# My Project

Generated by Doxygen 1.10.0

# Chapter 1

# Hierarchical Index

## 1.1 Class Hierarchy

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

# Chapter 2

# Class Index

## 2.1 Class List

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

# Chapter 3

# File Index

## 3.1 File List

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

# Chapter 4

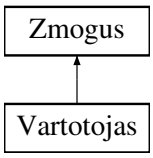
# Class Documentation

## 4.1 Vartotojas Class Reference

atstovauja vartotoja

`#include <funkcijos.h>`

Inheritance diagram for Vartotojas:




### Public Member Functions

- **Vartotojas** ()

  *Vartotojas klasės konstruktorius su nulinėmis reikšmėmis.*
- **Vartotojas** (const string vardas, const string pavarde, const vector< int > &nd, int egz, double vid, double gal, double med, double galmed, double galvid)

  *Vartotojas klasės konstruktorius su parametrais.*
- ~**Vartotojas** ()

  *Vartotojas klasės destruktorius.*
- **Vartotojas** (const Vartotojas &other)

  *kopijavimo konstruktorius.*
- Vartotojas & **operator=** (const Vartotojas &other)

  *copy assignment operatorius.*
- **Vartotojas** (Vartotojas &&other) noexcept

  *move (perkėlimo) konstruktorius.*
- Vartotojas & **operator=** (Vartotojas &&other) noexcept

  *move assignment operatorius.*
- void **setPaz** (int paz)

  *Nustato namų darbų pažymius.*
- void **setVid** (double vidurkis)

  *Nustato namų darbų pažymių vidurkį.*

- void **setMed** (double med)

  *Nustato namų darbų pažymių medianą.*
- void **setEgz** (int egz)

  *Nustato egzamino rezultatą.*
- void **setGal** (double gal)

  *Nustato galutinį pažymį.*
- void **setGalvid** (double galv)

  *Nustato galutinį pažymį su namų darbų vidurkiu.*
- void **setGalmed** (double galm)

  *Nustato galutinį pažymį su namų darbų mediana.*
- void **setVar** (const std::string &vard)

  *Nustato studento varda.*
- void **setPav** (const std::string &pav)

  *Nustato studento varda.*
- const vector< int > & **getPaz** () const

  *Grąžina namų darbų pažymius.*
- int **getEgz** () const

  *Grąžina egzamino rezultatą.*
- double **getVid** () const

  *Grąžina namų darbų pažymių vidurkį.*
- double **getGal** () const

  *Grąžina galutinį pažymį.*
- double **getMed** () const

  *Grąžina namų darbų medianą.*
- double **getGalmed** () const

  *Grąžina galutinį pažymį su mediana.*
- double **getGalvid** () const

  *Grąžina galutinį pažymį su vidurkiu.*
- string getVar () const override

  *Grąžina vardą.*
- string getPav () const override

  *Grąžina pavardę.*

## Public Member Functions inherited from Zmogus

- **Zmogus** ()=default

  *default Zmogus klasės konstruktorius.*
- **Zmogus** (string vardas, string pavarde)

  *Zmogus klasės konstruktorius su parametrais.*
- virtual void **setVar** (string vard)
- virtual void **setPav** (string pav)

**Private Attributes**

- vector< int > **nd_**

  *Studento namų darbai.*
- int **egz_**

  *Studento egzamino rezultatas.*
- double **vid_**

  *Studento namų darbų pažymių vidurkis.*
- double **gal_**

  *Studento galutinis įvertinimas.*
- double **med_**

  *Studento namų darbų pažymių mediana.*
- double **galmed_**

  *Studento galutinis įvertinimas su mediana.*
- double **galvid_**

  *Studento galutinis įvertinimas su vidurkiu.*

**Friends**

- ostream & **operator**<< (ostream &out, const Vartotojas &vart)

  *Išvesties perdengimo operatoriai.*
- istream & **operator**>> (istream &in, Vartotojas &vart)

  *Įvesties perdengimo operatoriai.*

**Additional Inherited Members**

## Protected Attributes inherited from Zmogus

- string **vardas_**

  *Studento vardas.*
- string **pavarde_**

  *Studento pavardė.*

### 4.1.1 Detailed Description

atstovauja vartotoja

### 4.1.2 Member Function Documentation

#### 4.1.2.1 getPav()

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

Grąžina pavardę.

