v2.0

2.0

Generated by Doxygen 1.10.0

1 Hierarchical Index

1.1 Class Hierarchy

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

2 Class Index

2.1 Class List

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

3 File Index

3.1 File List

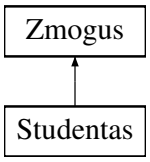
Here is a list of all files with brief descriptions:

4 Class Documentation

4.1 Studentas Class Reference

```
#include <student.h>
```

Inheritance diagram for Studentas:



Public Member Functions

- **Studentas** ()
- **Studentas** (const std::string &vardas_, const std::string &pavarde_, double galutinis_, const std::vector< int > &nd_)
- **Studentas** (const **Studentas** &other)
- **Studentas** & **operator=** (const **Studentas** &other)
- **Studentas** (**Studentas** &&other) noexcept
- **Studentas** & **operator=** (**Studentas** &&other) noexcept
- ~**Studentas** ()
- std::string **getVardas** () const
- std::string **getPavarde** () const
- const std::vector< int > & **getND** () const
- double **getGalutinis** () const
- double **getEgzaminas** ()
- void **setVardas** (const std::string &vardas)
- void **setPavarde** (const std::string &pavarde)
- void **setEgzaminas** (double egzaminas)
- void **setNd** (const std::vector< int > &nd)
- void **setGalutinis** (double galutinis)
- double **calculateGalutinis** (bool useMedian) const

Public Member Functions inherited from Zmogus

- virtual ~**Zmogus** ()=default

4.1.1 Constructor & Destructor Documentation

Studentas() [1/4]

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

Studentas() [2/4]

```
Studentas::Studentas (
            const std::string & vardas_,
            const std::string & pavarde_,
            double galutinis_,
            const std::vector< int > & nd_ )
```

Studentas() [3/4]

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

Studentas() [4/4]

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

~Studentas()

```
Studentas::~Studentas ( )  [default]
```

4.1.2 Member Function Documentation

calculateGalutinis()

```
double Studentas::calculateGalutinis (
            bool useMedian ) const
```

getEgzaminas()

```
double Studentas::getEgzaminas ( )  [inline], [virtual]
```

Implements **Zmogus** (p. 6).

getGalutinis()

```
double Studentas::getGalutinis ( ) const  [inline], [virtual]
```

Implements **Zmogus** (p. 6).

getND()

```
const std::vector< int > & Studentas::getND ( ) const  [inline], [virtual]
```

Implements **Zmogus** (p. 6).

getPavarde()

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

Implements **Zmogus** (p. 6).

getVardas()

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

Implements **Zmogus** (p. 6).

operator=() [1/2]

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

operator=() [2/2]

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

setEgzaminas()

```
void Studentas::setEgzaminas (
            double egzaminas )  [inline]
```

setGalutinis()

```
void Studentas::setGalutinis (
            double galutinis )  [inline]
```

setNd()

```
void Studentas::setNd (
            const std::vector< int > & nd )  [inline]
```

setPavarde()

```
void Studentas::setPavarde (
            const std::string & pavarde )  [inline]
```

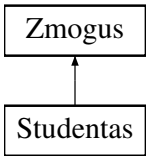
setVardas()

```
void Studentas::setVardas (
            const std::string & vardas )  [inline]
```

4.2 Zmogus Class Reference

```
#include <student.h>
```

Inheritance diagram for Zmogus:



Public Member Functions

- virtual ~**Zmogus** ()=default
- virtual std::string **getVardas** () const =0
- virtual std::string **getPavarde** () const =0
- virtual const std::vector< int > & **getND** () const =0
- virtual double **getGalutinis** () const =0
- virtual double **getEgzaminas** ()=0

4.2.1 Constructor & Destructor Documentation

~Zmogus()

```
virtual Zmogus::~Zmogus ( )  [virtual], [default]
```

4.2.2 Member Function Documentation

getEgzaminas()

```
virtual double Zmogus::getEgzaminas ( )  [pure virtual]
```

Implemented in **Studentas** (p. 4).

getGalutinis()

```
virtual double Zmogus::getGalutinis ( ) const  [pure virtual]
```

Implemented in **Studentas** (p. 4).

getND()

```
virtual const std::vector< int > & Zmogus::getND ( ) const  [pure virtual]
```

Implemented in **Studentas** (p. 4).

getPavarde()

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

Implemented in **Studentas** (p. 4).

getVardas()

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

Implemented in **Studentas** (p. 4).

