

OOP2V2.0

V2.0

Generated by Doxygen 1.14.0



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy . . . . .	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List . . . . .	3
<b>3 File Index</b>	<b>5</b>
3.1 File List . . . . .	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 Stud Class Reference . . . . .	7
4.1.1 Detailed Description . . . . .	8
4.1.2 Member Function Documentation . . . . .	8
4.1.2.1 calculateGalMediana() . . . . .	8
4.1.2.2 calculateGalVidurkis() . . . . .	8
4.1.2.3 getPav() . . . . .	9
4.1.2.4 getVar() . . . . .	9
4.1.2.5 setPav() . . . . .	9
4.1.2.6 setVar() . . . . .	9
4.2 Zmogus Class Reference . . . . .	9
4.2.1 Detailed Description . . . . .	10
<b>5 File Documentation</b>	<b>11</b>
5.1 functionsCallsVector.h . . . . .	11
5.2 meinelib.h . . . . .	11
5.3 studentas.h . . . . .	12
5.4 zmogus.h . . . . .	14
<b>Index</b>	<b>17</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

Zmogus . . . . .	9
Stud . . . . .	7



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Stud</a> . . . . .	<a href="#">7</a>
<a href="#">Zmogus</a> . . . . .	<a href="#">9</a>





# Chapter 3

## File Index

### 3.1 File List

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

Include/ <a href="#">functionsCallsVector.h</a> . . . . .	11
Include/ <a href="#">meinelib.h</a> . . . . .	11
Include/ <a href="#">studentas.h</a> . . . . .	12
Include/ <a href="#">zmogus.h</a> . . . . .	14



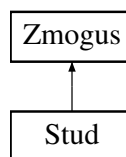
## Chapter 4

# Class Documentation

### 4.1 Stud Class Reference

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

Inheritance diagram for Stud:



#### Public Member Functions

- **Stud** (std::string var="", std::string pav="", std::vector< int > pazymys={}, int egz={})
- **Stud** (const Stud &other)
- Stud & **operator=** (const Stud &other)
- **Stud** (Stud &&other) noexcept
- Stud & **operator=** (Stud &&other) noexcept
- void **setVar** (const std::string &var) override
- void **setPav** (const std::string &pav) override
- void **setEgz** (const int egz)
- void **addPazymys** (const int pazymys)
- void **removePazymys** ()
- void **calculateGalVidurkis** ()
- void **calculateGalMediana** ()
- std::string **getVar** () const override
- std::string **getPav** () const override
- std::vector< int > **getPazymys** () const
- int **getEgz** () const
- float **getVidurkis** () const
- float **getMediana** () const

## Public Member Functions inherited from [Zmogus](#)

- **Zmogus** (std::string var="", std::string pav="")
- **Zmogus** (const [Zmogus](#) &other)
- **Zmogus & operator=** (const [Zmogus](#) &other)
- **Zmogus** ([Zmogus](#) &&other) noexcept
- **Zmogus & operator=** ([Zmogus](#) &&other) noexcept

## Static Public Member Functions

- static int **ivestiesPatikrinimas** (const int nuo, const int iki)
- static int **ivestiesPatikrinimas** (const int nuo, const int iki, const int sustabdymoSalyga)

## Friends

- std::istream & **operator>>** (std::istream &is, [Stud](#) &s)
- std::ostream & **operator<<** (std::ostream &os, const [Stud](#) &s)

## Additional Inherited Members

## Protected Attributes inherited from [Zmogus](#)

- std::string **var\_** {}
- std::string **pav\_** {}

### 4.1.1 Detailed Description

Struktura, kurioje laikomi studento duomenys.

### 4.1.2 Member Function Documentation

#### 4.1.2.1 calculateGalMediana()

```
void Stud::calculateGalMediana ()
```

Suskaiciuoja studento galutine mediana pagal namu darbu ir egzamino pazymius.

#### 4.1.2.2 calculateGalVidurkis()

```
void Stud::calculateGalVidurkis ()
```

Suskaiciuoja studento galutini vidurki pagal namu darbu ir egzamino pazymius.

#### 4.1.2.3 getPav()

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

Reimplemented from [Zmogus](#).

#### 4.1.2.4 getVar()

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

Reimplemented from [Zmogus](#).

#### 4.1.2.5 setPav()

```
void Stud::setPav (  
    const std::string & pav) [inline], [override], [virtual]
```

Implements [Zmogus](#).

#### 4.1.2.6 setVar()

```
void Stud::setVar (  
    const std::string & var) [inline], [override], [virtual]
```

Implements [Zmogus](#).

