

Objektinis Programavimas

V2.0

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 Studentas Class Reference	7
4.1.1 Detailed Description	9
4.1.2 Constructor & Destructor Documentation	9
4.1.2.1 Studentas() [1/4]	9
4.1.2.2 Studentas() [2/4]	9
4.1.2.3 ~Studentas()	10
4.1.2.4 Studentas() [3/4]	10
4.1.2.5 Studentas() [4/4]	10
4.1.3 Member Function Documentation	11
4.1.3.1 Get_Egzaminas()	11
4.1.3.2 Get_Last_Nd()	11
4.1.3.3 Get_Mediana()	11
4.1.3.4 Get_Nd()	11
4.1.3.5 Get_Vidurkis()	11
4.1.3.6 medianosSkaiciavimas()	11
4.1.3.7 Nd_dydis()	12
4.1.3.8 Nd_empty()	12
4.1.3.9 Nd_Suma()	12
4.1.3.10 operator=() [1/2]	13
4.1.3.11 operator=() [2/2]	14
4.1.3.12 Print()	14
4.1.3.13 setEgzaminas()	14
4.1.3.14 SetMediana()	15
4.1.3.15 setNd()	15
4.1.3.16 setVidurkis()	15
4.1.3.17 Vidurkis()	15
4.1.4 Friends And Related Symbol Documentation	16
4.1.4.1 operator<< [1/2]	16
4.1.4.2 operator<< [2/2]	16
4.1.4.3 operator>> [1/2]	17
4.1.4.4 operator>> [2/2]	17
4.2 Zmogus Class Reference	18
4.2.1 Detailed Description	19

4.2.2 Constructor & Destructor Documentation	19
4.2.2.1 Zmogus()	19
4.2.3 Member Function Documentation	19
4.2.3.1 Get_Pavarde()	19
4.2.3.2 Get_Vardas()	20
4.2.3.3 Print()	20
4.2.3.4 SetPavarde()	20
4.2.3.5 SetVardas()	20
4.2.4 Member Data Documentation	20
4.2.4.1 pavarde	20
5 File Documentation	23
5.1 funkcijos.h	23
5.2 Zmogus.h	24
Index	25

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Zmogus	18
Studentas	7

Chapter 2

Class Index

2.1 Class List

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

Studentas		
Studento klasė	7
Zmogus		
Abstrakti žmogaus klasė	18

Chapter 3

File Index

3.1 File List

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

Headers/ funkcijos.h	23
Headers/ Zmogus.h	24

Chapter 4

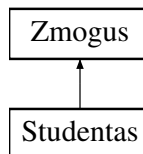
Class Documentation

4.1 Studentas Class Reference

Studento klasė.

```
#include <funkcijos.h>
```

Inheritance diagram for Studentas:



Public Member Functions

- [Studentas](#) ()
Default'inis konstruktorius.
- [Studentas](#) (const string &vard, const string &pavard)
Studento klasės konstruktorius.
- vector< int > [Get_Nd](#) () const
Namų darbų gavimui.
- int [Get_Egzaminas](#) () const
Gauti egzamino rezultata.
- double [Get_Mediana](#) () const
Gauti galutinį balą pagal medianą.
- double [Get_Vidurkis](#) () const
Gauti galutinį balą pagal vidurkį.
- bool [Nd_empty](#) () const
Patikrinti ar namų darbų vektorius yra tuščias.
- int [Nd_dydis](#) () const
Namų darbų vektoriaus dydis.
- void **nd_rusiavimas** ()
Namų darbų rūšiavimas didėjimo tvarka.
- int [Nd_Suma](#) ()

