OOP Pažymio skaičiavimas

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

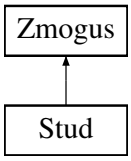
Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 Stud Class Reference

Inheritance diagram for Stud:



Public Member Functions

- **Stud** (vector< int > &namuDarbai, string &vardas, string &pavarde, int egzaminas, double gal, int ndcount)
- **Stud** (const Stud &other)
- Stud & **operator=** (const Stud &other)
- **Stud** (Stud &&other) noexcept
- Stud & **operator=** (Stud &&other) noexcept
- vector< int > **getNamuDarbai** () const
- string **getVardas** () const
- string **getPavarde** () const
- int **getEgzaminas** () const
- double **getGal** () const
- int **getNdcount** () const
- void **setNamuDarbai** (const vector< int > &namuDarbai)
- void **setVardas** (const string &vardas)
- void **setPavarde** (const string &pavarde)
- void **setEgzaminas** (int egzaminas)
- void **setGal** (double gal)
- void **setNdcount** (int ndcount)
- void **addND** (int namuDarbai)
- void **clearND** ()

Public Member Functions inherited from Zmogus

- **Zmogus** (const std::string &vardas, const std::string &pavarde)

Private Attributes

- int **egzaminas_**
- int **ndcount_**
- double **gal_**
- vector< int > **namuDarbai_**

Friends

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

Additional Inherited Members

Public Attributes inherited from Zmogus

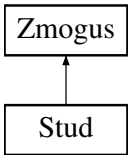
- std::string **vardas_**
- std::string **pavarde_**

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

- vector/student.h

4.2 Zmogus Class Reference

Inheritance diagram for Zmogus:



Public Member Functions

- **Zmogus** (const std::string &vardas, const std::string &pavarde)

Public Attributes

- std::string **vardas_**
- std::string **pavarde_**

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

- vector/zmogus.h

Chapter 5

File Documentation

5.1 funkcijos.h

```
00001 #ifndef FUNKCIJOS_H
00002 #define FUNKCIJOS_H
00003
00004 #include "student.h"
00005
00006 bool rusiavimas(const Stud& a, const Stud& b, char metPas);
00007 double vidurkis(Stud&);
00008 double mediana(Stud&);
00009 void isFailo(const std::string& failPav, std::vector<Stud>& studentai, int dyd);
00010 void failuGen(const std::string& failPav, int numRecords);
00011 void rusiavimasGen(const std::string& failPav, std::vector<Stud>& studentai, vector<Stud>&
      luzeriukai);
00012 void ranka(std::vector<Stud>& studentai);
00013 bool rusiavimasV(const Stud& a, const Stud& b);
00014 bool rusiavimasP(const Stud& a, const Stud& b);
00015 bool rusiavimasG(const Stud& a, const Stud& b);
00016 void testai();
00017 void isvedimas_i_ekrana(const vector<Stud>& luzeriukai, const vector<Stud>& studentai);
00018 void isvedimas_i_faila(const vector<Stud>& luzeriukai, const vector<Stud>& studentai, const string&
      failPav);
00019
00020 #endif
```