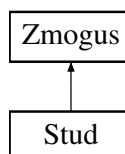
The documentation for this class was generated from the following files:

- Include/studentas.h
- src/Stud.cpp

## 4.2 Zmogus Class Reference

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

Inheritance diagram for Zmogus:



### Public Member Functions

- **Zmogus** (std::string var="", std::string pav="")
- **Zmogus** (const [Zmogus](#) &other)
- **Zmogus & operator=** (const [Zmogus](#) &other)
- **Zmogus** ([Zmogus](#) &&other) noexcept
- **Zmogus & operator=** ([Zmogus](#) &&other) noexcept
- virtual void **setVar** (const std::string &var)=0
- virtual void **setPav** (const std::string &pav)=0
- virtual std::string **getVar** () const
- virtual std::string **getPav** () const

### Protected Attributes

- std::string **var\_** {}
- std::string **pav\_** {}

### Friends

- std::istream & **operator>>** (std::istream &is, [Zmogus](#) &s)
- std::ostream & **operator<<** (std::ostream &os, const [Zmogus](#) &s)

## 4.2.1 Detailed Description

Bazine abstrakti [Zmogus](#) klase, kuri saugo varda ir pavarde.

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

- Include/zmogus.h

# Chapter 5

## File Documentation

### 5.1 functionsCallsVector.h

```
00001 #ifndef FUNCTIONSCALLSV_H
00002 #define FUNCTIONSCALLSV_H
00003
00004 // Laikomi funkciju prototipai.
00005
00006 // Prototipai
00007
00008 // iverstis.cpp
00009 void readRanka(Stud& stu);
00010 void readName_makeGrade(Stud& stu);
00011 void makeStud(Stud& stu);
00012 void fileRead(vector<Stud>& studentai, string vardas);
00013
00014 // isvestis.cpp
00015 void isvestiesMenu(vector<Stud>& studentai);
00016 void isvestis(vector<Stud>& studentai, std::ostream& isvestiesMetodas, const int
    galutinioPasirinkimas);
00017
00018 // studentuRusiavimas.cpp
00019 void rusiavimas(vector<Stud>& studentai, int rusiavimoPasirinkimas);
00020
00021 // generators.cpp
00022 void randomStudentas(Stud& studentas, bool vyras);
00023 void randomAtsitiktinisPazymys(Stud& stu);
00024
00025 // fileGenerator.cpp
00026 void failoKurimas(int studentuSk);
00027
00028 // studentuSkirstymas.cpp
00029 void fileFilter(vector<Stud>& studentai, const int galutinioPasirinkimas, const int
    rusiavimoPasirinkimas);
00030
00031 // test.cpp
00032 void testMenu();
00033 void nuskaitymoTestas();
00034 void studentuTest();
00035 void programTest();
00036
00037 // iverstisPatikrinimas.cpp
00038 int iverstiesPatikrinimas(const int nuo, const int iki);
00039 int iverstiesPatikrinimas(const int nuo, const int iki, const int sustabdymoSalyga);
00040
00041 // Stud.cpp
00042 // Stud klases Medianos ir Vidurkio funkciju deklaracijos pacioje klaseje
00043 void studentuGalutiniuSkaiciavimas(vector<Stud>& studentai);
00044
00045
00046 #endif
```

### 5.2 meinelib.h

```
00001 #ifndef MEINELIB_H
00002 #define MEINELIB_H
```

```

00003
00005 #include <iostream>
00006 #include <iomanip>
00007 #include <string>
00008 #include <vector>
00009 #include <deque>
00010 #include <list>
00011 #include <fstream>
00012 #include <algorithm>
00013 #include <numeric>
00014 #include <sstream>
00015 #include <cstdlib>
00016 #include <cmath>
00017 #include <chrono>
00018 #include <filesystem>
00019
00020 namespace fs = std::filesystem;
00021
00022 using std::cout;
00023 using std::cin;
00024 using std::endl;
00025 using std::string;
00026 using std::vector;
00027 using std::deque;
00028 using std::list;
00029 using std::fixed;
00030 using std::setprecision;
00031 using std::sort;
00032 using hrClock = std::chrono::high_resolution_clock;
00033 using ms = std::chrono::milliseconds;
00034 using sec = std::chrono::duration<double>;
00035
00036
00037 #endif