Implements Zmogus.

#### 4.1.2.2 getVar()

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

Grąžina vardą.

Implements Zmogus.

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

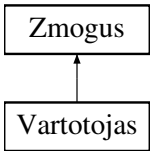
- funkcijos.h

## 4.2 Zmogus Class Reference

atstovauja zmogu su vardu ir pavarde

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

Inheritance diagram for Zmogus:




### Public Member Functions

- **Zmogus** ()=default

  *default Zmogus klasės konstruktorius.*
- **Zmogus** (string vardas, string pavarde)

  *Zmogus klasės konstruktorius su parametrais.*
- virtual void **setVar** (string vard)
- virtual void **setPav** (string pav)
- virtual string getVar () const =0
- virtual string getPav () const =0

### Protected Attributes

- string **vardas_**

  *Studento vardas.*
- string **pavarde_**

  *Studento pavardė.*

### 4.2.1 Detailed Description

atstovauja zmogu su vardu ir pavarde

### 4.2.2 Member Function Documentation

#### 4.2.2.1 getPav()

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

Implemented in Vartotojas.

#### 4.2.2.2 getVar()

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

Implemented in Vartotojas.

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

- funkcijos.h

# Chapter 5

# File Documentation

## 5.1 biblioteka.h File Reference

biblioteku deklaracija

```
#include <iostream>
#include <iomanip>
#include <algorithm>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <fstream>
#include <sstream>
#include <chrono>
#include <cctype>
#include <random>
#include <list>
#include <deque>
#include <utility>
#include <assert.h>
```

### 5.1.1 Detailed Description

biblioteku deklaracija

## 5.2 biblioteka.h

Go to the documentation of this file.

```
00001 //
00002 //  biblioteka.h
00003 //  v2.0
00004 //
00005 //  Created by Kamilė Zobėlaitė on 2024-04-29.
00006 //
00012 #ifndef biblioteka_h
00013 #define biblioteka_h
00014
00015 #include <iostream>
```

```
00016 #include <iomanip>
00017 #include <algorithm>
00018 #include <cstdlib>
00019 #include <ctime>
00020 #include <vector>
00021 #include <fstream>
00022 #include <sstream>
00023 #include <chrono>
00024 #include <cctype>
00025 #include <random>
00026 #include <list>
00027 #include <deque>
00028 #include <utility>
00029 #include <assert.h>
00030
00031
00032 using std::cout;
00033 using std::cin;
00034 using std::endl;
00035 using std::left;
00036 using std::srand;
00037 using std::rand;
00038 using std::vector;
00039 using std::setw;
00040 using std::fixed;
00041 using std::setprecision;
00042 using std::string;
00043 using std::sort;
00044 using std::istringstream;
00045 using std::ifstream;
00046 using std::ofstream;
00047 using std::exception;
00048 using std::numeric_limits;
00049 using std::streamsize;
00050 using std::runtime_error;
00051 using std::cerr;
00052 using std::list;
00053 using std::deque;
00054 using std::move;
00055 using std::ostream;
00056 using std::istream;
00057 using std::accumulate;
00058 #endif /* biblioteka_h */
```