- Apskaičiuoti namų darbų sumą.*
- int [Get_Last_Nd](#) ()
Gauti namų darbų vektoriaus paskutinį pažymį.
- [~Studentas](#) ()
Studento klasės destruktorius.
- void [setEgzaminas](#) (int egz)
Priskirti egzamino rezultatą.
- void [SetMediana](#) (double med)
Priskirti galutinį balą pagal medianą.
- void [setVidurkis](#) (double vid)
Priskirti galutinį balą pagal vidurkį.
- void [setNd](#) (int nd)
Pridėti namų darbo rezultatą į vektorių.
- void [DeleteLastNd](#) ()
Ištrinti paskutinį namų darbų rezultatą.
- void [ClearEverything](#) ()
Išvalyti visą studentą.
- void [ND_clear](#) ()
Išvalyti namų darbų vektorių.
- double [Vidurkis](#) (int nd_kiekis, int nd_suma, int egzaminas)
Apskaičiuoti galutinį balą pagal vidurkį.
- double [medianosSkaiciavimas](#) (const vector< int > &namu_darbai, int nd_kiekis, int egzaminas)
Apskaičiuoti galutinį balą pagal medianą.
- void [Print](#) () const
Tik tam, kad būtų galima paveldėti [Zmogus](#) klasę.
- [Studentas](#) (const [Studentas](#) &LaikinasStudentas)
Kopijavimo konstruktorius.
- [Studentas](#) ([Studentas](#) &&LaikinasStudentas) noexcept
Perkėlimo konstruktorius.
- [Studentas](#) & [operator=](#) (const [Studentas](#) &LaikinasStudentas)
Kopijavimo priskyrimo operatorius.
- [Studentas](#) & [operator=](#) ([Studentas](#) &&LaikinasStudentas)
Perkėlimo priskyrimo operatorius.

Public Member Functions inherited from [Zmogus](#)

- string [Get_Vardas](#) () const
Vardo gavimui.
- string [Get_Pavarde](#) () const
Pavardės gavimui.
- void [SetVardas](#) (string vard)
Vardo nustatymui.
- void [SetPavarde](#) (string pav)
Pavardės nustatymui.

Friends

- `istream & operator>>` (`istream &filename`, [Studentas](#) &LaikinasStudentas)
Perkrautas įvesties operatorius darbui su failais.
- `istream & operator>>` (`istream &manual`, [Studentas](#) &LaikinasStudentas)
Perkrautas įvesties operatorius darbui su vartotoju per konsolę.
- `ostream & operator<<` (`ostream &console`, `const` [Studentas](#) &LaikinasStudentas)
Perkrautas išvesties operatorius į konsolę.
- `ofstream & operator<<` (`ofstream &filename`, `const` [Studentas](#) &LaikinasStudentas)
Perkrautas išvesties operatorius į failą.

Additional Inherited Members

Protected Member Functions inherited from [Zmogus](#)

- [Zmogus](#) (`const string &vard`, `const string &pavard`)
[Zmogus](#) klasės konstruktorius.
- `virtual ~Zmogus` ()
[Zmogus](#) klasės destruktorius.

Protected Attributes inherited from [Zmogus](#)

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

4.1.1 Detailed Description

Studento klasė.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Studentas() [1/4]

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

Default'inis konstruktorius.

Studento klasės default'inis konstruktorius. Vardo ir pavardės nustatymas į tuščią string'ą. Namų darbų vektorius išvalomas. Egzamino, medianos ir vidurkio balas nustatomas į 0.

4.1.2.2 Studentas() [2/4]

```
Studentas::Studentas (
    const string & vard,
    const string & pavard )
```

Studento klasės konstruktorius.

Parameters

<i>vard</i>	Studento vardas.
<i>pavard</i>	Studento pavardė.
<i>vard</i>	Studento vardas.
<i>pavard</i>	Studento pavardė. Vardo ir pavardės nustatymas į perduotus parametrus. Namų darbų vektorius išvalomas. Egzamino, medianos ir vidurkio balas nustatomas į 0.

4.1.2.3 ~Studentas()

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

Studento klasės destruktorius.

Studento klasės destruktorius. Namų darbų vektorius išvalomas.

4.1.2.4 Studentas() [3/4]

```
Studentas::Studentas (
    const Studentas & LaikinasStudentas )
```

Kopijavimo konstruktorius.

Studento klasės kopijavimo konstruktorius.

Parameters

<i>LaikinasStudentas</i>	Objektas, kurį reikia kopijuoti.
<i>LaikinasStudentas</i>	Studentas , kurio duomenys kopijuojami. Vardas ir pavardė nukopijuojami. Namų darbų vektorius nukopijuojamas. Egzamino, medianos ir vidurkio balas nukopijuojamas.

4.1.2.5 Studentas() [4/4]

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

Perkėlimo konstruktorius.

Studento klasės perkėlimo konstruktorius.

Parameters

<i>LaikinasStudentas</i>	Objektas, kurį reikia perkelti.
<i>LaikinasStudentas</i>	Studentas , kurio duomenys perkeliama. Vardas ir pavardė perkeliama. Namų darbų vektorius perkeliama. Egzamino, medianos ir vidurkio balas perkeliama.

4.1.3 Member Function Documentation

4.1.3.1 Get_Egzaminas()

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

Gauti egzamino rezultatą.