5.2 student.h

```
00001 #ifndef STUDENT_H
00002 #define STUDENT_H
00003
00004 #include <iostream>
00005 #include <string>
00006 #include <vector>
00007 #include <iomanip>
00008 #include "zmogus.h"
00009
00010 using namespace std;
00011
00012 class Stud : public Zmogus{
00013 private:
00014   //  string vardas_, pavarde_;
00015     int egzaminas_, ndcount_;
00016     double gal_;
00017     vector<int> namuDarbai_;
00018 public:
00019     Stud() : egzaminas_(0), gal_(0), ndcount_(0) { }  // default konstruktorius
00020     Stud(vector <int>& namuDarbai, string& vardas, string& pavarde, int egzaminas, double gal, int
      ndcount)
00021         : namuDarbai_(namuDarbai), Zmogus(vardas, pavarde), egzaminas_(egzaminas), gal_(gal),
      ndcount_(ndcount) {}
00022
00023     ~Stud() {namuDarbai_.clear(); vardas_.clear(), pavarde_.clear();}
00024
00025     // Copy constructor
00026     Stud(const Stud& other)
```

```
00027         : Zmogus(other.vardas_, other.pavarde_),  egzaminas_(other.egzaminas_),  gal_(other.gal_),
      namuDarbai_(other.namuDarbai_), ndcount_(other.ndcount_) {}
00028
00029     // Copy assignment operator
00030     Stud& operator=(const Stud& other) {
00031         if (this != &other) {
00032             vardas_ = other.vardas_;
00033             pavarde_ = other.pavarde_;
00034             egzaminas_ = other.egzaminas_;
00035             gal_ = other.gal_;
00036             namuDarbai_ = other.namuDarbai_;
00037             ndcount_ = other.ndcount_;
00038         }
00039         return *this;
00040     }
00041
00042     // Move constructor
00043     Stud(Stud&& other) noexcept
00044         : Zmogus(move(other.vardas_), move(other.pavarde_)), egzaminas_(other.egzaminas_),
      gal_(other.gal_), namuDarbai_(move(other.namuDarbai_)), ndcount_(move(other.ndcount_)){
00045         other.vardas_.clear(); other.pavarde_.clear(); other.ndcount_ = 0;  other.egzaminas_ = 0;
      other.gal_ = 0; other.namuDarbai_.clear();}// clearint
00046
00047
00048     // Move assignment operator
00049     Stud& operator=(Stud&& other) noexcept {
00050         if (this != &other) {
00051             vardas_ = move(other.vardas_);
00052             pavarde_ = move(other.pavarde_);
00053             egzaminas_ = move(other.egzaminas_);
00054             gal_ = move(other.gal_);
00055             namuDarbai_ = move(other.namuDarbai_);
00056             ndcount_ = move(other.ndcount_);
00057         }
00058         return *this;
00059     }
00060
00061
00062
00063     // getteriai
00064     vector <int> getNamuDarbai() const {return namuDarbai_;}
00065     string getVardas() const {return vardas_;}
00066     string getPavarde() const {return pavarde_;}
00067     int getEgzaminas() const {return egzaminas_;}
00068     double getGal() const {return gal_;}
00069     int getNdcount() const {return ndcount_;}
00070
00071     //setteriai
00072     void setNamuDarbai(const vector <int>& namuDarbai) {namuDarbai_ = namuDarbai;}
00073     void setVardas(const string& vardas) {vardas_ = vardas;}
00074     void setPavarde(const string& pavarde) {pavarde_ = pavarde;}
00075     void setEgzaminas (int egzaminas) {egzaminas_ = egzaminas;}
00076     void setGal (double gal) {gal_ = gal;}
00077     void setNdcount (int ndcount) {ndcount_ = ndcount;}
00078
00079     //kiti
00080     void addND(int namuDarbai) { namuDarbai_.push_back(namuDarbai); }
00081     void clearND() { namuDarbai_.clear(); }
00082
00083   // Input Operator
00084 friend std::istream& operator»(std::istream& is, Stud& stud) {
00085     is » stud.vardas_ » stud.pavarde_;
00086     stud.namuDarbai_.clear();
00087     int balas;
00088     for (int i = 0; i < stud.getNdcount(); ++i) {
00089         is » balas;
00090         stud.namuDarbai_.push_back(balas);
00091     }
00092
00093     is » stud.egzaminas_;
00094     return is;
00095 }
00096
00097 // Output Operator
00098 friend std::ostream& operator«(std::ostream& os, const Stud& stud) {
00099     os « stud.vardas_ « setw(20) « stud.pavarde_ « setw(20) « stud.gal_ « setw(20) « "\n";
00100     return os;
00101 }
00102
00103
00104
00105 };
00106
00107
00108 #endif
```

5.3 zmogus.h

```
00001 #ifndef ZMOGUS_H
00002 #define ZMOGUS_H
00003 #include <string>
00004
00005 class Zmogus {
00006
00007     public:
00008     std::string vardas_;
00009     std::string pavarde_;
00010
00011     Zmogus() : vardas_(""), pavarde_("") {}
00012     Zmogus(const std::string& vardas, const std::string& pavarde)
00013     : vardas_(vardas), pavarde_(pavarde) {}
00014
00015 };
00016
00017 #endif
```

Index