```

## 5.3 studentas.h

```

00001 #ifndef STUDENTAS_H
00002 #define STUDENTAS_H
00003
00004 #include "zmogus.h"
00005
00007 class Stud : public Zmogus {
00008 private:
00009     std::vector<int> pazymys_{};
00010     int egz_{};
00011     float galVidurkis_{};
00012     float galMediana_{};
00013
00014 public:
00015     // Konstruktorius ir desktrutorius
00016     explicit Stud(std::string var = "", std::string pav = "", std::vector<int> pazymys = {}, int egz =
00017 {}):
00018         Zmogus(std::move(var), std::move(pav)), pazymys_(std::move(pazymys)), egz_(egz) {}
00019
00019     ~Stud() = default;
00020
00021     // Copy constructor
00022     Stud(const Stud& other):
00023         Zmogus(other.var_, other.pav_),
00024         pazymys_(other.pazymys_),
00025         egz_(other.egz_),
00026         galVidurkis_(other.galVidurkis_),
00027         galMediana_(other.galMediana_) {}
00028
00029     // Copy assignment operator
00030     Stud& operator=(const Stud& other) {
00031         if (this != &other) {
00032             Zmogus::operator=(other);
00033             pazymys_ = other.pazymys_;
00034             egz_ = other.egz_;
00035             galVidurkis_ = other.galVidurkis_;
00036             galMediana_ = other.galMediana_;
00037         }
00038         return *this;
00039     }
00040
00041     // Move constructor
00042     Stud(Stud&& other) noexcept:
00043         Zmogus(std::move(other)),
00044         pazymys_(std::move(other.pazymys_)),
00045         egz_(other.egz_),
00046         galVidurkis_(other.galVidurkis_),

```



```

00047     galMediana_(other.galMediana_) {
00048         other.egz_ = 0;
00049     }
00050
00051     // Move assignment operator
00052     Stud& operator=(Stud&& other) noexcept {
00053         if (this != &other) {
00054             Zmogus::operator=(std::move(other));
00055             pazymys_ = std::move(other.pazymys_);
00056             egz_ = other.egz_;
00057             galVidurkis_ = other.galVidurkis_;
00058             galMediana_ = other.galMediana_;
00059             other.egz_ = 0;
00060         }
00061         return *this;
00062     }
00063
00064     // Ivesties operatorius
00065     friend std::istream& operator>>(std::istream& is, Stud& s) {
00066         is >> static_cast<Zmogus&>(s);
00067
00068         int egz;
00069         std::vector<int> pazymiai;
00070         int paz;
00071
00072         cout << "Iveskite egzamino pazymi: ";
00073         egz = ivestiesPatikrinimas(0, 10);
00074
00075         cout << "Iveskite pazymius 0 iki 10, norint baigti iveskite -1:\n";
00076         while(true) {
00077             paz = ivestiesPatikrinimas(0, 10, -1);
00078             if (paz == -1) break;
00079             pazymiai.push_back(paz);
00080         }
00081
00082         s.setEgz(egz);
00083         for (int p : pazymiai) s.addPazymys(p);
00084
00085         s.calculateGalVidurkis();
00086         s.calculateGalMediana();
00087         return is;
00088     }
00089
00090     // Ivesties operatorius
00091     friend std::ostream& operator<<(std::ostream& os, const Stud& s) {
00092         os << static_cast<const Zmogus&>(s);
00093         os << " Egzaminas: " << s.getEgz() << " Pazymiai: ";
00094         for (int p : s.getPazymys()) {
00095             os << p << " ";
00096         }
00097         os << "Vidurkis: " << s.getVidurkis() << " Mediana: " << s.getMediana();
00098         return os;
00099     }
00100
00101     // Setteriai, kurie nustato studento varda, pavarde, uzduotis ir egzamino pazymi.
00102     void setVar(const std::string& var) override { var_ = var; }
00103     void setPav(const std::string& pav) override { pav_ = pav; }
00104     void setEgz(const int egz) { egz_ = egz; }
00105
00106     // Papildomos funkcijos, kurias prideda ir pasalina uzduociu pazymius
00107     void addPazymys(const int pazymys) { pazymys_.push_back(pazymys); }
00108     void removePazymys() { pazymys_.pop_back(); }
00109
00110     void calculateGalVidurkis();
00111     void calculateGalMediana();
00112
00113     // Getteriai, kurie grazina studento varda, pavarde, uzduotis, egzamino pazymi ir galutini pazymi.
00114     std::string getVar() const override { return var_; }
00115     std::string getPav() const override { return pav_; }
00116     std::vector<int> getPazymys() const { return pazymys_; }
00117     int getEgz() const { return egz_; }
00118     float getVidurkis() const { return galVidurkis_; }
00119     float getMediana() const { return galMediana_; }
00120
00121     // Ivesties patikrinimas
00122     static int ivestiesPatikrinimas(const int nuo, const int iki) {
00123         int input{};
00124         while (true) {
00125             try {
00126                 cin >> input;
00127                 if (input < nuo || input > iki) {
00128                     cout << "\n\n!!!!Iveskite skaiciu nuo " << nuo << " iki " << iki << ".!!!!\n\n";
00129                     continue;
00130                 }
00131             }
00132             catch (...) {
00133                 cin.clear();