5 File Documentation

5.1 main.cpp File Reference

```
#include "menu.h"
#include "student.h"
```

Functions

- void **testConstructors** ()
- int **main** ()

5.1.1 Function Documentation

main()

```
int main ( )
```

testConstructors()

```
void testConstructors ( )
```

5.2 menu.cpp File Reference

```
#include "menu.h"
#include "student.h"
```

Functions

- void **Menu_execute** ()

5.2.1 Function Documentation

Menu_execute()

```
void Menu_execute ( )
```

5.3 menu.h File Reference

```
#include <string>
#include <vector>
#include "student.h"
```

Functions

- void **Menu_execute** ()

5.3.1 Function Documentation

Menu_execute()

```
void Menu_execute ( )
```

5.4 menu.h

Go to the documentation of this file.

```
00001 #ifndef MENU_H
00002 #define MENU_H
00003
00004 #include <string>
00005 #include <vector>
00006 #include "student.h"
00007
00008 void Menu_execute();
00009
00010 #endif // MAIN_LOGIC_H
00011 // Path: v0.3/main_logic.cpp
```

5.5 student.cpp File Reference

```
#include "student.h"
```

Functions

- std::ostream & **operator**<< (std::ostream &os, const **Studentas** &student)
- std::istream & **operator**>> (std::istream &is, **Studentas** &student)
- bool **isNumber** (const string &str)
- double **useMediana** (const vector< int > &grades)
- void **Ivedimas** (vector< **Studentas** > &stud, bool randomNames, bool randomGrades, bool studentCount)
- void **Spausdinimas** (const vector< **Studentas** > &stud, bool Mediana)
- void **sortStudents** (vector< **Studentas** > &students, const string &sortBy)
- void **Pasirinkimai** (vector< **Studentas** > &students)
- void **Generacija** (int Pas)

Variables

- const int **NUM_NAMES** = 10
- vector< string > **vardai** = { "Jonas", "Petras", "Algis", "Marius", "Gintaras", "Tomas", "Lukas", "Simas", "Gabrielius", "Olegas" }
- vector< string > **pavardes** = { "Kelmutis", "Kelmutaite", "Dangavicius", "Pieliauskas", "Lukavicius", "Simonavicius", "Skaudavicius", "Juzenas", "Darbavicius", "Stankevicius" }

5.5.1 Function Documentation

Generacija()

```
void Generacija (
            int Pas )
```

isNumber()

```
bool isNumber (
            const string & str )
```

Ivedimas()

```
void Ivedimas (
            vector<  Studentas > & stud,
            bool randomNames,
            bool randomGrades,
            bool studentCount )
```

operator<<()

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

operator>>()

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

Pasirinkimai()

```
void Pasirinkimai (
            vector<  Studentas > & students )
```

sortStudents()

```
void sortStudents (
            vector<  Studentas > & students,
            const string & sortBy )
```

Spausdinimas()

```
void Spausdinimas (
            const vector<  Studentas > & stud,
            bool Mediana )
```

useMediana()

```
double useMediana (
            const vector< int > & grades )
```

5.5.2 Variable Documentation

NUM_NAMES

```
const int NUM_NAMES = 10
```

pavardes

```
vector<string> pavardes = { "Kelmutis", "Kelmutaite", "Dangavicius", "Pieliauskas", "Lukavicius",
"Simonavicius", "Skaudavicius", "Juzenas", "Darbavicius", "Stankevicius" }
```

vardai

```
vector<string> vardai = { "Jonas", "Petras", "Algis", "Marius", "Gintaras", "Tomas", "Lukas",
"Simas", "Gabrielius", "Olegas" }
```

5.6 student.h File Reference

```
#include <iostream>
#include <string>
#include <iomanip>
#include <algorithm>
#include <vector>
#include <numeric>
#include <ctime>
#include <fstream>
#include <sstream>
#include <cstdlib>
#include <cctype>
#include <cmath>
#include <chrono>
#include <random>
#include <iterator>
#include <functional>
#include <stdexcept>
```

Classes

- class **Zmogus**
- class **Studentas**

Functions

- bool **compare** (const **Studentas** &, const **Studentas** &)
- bool **comparePagalPavarde** (const **Studentas** &, const **Studentas** &)
- bool **comparePagalEgza** (const **Studentas** &, const **Studentas** &)
- void **Ivedimas** (vector< **Studentas** > &stud, bool randomNames=false, bool randomGrades=false, bool studentCount=false)
- void **Pasirinkimai** (vector< **Studentas** > &students)
- void **Spausdinimas** (const vector< **Studentas** > &students, bool Mediana)
- bool **isNumber** (const string &str)
- double **useMediana** (const vector< int > &grades)
- void **Generacija** (int Pas)