Returns

Egzamino rezultatą.

4.1.3.2 Get_Last_Nd()

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

Gauti namų darbų vektoriaus paskutinį pažymį.

Returns

Paskutinį pažymį namų darbų vektoriuje.

4.1.3.3 Get_Mediana()

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

Gauti galutinį balą pagal medianą.

Returns

Galutinis balas pagal medianą.

4.1.3.4 Get_Nd()

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

Namų darbų gavimui.

Returns

Namų darbų vektorių.

4.1.3.5 Get_Vidurkis()

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

Gauti galutinį balą pagal vidurkį.

Returns

galutinis balas pagal vidurkį.

4.1.3.6 medianosSkaiciavimas()

```
double Studentas::medianosSkaiciavimas (
    const vector< int > & namu_darbai,
    int nd_kiekis,
    int egzaminas )
```

Apskaičiuoti galutinį balą pagal medianą.

Studento klasės galutinio balo skaičiavimas pagal medianą.

Parameters

<i>namu_darbai</i>	Namų darbų vektorius.
<i>nd_kiekis</i>	Namų darbų rezultatų kiekis.
<i>egzaminas</i>	Egzamino rezultatas.

Returns

Galutinį balą pagal medianą.

Parameters

<i>namu_darbai</i>	Namų darbų vektorius.
<i>nd_kiekis</i>	Namų darbų kiekis.
<i>egzaminas</i>	Egzamino balas.

Returns

Grąžinamas galutinis balas pagal medianą.

4.1.3.7 Nd_dydis()

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

Namų darbų vektoriaus dydis.

Returns

Namų darbų vektoriaus dydį.

4.1.3.8 Nd_empty()

```
bool Studentas::Nd_empty ( ) const [inline]
```

Patikrinti ar namų darbų vektorius yra tuščias.

Returns

True, jei vektorius tuščias. False, jei vektorius turi elementų.

4.1.3.9 Nd_Suma()

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

Apskaičiuoti namų darbų sumą.

Returns

namų darbų vektoriaus elementų sumą.

4.1.3.10 operator=() [1/2]

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

Kopijavimo priskyrimo operatorius.

Studento klasės kopijavimo operatorius.

Parameters

<i>LaikinasStudentas</i>	Objektas, kurį reikia kopijuoti.
<i>LaikinasStudentas</i>	Studentas , kurio duomenys kopijuojami.

Returns

Grąžinamas [Studentas](#) su nukopijuotais duomenimis.

4.1.3.11 operator=() [2/2]

```
Studentas & Studentas::operator= (
    Studentas && LaikinasStudentas )
```

Perkėlimo priskyrimo operatorius.

Studento klasės perkėlimo operatorius.

Parameters

<i>LaikinasStudentas</i>	Objektas, kurį reikia perkelti.
<i>LaikinasStudentas</i>	Studentas , kurio duomenys perkeliama.

Returns

Grąžinamas [Studentas](#) su perkeltais duomenimis.

4.1.3.12 Print()

```
void Studentas::Print ( ) const [virtual]
```

Tik tam, kad būtų galima paveldėti [Zmogus](#) klasę.

Implements [Zmogus](#).

4.1.3.13 setEgzaminas()

```
void Studentas::setEgzaminas (
    int egz ) [inline]
```

Priskirti egzamino rezultatą.

Parameters

<i>egz</i>	egzamino rezultatas.
------------	----------------------

4.1.3.14 SetMediana()

```
void Studentas::SetMediana (
    double med ) [inline]
```

Priskirti galutinį balą pagal medianą.

Parameters

<i>med</i>	galutinio balo pagal medianą rezultatas.
------------	--

4.1.3.15 setNd()

```
void Studentas::setNd (
    int nd ) [inline]
```

Pridėti namų darbo rezultatą į vektorių.

Parameters

<i>nd</i>	Namų darbo rezultatas.
-----------	------------------------

4.1.3.16 setVidurkis()

```
void Studentas::setVidurkis (
    double vid ) [inline]
```

Priskirti galutinį balą pagal vidurkį.

Parameters

<i>vid</i>	Galutinio balo pagal vidurkį rezultatas.
------------	--

4.1.3.17 Vidurkis()

```
double Studentas::Vidurkis (
    int nd_kiekis,
    int nd_suma,
    int egzaminas )
```

Apskaičiuoti galutinį balą pagal vidurkį.

Studento klasės galutinio balo skaičiavimas pagal vidurkį.

