addNamuDarbas(int pazymys) override;
00097
00102     void setEgzaminas(int egzaminas) override;
00103
00108     int getEgzaminas() const override;
00109
00114     double skaiciuotiVidurki() const override;
00115
00120     double skaiciuotiMediana() const override;
00121
00127     double skaiciuotiGalutini(bool naudotiVidurki) const override;
00128
00132     void atsitiktiniai() override;
00133
00137     void atsitiktiniaiStudentai() override;
00138
00139     // testavimas
00143     void testRuleOfFive();
00144
00151     friend std::ostream& operator<<(std::ostream& os, const Studentas& student);
00152
00159     friend std::istream& operator>>(std::istream& is, Studentas& student);
00160
00161 private:
00162     std::string vardas;
00163     std::string pavarde;
00164     std::vector<int> namuDarbai;
00165     int egzaminas;
00166 };

```

```
00167
00168 #endif // STUDENTAS_H
```

5.5 zmogus.h

```
00001 #ifndef ZMOGUS_H
00002 #define ZMOGUS_H
00003
00004 #include <string>
00005 #include <vector>
00006
00010 class Human {
00011 public:
00015     virtual ~Human() = default;
00016
00021     virtual void setVardas(const std::string& vardas) = 0;
00022
00027     virtual std::string getVardas() const = 0;
00028
00033     virtual void setPavarde(const std::string& pavarde) = 0;
00034
00039     virtual std::string getPavarde() const = 0;
00040
00045     virtual void setNamuDarbai(const std::vector<int>& nd) = 0;
00046
00051     virtual std::vector<int> getNamuDarbai() const = 0;
00052
00057     virtual void addNamuDarbas(int pazymys) = 0;
00058
00063     virtual void setEgzaminas(int egzaminas) = 0;
00064
00069     virtual int getEgzaminas() const = 0;
00070
00075     virtual double skaiciuotiVidurki() const = 0;
00076
00081     virtual double skaiciuotiMediana() const = 0;
00082
00088     virtual double skaiciuotiGalutini(bool naudotiVidurki) const = 0;
00089
00093     virtual void atsitiktiniai() = 0;
00094
00098     virtual void atsitiktiniaiStudentai() = 0;
00099 };
00100
00101 #endif // ZMOGUS_H
```


Index

- addNamuDarbas
 - Human, [8](#)
 - Studentas, [14](#)
- atsitiktiniai
 - Human, [8](#)
 - Studentas, [14](#)
- atsitiktiniaiStudentai
 - Human, [8](#)
 - Studentas, [14](#)
- getEgzaminas
 - Human, [8](#)
 - Studentas, [14](#)
- getNamuDarbai
 - Human, [9](#)
 - Studentas, [15](#)
- getPavarde
 - Human, [9](#)
 - Studentas, [15](#)
- getVardas
 - Human, [9](#)
 - Studentas, [15](#)
- Human, [7](#)
 - addNamuDarbas, [8](#)
 - atsitiktiniai, [8](#)
 - atsitiktiniaiStudentai, [8](#)
 - getEgzaminas, [8](#)
 - getNamuDarbai, [9](#)
 - getPavarde, [9](#)
 - getVardas, [9](#)
 - setEgzaminas, [9](#)
 - setNamuDarbai, [10](#)
 - setPavarde, [10](#)
 - setVardas, [10](#)
 - skaiciuotiGalutini, [10](#)
 - skaiciuotiMediana, [11](#)
 - skaiciuotiVidurki, [11](#)
- operator<<
 - Studentas, [18](#)
- operator>>
 - Studentas, [18](#)
- operator=
 - Studentas, [15](#), [16](#)
- setEgzaminas
 - Human, [9](#)
 - Studentas, [16](#)
- setNamuDarbai
 - Human, [10](#)
 - Studentas, [16](#)
- setPavarde
 - Human, [10](#)
 - Studentas, [17](#)
- setVardas
 - Human, [10](#)
 - Studentas, [17](#)
- skaiciuotiGalutini
 - Human, [10](#)
 - Studentas, [17](#)
- skaiciuotiMediana
 - Human, [11](#)
 - Studentas, [17](#)
- skaiciuotiVidurki
 - Human, [11](#)
 - Studentas, [18](#)
- Studentas, [12](#)
 - addNamuDarbas, [14](#)
 - atsitiktiniai, [14](#)
 - atsitiktiniaiStudentai, [14](#)
 - getEgzaminas, [14](#)
 - getNamuDarbai, [15](#)
 - getPavarde, [15](#)
 - getVardas, [15](#)
 - operator<<, [18](#)
 - operator>>, [18](#)
 - operator=, [15](#), [16](#)
 - setEgzaminas, [16](#)
 - setNamuDarbai, [16](#)
 - setPavarde, [17](#)
 - setVardas, [17](#)
 - skaiciuotiGalutini, [17](#)
 - skaiciuotiMediana, [17](#)
 - skaiciuotiVidurki, [18](#)
 - Studentas, [13](#), [14](#)