5.6.1 Function Documentation

compare()

```
bool compare (
            const  Studentas & ,
            const  Studentas &  )
```

comparePagalEgza()

```
bool comparePagalEgza (
            const  Studentas & ,
            const  Studentas &  )
```

comparePagalPavarde()

```
bool comparePagalPavarde (
            const  Studentas & ,
            const  Studentas &  )
```

Generacija()

```
void Generacija (
            int Pas )
```

isNumber()

```
bool isNumber (
            const string & str )
```

Ivedimas()

```
void Ivedimas (
             vector<  Studentas > & stud,
             bool randomNames = false,
             bool randomGrades = false,
             bool studentCount = false )
```

Pasirinkimai()

```
void Pasirinkimai (
             vector<  Studentas > & students )
```

Spausdinimas()

```
void Spausdinimas (
             const vector<  Studentas > & students,
             bool Mediana )
```

useMediana()

```
double useMediana (
             const vector< int > & grades )
```

5.7 student.h

Go to the documentation of this file.

```
00001 #ifndef STUDENT_H
00002 #define STUDENT_H
00003
00004 #include <iostream>
00005 #include <string>
00006 #include <iomanip>
00007 #include <algorithm>
00008 #include <vector>
00009 #include <numeric>
00010 #include <ctime>
00011 #include <fstream>
00012 #include <sstream>
00013 #include <cstdlib>
00014 #include <cctype>
00015 #include <cmath>
00016 #include <chrono>
00017 #include <random>
00018 #include <iterator>
00019 #include <functional>
00020 #include <stdexcept>
00021
00022
00023 using namespace std;
00024 using namespace std::chrono;
00025
00026 class Zmogus{
00027     public:
00028     virtual ~Zmogus() = default;
00029
00030     //geteriai
00031     virtual std::string getVardas() const = 0;
00032     virtual std::string getPavarde() const = 0;
00033     virtual const std::vector<int>& getND() const = 0;
00034     virtual double getGalutinis() const = 0;
00035     virtual double getEgzaminas() = 0;
00036 };
```

```
00037
00038 class Studentas: public Zmogus{
00039 private:
00040     std::string vardas_;
00041     std::string pavarde_;
00042     double egzaminas_;
00043     std::vector<int> nd_;
00044     double galutinis_;
00045
00046 public:
00047         // Default constructor
00048         Studentas();
00049
00050         // Constructor with parameters
00051         Studentas(const std::string& vardas_, const std::string& pavarde_, double galutinis_, const
      std::vector<int>& nd_);
00052
00053         // Copy constructor
00054         Studentas(const Studentas& other);
00055
00056         // Copy assignment operator
00057         Studentas& operator=(const Studentas& other);
00058
00059         // Move constructor
00060         Studentas(Studentas&& other) noexcept;
00061
00062         // Move assignment operator
00063         Studentas& operator=(Studentas&& other) noexcept;
00064
00065         // Destructor
00066         ~Studentas();
00067
00068         // Getters
00069         inline std::string getVardas() const { return vardas_; }
00070         inline std::string getPavarde() const { return pavarde_; }
00071         const std::vector<int>& getND() const { return nd_; }
00072         double getGalutinis() const { return galutinis_; }
00073         double getEgzaminas() { return egzaminas_; }
00074
00075         // Setters
00076         void setVardas(const std::string& vardas) { vardas_ = vardas; }
00077         void setPavarde(const std::string& pavarde) { pavarde_ = pavarde; }
00078         void setEgzaminas(double egzaminas) { egzaminas_ = egzaminas; }
00079         void setNd(const std::vector<int>& nd) { nd_ = nd; }
00080         void setGalutinis(double galutinis) { this->galutinis_ = galutinis;}
00081         // Calculate the final grade
00082         double calculateGalutinis(bool useMedian) const; // Add the missing implementation
00083 };
00084
00085 // Comparison functions for sorting
00086 bool compare(const Studentas&, const Studentas&);
00087 bool comparePagalPavarde(const Studentas&, const Studentas&);
00088 bool comparePagalEgza(const Studentas&, const Studentas&);
00089
00090 // Input function
00091 void Ivedimas(vector<Studentas>& stud, bool randomNames = false, bool randomGrades = false, bool
      studentCount = false);
00092
00093 // Menu function
00094 void Pasirinkimai(vector<Studentas>& students);
00095
00096 // Output function
00097 void Spausdinimas(const vector<Studentas>& students, bool Mediana);
00098
00099 // Check if a string is a number
00100 bool isNumber(const string& str);
00101
00102 // Calculate the final grade using median
00103 double useMediana(const vector<int>& grades);
00104
00105 // Generate random data
00106 void Generacija(int Pas);
00107
00108 #endif // STUDENT_H
00109 // Path: v0.3/v0.3.cpp
```

Index