Parameters

<i>nd_kiekis</i>	Namų darbų rezultatų kiekis.
<i>nd_suma</i>	Visų namų darbų suma.
<i>egzaminas</i>	Egzamino rezultatas.

Returns

Galutinį balą pagal vidurkį.

Parameters

<i>nd_kiekis</i>	Namų darbų kiekis.
<i>nd_suma</i>	Namų darbų suma.
<i>egzaminas</i>	Egzamino balas.

Returns

Grąžinamas galutinis balas pagal vidurkį.

4.1.4 Friends And Related Symbol Documentation

4.1.4.1 operator<< [1/2]

```
ostream & operator<< (
    ostream & filename,
    const Studentas & LaikinasStudentas ) [friend]
```

Perkrautas išvesties operatorius į failą.

Parameters

<i>filename</i>	Išvesties ostream objektas.
<i>LaikinasStudentas</i>	Objektas, kurį reikia išvesti.

Returns

Išvesties ostream objektą.

Parameters

<i>filename</i>	Išvesties failas.
<i>LaikinasStudentas</i>	Studentas, kurio duomenys išvedami.

Returns

Grąžinamas Studentas su išvestais duomenimis.

4.1.4.2 operator<< [2/2]

```
ostream & operator<< (
    ostream & console,
    const Studentas & LaikinasStudentas ) [friend]
```

Perkrautas išvesties operatorius į konsolę.

Parameters

<i>console</i>	Išvesties ostream objektas.
<i>LaikinasStudentas</i>	Objektas, kurį reikia išvesti.

Returns

Išvesties ostream objektą.

Parameters

<i>console</i>	Išvesties srautas.
<i>LaikinasStudentas</i>	Studentas , kurio duomenys išvedami.

Returns

Grąžinamas [Studentas](#) su išvestais duomenimis.

4.1.4.3 operator>> [1/2]

```
istream & operator>> (
    istream & manual,
    Studentas & LaikinasStudentas ) [friend]
```

Perkrautas įvesties operatorius darbui su vartotoju per konsolę.

Parameters

<i>manual</i>	Įvesties istream objektas.
<i>LaikinasStudentas</i>	Objektas į kurį reikia skaityti duomenis.

Returns

Įvesties istream objektą.

Parameters

<i>manual</i>	Įvesties srautas.
<i>LaikinasStudentas</i>	Studentas , kuriam bus priskiriami duomenys.

Returns

Grąžinamas [Studentas](#) su nuskaitytais duomenimis.

4.1.4.4 operator>> [2/2]

```
istreamstream & operator>> (
```

```
istream & filename,
Studentas & LaikinasStudentas ) [friend]
```

Perkrautas įvesties operatorius darbui su failais.

Parameters

<i>filename</i>	Įvesties stringstream objektas.
<i>LaikinasStudentas</i>	Objektas į kurį reikia skaityti duomenis.

Returns

Įvesties stringstream objektą.

Parameters

<i>filename</i>	Įvesties failas.
<i>LaikinasStudentas</i>	Studentas , kuriam bus priskiriami duomenys.

Returns

Grąžinamas [Studentas](#) su nuskaitytais duomenimis.

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

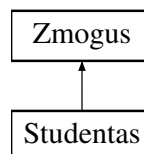
- Headers/funkcijos.h
- Sources/funkcijos.cpp

4.2 Zmogus Class Reference

Abstrakti žmogaus klasė.

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

Inheritance diagram for Zmogus:



Public Member Functions

- string [Get_Vardas](#) () const
Vardo gavimui.
- string [Get_Pavarde](#) () const
Pavardės gavimui.
- void [SetVardas](#) (string vard)
Vardo nustatymui.
- void [SetPavarde](#) (string pav)
Pavardės nustatymui.
- virtual void [Print](#) () const =0
Spausdinimo funkcija padaranti klasę abstrakčia.

Protected Member Functions

- [Zmogus](#) (const string &vard, const string &pavard)
[Zmogus](#) klasės konstruktorius.
- virtual `~Zmogus` ()
[Zmogus](#) klasės destruktorius.

Protected Attributes

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

4.2.1 Detailed Description

Abstrakti žmogaus klasė.

[Zmogus](#) klasė yra abstrakti klasė, kuri turi tik vardą ir pavardę. Ji turi virtualią funkciją [Print\(\)](#), kuri yra privaloma klasėms, kurios paveldi [Zmogus](#) klasę.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Zmogus()

```
Zmogus::Zmogus (
    const string & vard,
    const string & pavard ) [inline], [protected]
```

[Zmogus](#) klasės konstruktorius.