## 5.3 funkcijos.cpp File Reference

Pagalbiniu funciju failo vykdymas.

```
#include "funkcijos.h"
```

### Functions

- double **Vidurkis** (double suma, int nariai)
- double **Mediana** (vector< int > paz, int nariai)
- void **spausdinti** (int rnkts, vector< Vartotojas > &vart, int n)
- double **generavimasPaz** ()
- string **generavimasVard** (int pas)
- string **generavimasPav** (int pas)
- void **skaityti** (vector< Vartotojas > &vart, string pavadinimas, int vm)
- void **rezrikiavimas** (vector< Vartotojas > &vart)
- void **spausdinti_skaitomus_duomenis** (vector< Vartotojas > &vart)
- bool **rikiuotiVarda** (const Vartotojas &a, const Vartotojas &b)
- bool **rikiuotiPavarde** (const Vartotojas &a, const Vartotojas &b)
- bool **rikiuotiVid** (const Vartotojas &a, const Vartotojas &b)
- bool **rikiuotiMed** (const Vartotojas &a, const Vartotojas &b)
- bool **arZodis** (string tekstas)

- bool **arSveikasisSk** (string tekstas)
- void **FailuGeneravimas** (int studSk)
- void **RusiavimasDviGrupes** (vector< Vartotojas > &vart, vector< Vartotojas > &vargsai, vector< Vartotojas > &laimingi, int vm)
- void **spausdintiLaimingiVargsai** (vector< Vartotojas > &vargsai, vector< Vartotojas > &laimingi, int vm)
- void **RusiavimasDviGrupes2** (vector< Vartotojas > &vart, vector< Vartotojas > &vargsai, int vm)
- void **RusiavimasDviGrupes3** (vector< Vartotojas > &vart, vector< Vartotojas > &vargsai, int vm)
- void **testas** ()

### 5.3.1 Detailed Description

Pagalbiniu funciju failo vykdymas.

## 5.4 funkcijos.h File Reference

Zmogus ir Vartotojas klases deklaracija ir funkciju reiksmiu priskyrimas.

```
#include "biblioteka.h"
```

#### Classes

- class Zmogus

  *atstovauja zmogu su vardu ir pavarde*
- class Vartotojas

  *atstovauja vartotoja*

#### Functions

- double **Vidurkis** (double suma, int nariai)
- double **Mediana** (vector< int > paz, int nariai)
- void **spausdinti** (int rnkts, vector< Vartotojas > &vart, int n)
- double **generavimasPaz** ()
- string **generavimasVard** (int pas)
- string **generavimasPav** (int pas)
- void **skaityti** (vector< Vartotojas > &vart, string pavadinimas, int vm)
- void **rezrikiavimas** (vector< Vartotojas > &vart)
- void **spausdinti_skaitomus_duomenis** (vector< Vartotojas > &vart)
- bool **rikiuotiVarda** (const Vartotojas &a, const Vartotojas &b)
- bool **rikiuotiPavarde** (const Vartotojas &a, const Vartotojas &b)
- bool **rikiuotiVid** (const Vartotojas &a, const Vartotojas &b)
- bool **rikiuotiMed** (const Vartotojas &a, const Vartotojas &b)
- bool **arZodis** (string tekstas)
- bool **arSveikasisSk** (string tekstas)
- void **FailuGeneravimas** (int studSk)
- void **RusiavimasDviGrupes** (vector< Vartotojas > &vart, vector< Vartotojas > &vargsai, vector< Vartotojas > &laimingi, int vm)
- void **spausdintiLaimingiVargsai** (vector< Vartotojas > &vargsai, vector< Vartotojas > &laimingi, int vm)
- void **RusiavimasDviGrupes2** (vector< Vartotojas > &vart, vector< Vartotojas > &vargsai, int vm)
- void **RusiavimasDviGrupes3** (vector< Vartotojas > &vart, vector< Vartotojas > &vargsai, int vm)
- void **testas** ()

### 5.4.1 Detailed Description

Zmogus ir Vartotojas klases deklaracija ir funkciju reiksmiu priskyrimas.