```

```

00134         cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
00135         cout << "\n\n!!!!Ivestis neteisinga. Bandykite isnaujo!!!!\n\n\n";
00136         continue;
00137     }
00138     break;
00139 }
00140 return input;
00141 }
00142
00143 static int ivestiesPatikrinimas(const int nuo, const int iki, const int sustabdymoSalyga) {
00144     int input{};
00145     while (true) {
00146         try {
00147             cin >> input;
00148             if (input == sustabdymoSalyga) {
00149                 return sustabdymoSalyga;
00150             }
00151
00152             if (input < nuo || input > iki) {
00153                 cout << "\n\n!!!!Iveskite skaiciu nuo " << nuo << " iki " << iki << ".!!!!\n\n\n";
00154                 continue;
00155             }
00156         }
00157         catch (...) {
00158             cin.clear();
00159             cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
00160             cout << "\n\n!!!!Ivestis neteisinga. Bandykite isnaujo!!!!\n\n\n";
00161             continue;
00162         }
00163         break;
00164     }
00165     return input;
00166 }
00167 };
00168
00169 #endif

```

## 5.4 zmogus.h

```

00001 #ifndef ZMOGUS_H
00002 #define ZMOGUS_H
00003
00004
00005 class Zmogus {
00006 protected:
00007     std::string var_{}, pav_{};
00008
00009 public:
00010     // Konstruktorius ir desktrutorius
00011     explicit Zmogus(std::string var = "", std::string pav = "") :
00012         var_(std::move(var)), pav_(std::move(pav)) {}
00013
00014     ~Zmogus() = default;
00015
00016     // Copy constructor
00017     Zmogus(const Zmogus& other) :
00018         var_(other.var_),
00019         pav_(other.pav_){}
00020
00021     // Copy assignment operator
00022     Zmogus& operator=(const Zmogus& other) {
00023         if (this != &other) {
00024             var_ = other.var_;
00025             pav_ = other.pav_;
00026         }
00027         return *this;
00028     }
00029
00030     // Move constructor
00031     Zmogus(Zmogus&& other) noexcept :
00032         var_(std::move(other.var_)),
00033         pav_(std::move(other.pav_)){}
00034
00035     // Move assignment operator
00036     Zmogus& operator=(Zmogus&& other) noexcept {
00037         if (this != &other) {
00038             var_ = std::move(other.var_);
00039             pav_ = std::move(other.pav_);
00040         }
00041         return *this;
00042     }
00043
00044     // Ivesties operatorius
00045     friend std::istream& operator>>(std::istream& is, Zmogus& s) {

```

```
00046         std::string var, pav;
00047
00048         cout << "Iveskite varda: ";
00049         is » var;
00050         cout << "Iveskite pavarde: ";
00051         is » pav;
00052
00053         s.setVar(var);
00054         s.setPav(pav);
00055         return is;
00056     }
00057
00058     // Ivesties operatorius
00059     friend std::ostream& operator<<(std::ostream& os, const Zmogus& s) {
00060         os << "Vardas: " << s.getVar() << " Pavarde: " << s.getPav();
00061         return os;
00062     }
00063
00064     // Setteriai, kurie nustato zmogaus varda ir pavarde
00065     virtual void setVar(const std::string& var) = 0;
00066     virtual void setPav(const std::string& pav) = 0;
00067
00068     // Getteriai, kurie grazina zmogaus varda ir pavarde
00069     virtual std::string getVar() const { return var_; }
00070     virtual std::string getPav() const { return pav_; }
00071 };
00072
00073 #endif
```



# Index

calculateGalMediana

Stud, [8](#)

calculateGalVidurkis

Stud, [8](#)

getPav

Stud, [8](#)

getVar

Stud, [9](#)

Include/functionsCallsVector.h, [11](#)

Include/meinelib.h, [11](#)

Include/studentas.h, [12](#)

Include/zmogus.h, [14](#)

setPav

Stud, [9](#)

setVar

Stud, [9](#)

Stud, [7](#)

calculateGalMediana, [8](#)

calculateGalVidurkis, [8](#)

getPav, [8](#)

getVar, [9](#)

setPav, [9](#)

setVar, [9](#)

Zmogus, [9](#)