Default'inis konstruktorius

Parameters

<i>vard</i>	Studento vardas.
<i>pavard</i>	Studento pavardė.

4.2.3 Member Function Documentation

4.2.3.1 Get_Pavarde()

```
string Zmogus::Get_Pavarde ( ) const [inline]
```

Pavardės gavimui.

Returns

Pavardę.

4.2.3.2 Get_Vardas()

```
string Zmogus::Get_Vardas ( ) const [inline]
```

Vardo gavimui.

Returns

Vardą.

4.2.3.3 Print()

```
virtual void Zmogus::Print ( ) const [pure virtual]
```

Spausdinimo funkcija padaranti klasę abstrakčia.

Implemented in [Studentas](#).

4.2.3.4 SetPavarde()

```
void Zmogus::SetPavarde (
    string pav ) [inline]
```

Pavardės nustatymui.

Parameters

<i>pav</i>	Pavardė.
------------	----------

4.2.3.5 SetVardas()

```
void Zmogus::SetVardas (
    string vard ) [inline]
```

Vardo nustatymui.

Parameters

<i>vard</i>	Vardas.
-------------	---------

4.2.4 Member Data Documentation

4.2.4.1 pavarde

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


Saugo vardą ir pavardę

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

- Headers/Zmogus.h

Chapter 5

File Documentation

5.1 funkcijos.h

```
00001 #ifndef FUNKCIJOS_H
00002 #define FUNKCIJOS_H
00003
00004 #include "Zmogus.h"
00005 #include <vector>
00006 #include <string>
00007 #include <iostream>
00008 #include <iomanip>
00009 #include <algorithm>
00010 #include <limits>
00011 #include <numeric>
00012 #include <random>
00013 #include <ctime>
00014 #include <fstream>
00015 #include <chrono>
00016
00017 using namespace std;
00018
00022 class Studentas : public Zmogus {
00023     private:
00024         vector<int> namu_darbai;
00025         int egzaminas;
00026         double mediana;
00027         double vidurkis;
00028     public:
00032         Studentas();
00033
00039         Studentas(const string &vard, const string &pavard);
00040
00045         vector<int> Get_Nd() const { return namu_darbai; }
00046
00051         int Get_Egzaminas() const { return egzaminas; }
00052
00057         double Get_Mediana() const { return mediana; }
00058
00063         double Get_Vidurkis() const { return vidurkis; }
00064
00069         bool Nd_empty() const { return namu_darbai.empty(); }
00070
00075         int Nd_dydis() const { return namu_darbai.size(); }
00076
00080         void nd_rusiavimas() { sort(namu_darbai.begin(), namu_darbai.end()); }
00081
00086         int Nd_Suma() { return accumulate(namu_darbai.begin(), namu_darbai.end(), 0); }
00087
00092         int Get_Last_Nd() { return namu_darbai.back(); }
00093
00097         ~Studentas();
00098
00103         void setEgzaminas(int egz) { this->egzaminas = egz; }
00104
00109         void SetMediana(double med) { this->mediana = med; }
00110
00115         void setVidurkis(double vid) { this->vidurkis = vid; }
00116
00121         void setNd(int nd) { this->namu_darbai.push_back(move(nd)); }
00122
00126         void DeleteLastNd() { this->namu_darbai.pop_back(); }
```

```

00127
00131     void ClearEverything() { this->vardas.clear(); this->pavarde.clear();
this->namu_darbai.clear(); this->egzaminas = 0; this->mediana = 0.0; this->vidurkis = 0.0; }
00132
00136     void ND_clear() { this->namu_darbai.clear(); }
00137
00145     double Vidurkis(int nd_kiekis, int nd_suma, int egzaminas);
00146
00154     double medianosSkaiciavimas(const vector<int> &namu_darbai, int nd_kiekis, int egzaminas);
00155
00159     void Print() const;
00160
00165     Studentas(const Studentas &LaikinasStudentas);
00166
00171     Studentas(Studentas &&LaikinasStudentas) noexcept;
00172
00177     Studentas& operator=(const Studentas &LaikinasStudentas);
00178
00183     Studentas& operator=(Studentas &&LaikinasStudentas);
00184
00191     friend istream& operator>(istream& filename, Studentas &LaikinasStudentas);
00192
00199     friend istream& operator>(istream& manual, Studentas &LaikinasStudentas);
00200
00207     friend ostream& operator<(ostream& console, const Studentas &LaikinasStudentas);
00208
00215     friend ofstream& operator<(ofstream& filename, const Studentas &LaikinasStudentas);
00216 };
00217
00218 extern int norima_isvedimo_vieta;
00219 extern int programos_veikimas;
00220 extern int norimas_rikiavimas;
00221 extern char programos_tesinys;
00222 extern char choice3;
00224 extern vector<string> Vardai;
00225 extern vector<string> Pavardes;
00226 extern vector<Studentas> Lievi;
00227 extern vector<Studentas> Studentai;
00235 bool palygintiPagalVarda(const Studentas &a, const Studentas &b);
00236
00243 bool palygintiPagalPavarde(const Studentas &a, const Studentas &b);
00244
00251 bool palygintiPagalVidurki(const Studentas &a, const Studentas &b);
00252
00259 bool palygintiPagalMediana(const Studentas &a, const Studentas &b);
00260
00266 void PasalintiKietusStudentus(vector<Studentas> &Studentai, int norimas_rikiavimas);
00267
00273 void GeneruotiFaila(int kiekis, int nd_kiekis);
00274
00279 void GeneruotiPazymius(Studentas &LaikinasStudentas);
00280
00285 void GeneruotiStudenta(Studentas &LaikinasStudentas);
00286
00293 void IsvestiRezultatus(string pavadinimas, const vector<Studentas> &Studentai, int
norima_isvedimo_vieta);
00294
00295 #endif