## 5.5 funkcijos.h

Go to the documentation of this file.

```
00001 //
00002 //  funkcijos.h
00003 //  v2.0
00004 //
00005 //  Created by Kamilė Zobėlaitė on 2024-04-29.
00006 //
00007
00012 #ifndef funkcijos_h
00013 #define funkcijos_h
00014
00015 #include "biblioteka.h"
00020 class Zmogus
00021 {
00022 protected:
00023
00024     string vardas_;
00025     string pavarde_;
00026 public:
00030     Zmogus() = default;
00034     Zmogus(string vardas, string pavarde)
00035     : vardas_(vardas), pavarde_(pavarde) {}
00036     virtual ~Zmogus() {}
00037    // virtual void  kazkas() const = 0;
00038     virtual void setVar (string vard) {
00039         vardas_=vard;
00040     }
00041     virtual void setPav (string pav) {
00042         pavarde_=pav;
00043     }
00044
00045     virtual string getVar() const = 0;
00046     virtual string getPav() const = 0;
00047 };
00052 class Vartotojas : public Zmogus
00053 {
00054 private:
00055     vector<int> nd_;
00056     int egz_;
00057     double vid_;
00058     double gal_;
00059     double med_;
00060     double galmed_;
00061     double galvid_;
00062 public:
00066     Vartotojas() : vid_(0.0), gal_(0.0), med_(0.0), galmed_(0.0), galvid_(0.0) {}
00070     Vartotojas(const string vardas, const string pavarde, const vector<int>& nd, int egz, double vid,
      double gal, double med, double galmed, double galvid)
00071     : Zmogus(vardas, pavarde), nd_(nd), egz_(egz), vid_(vid), gal_(gal), med_(med), galmed_(galmed),
      galvid_(galvid) {}
00072     // destruktorius
00076     ~Vartotojas() {
00077         //cout « "Objektas sunaikintas" « endl;
00078         nd_.clear();
00079         }
00080     // copy konstruktorius
00083     Vartotojas(const Vartotojas& other)
00084     : Zmogus(other.getVar(), other.getPav()), nd_(other.nd_), egz_(other.egz_), vid_(other.vid_),
      gal_(other.gal_), med_(other.med_), galmed_(other.galmed_), galvid_(other.galvid_) {}
00085     //void kazkas () const override{}
00086
00087     // Copy assignment operatorius
00090         Vartotojas& operator=(const Vartotojas& other) {
00091             if (this != &other) {
00092                 Zmogus::setVar(other.getVar());
00093                 Zmogus::setPav(other.getPav());
00094                 nd_ = other.nd_;
00095                 egz_ = other.egz_;
00096                 vid_ = other.vid_;
00097                 gal_ = other.gal_;
00098                 med_ = other.med_;
00099                 galmed_ = other.galmed_;
00100                 galvid_ = other.galvid_;
```

```
00101             }
00102             return *this;
00103         }
00104     // move konstruktorius
00107     Vartotojas(Vartotojas&& other) noexcept
00108     : Zmogus(std::move(other.vardas_), std::move(other.pavarde_)), nd_(std::move(other.nd_)),
      egz_(other.egz_), vid_(other.vid_), gal_(other.gal_), med_(other.med_), galmed_(other.galmed_),
      galvid_(other.galvid_) {
00109         other.egz_ = 0;
00110         other.vid_ = 0.0;
00111         other.gal_ = 0.0;
00112         other.med_ = 0.0;
00113         other.galmed_ = 0.0;
00114         other.galvid_ = 0.0;
00115     }
00116     // move assignment operatorius
00119     Vartotojas& operator=(Vartotojas&& other) noexcept {
00120             if (this != &other) {
00121                 Zmogus::setVar(std::move(other.vardas_));
00122                 Zmogus::setPav(std::move(other.pavarde_));
00123                 nd_ = std::move(other.nd_);
00124                 egz_ = other.egz_;
00125                 vid_ = other.vid_;
00126                 gal_ = other.gal_;
00127                 med_ = other.med_;
00128                 galmed_ = other.galmed_;
00129                 galvid_ = other.galvid_;
00130                 other.egz_ = 0;
00131                 other.vid_ = 0.0;
00132                 other.gal_ = 0.0;
00133                 other.med_ = 0.0;
00134                 other.galmed_ = 0.0;
00135                 other.galvid_ = 0.0;
00136             }
00137             return *this;
00138         }
00141     friend ostream& operator«(ostream& out, const Vartotojas &vart){
00142         out « left « setw(20) « vart.vardas_ « setw(20) « vart.pavarde_ « setw(20) « fixed «
      setprecision(2) « vart.gal_ « endl;
00143         return out;
00144     }
00147     friend istream& operator»(istream& in, Vartotojas &vart){
00148         in » vart.vardas_ » vart.pavarde_;
00149         int paz;
00150         vector <int> pzm;
00151         while(in » paz){
00152             pzm.push_back(paz);
00153         }
00154         if (!pzm.empty()) {
00155                 vart.egz_ = pzm.back();
00156                 pzm.pop_back();
00157                 }
00158         vart.nd_=pzm;
00159         return in;
00160     }
00163     void setPaz(int paz){
00164         nd_.push_back(paz);
00165     }
00168     void setVid(double vidurkis){
00169         vid_=vidurkis;
00170     }
00173     void setMed(double med){
00174         med_=med;
00175     }
00178     void setEgz (int egz){
00179         egz_ = egz;
00180     }
00183     void setGal(double gal){
00184         gal_ = gal;
00185     }
00188     void setGalvid(double galv){
00189         galvid_=galv;
00190     }
00193     void setGalmed(double galm){
00194         galmed_=galm;
00195     }
00198     void setVar(const std::string& vard) { Zmogus::setVar(vard); }
00202     void setPav(const std::string& pav) { Zmogus::setPav(pav);
00203     }
00206     const vector<int>& getPaz() const { return nd_; }
00209     int getEgz() const { return egz_; }
00212     double getVid() const { return vid_; }
00215     double getGal() const { return gal_; }
00218     double getMed() const { return med_; }
00221     double getGalmed() const { return galmed_; }
00224     double getGalvid() const { return galvid_; }
00227     string getVar() const override { return vardas_; }
```

```
00230     string getPav() const override { return pavarde_; }
00231
00232
00233 };
00234
00235 double Vidurkis(double suma, int nariai);
00236 double Mediana(vector<int> paz, int nariai);
00237 void spausdinti(int rnkts, vector<Vartotojas>& vart, int n);
00238 double generavimasPaz();
00239 string generavimasVard(int pas);
00240 string generavimasPav(int pas);
00241 void skaityti(vector<Vartotojas>& vart, string pavadinimas, int vm);
00242 void rezrikiavimas(vector<Vartotojas>& vart);
00243 void spausdinti_skaitomus_duomenis(vector<Vartotojas>& vart);
00244 bool rikiuotiVarda(const Vartotojas &a, const Vartotojas &b);
00245 bool rikiuotiPavarde(const Vartotojas &a, const Vartotojas &b);
00246 bool rikiuotiVid(const Vartotojas &a, const Vartotojas &b);
00247 bool rikiuotiMed(const Vartotojas &a, const Vartotojas &b);
00248 bool arZodis(string tekstas);
00249 bool arSveikasisSk(string tekstas);
00250 void FailuGeneravimas (int studSk);
00251 void RusiavimasDviGrupes(vector<Vartotojas>& vart, vector<Vartotojas>& vargsai, vector<Vartotojas>&
     laimingi, int vm);
00252 void spausdintiLaimingiVargsai (vector<Vartotojas>& vargsai, vector<Vartotojas>& laimingi, int vm);
00253 void RusiavimasDviGrupes2(vector<Vartotojas>& vart, vector<Vartotojas>& vargsai, int vm);
00254 void RusiavimasDviGrupes3(vector<Vartotojas>& vart, vector<Vartotojas>& vargsai, int vm);
00255 void testas();
00256
00257
00258 #endif /* funkcijos_h */
```

## 5.6 v2.cpp File Reference

Pagrindinio failo vykdymas.

```
#include "funkcijos.h"
```

### Functions

- int **main** ()

### 5.6.1 Detailed Description

Pagrindinio failo vykdymas.

# Index