```

5.2 Zmogus.h

```

00001 #ifndef ZMOGUS_H
00002 #define ZMOGUS_H
00003
00004 #include <string>
00005 using namespace std;
00006
00007
00015 class Zmogus {
00016     protected:
00017         string vardas, pavarde;
00019         Zmogus() : vardas(""), pavarde("") {};
00025         Zmogus(const string &vard, const string &pavard) : vardas(vard), pavarde(pavard) {};
00029         virtual ~Zmogus() {};
00030     public:
00035         inline string Get_Vardas() const { return vardas; }
00040         inline string Get_Pavarde() const { return pavarde; }
00045         void SetVardas(string vard) { this->vardas = vard; }
00050         void SetPavarde(string pav) { this->pavarde = pav; }
00054         virtual void Print() const = 0;
00055 };
00056
00057 #endif

```

Index

~Studentas
Studentas, [10](#)

Get_Egzaminas
Studentas, [11](#)

Get_Last_Nd
Studentas, [11](#)

Get_Mediana
Studentas, [11](#)

Get_Nd
Studentas, [11](#)

Get_Pavarde
Zmogus, [19](#)

Get_Vardas
Zmogus, [19](#)

Get_Vidurkis
Studentas, [11](#)

Headers/funkcijos.h, [23](#)

Headers/Zmogus.h, [24](#)

medianosSkaiciavimas
Studentas, [11](#)

Nd_dydis
Studentas, [12](#)

Nd_empty
Studentas, [12](#)

Nd_Suma
Studentas, [12](#)

operator<<
Studentas, [16](#)

operator>>
Studentas, [17](#)

operator=
Studentas, [12](#), [14](#)

pavarde
Zmogus, [20](#)

Print
Studentas, [14](#)
Zmogus, [20](#)

setEgzaminas
Studentas, [14](#)

SetMediana
Studentas, [14](#)

setNd
Studentas, [15](#)

SetPavarde

Zmogus, [20](#)

SetVardas
Zmogus, [20](#)

setVidurkis
Studentas, [15](#)

Studentas, [7](#)
~Studentas, [10](#)
Get_Egzaminas, [11](#)
Get_Last_Nd, [11](#)
Get_Mediana, [11](#)
Get_Nd, [11](#)
Get_Vidurkis, [11](#)
medianosSkaiciavimas, [11](#)

Nd_dydis, [12](#)

Nd_empty, [12](#)

Nd_Suma, [12](#)

operator<<, [16](#)

operator>>, [17](#)

operator=, [12](#), [14](#)

Print, [14](#)

setEgzaminas, [14](#)

SetMediana, [14](#)

setNd, [15](#)

setVidurkis, [15](#)

Studentas, [9](#), [10](#)

Vidurkis, [15](#)

Vidurkis
Studentas, [15](#)

Zmogus, [18](#)
Get_Pavarde, [19](#)
Get_Vardas, [19](#)
pavarde, [20](#)
Print, [20](#)
SetPavarde, [20](#)
SetVardas, [20](#)
Zmogus, [19](#)