OBJEKTINIS

Generated by Doxygen 1.13.2

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 Human Class Reference

Inheritance diagram for Human:

4.2 student Class Reference

Inheritance diagram for student:

Collaboration diagram for student:

Public Member Functions

- **student** (const student &other) noexcept
- student & **operator=** (const student &other) noexcept
- **student** (student &&other) noexcept
- student & **operator=** (student &&other) noexcept
- **student** (std::string v, std::string p, std::vector< float > pazymiai, float egz) noexcept
- void **setPazymiai** (std::vector< float > paz) noexcept
- void **setEgzaminoRezultatas** (float egz) noexcept
- void **setGalutinisV** (float V) noexcept
- void **setGalutinisM** (float M) noexcept
- const std::vector< float > & **getPazymiai** () const
- float **getEgzaminoRezultatas** () const
- float **getGalutinisV** () const
- float **getGalutinisM** () const
- float **skaiciuotiVid** () const
- float **skaiciuotiMed** () const
- void skaiciuotiGalutini (char galutinioBudas) override
- void **addPazymys** (float pazymys)

Public Member Functions inherited from Human

- **Human** (const std::string &v, const std::string &p)
- const std::string & **getVardas** () const
- const std::string & **getPavarde** () const
- void **setVardas** (const std::string &v)
- void **setPavarde** (const std::string &p)

Friends

- std::ostream & **operator**<< (std::ostream &os, const student &studentas)
- std::istream & **operator**>> (std::istream &in, student &studentas)

Additional Inherited Members

Protected Attributes inherited from Human

- std::string **vardas**
- std::string **pavarde**

4.2.1 Member Function Documentation

4.2.1.1 skaiciuotiGalutini()

```
void student::skaiciuotiGalutini (
            char galutinioBudas)  [inline], [override], [virtual]
```

Implements Human.

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

- C:/Users/Lenovo/Desktop/2-uzduotis/vector/include/student.h
- C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/student.cpp

Chapter 5

File Documentation

5.1 C:/Users/Lenovo/Desktop/2-uzduotis/vector/include/functions.h File Reference

this file contains functions declarations

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

Include dependency graph for functions.h: This graph shows which files directly or indirectly include this file:

Functions

- void **rusiuotiOutput** (std::vector< student > &grupe, char rusiavimoBudas, char galutinioBudas)
- void **spausdinimasTerminale** (const std::vector< student > &grupe, char galutinioBudas)
- void **spausdinimasFaile** (const std::vector< student > &grupe, char galutinioBudas)
- void **generuotiFaila** (int pKiekis, int studentuKiekis, const std::string &failoPavadinimas)
- void **spausdinimas** (char spausBudas, char rusiavimoBudas, char galutinioBudas, std::vector< student > &grupe)
- void **sortedStudentSpausdinimas** (std::string lowGradeFailas, std::string highGradeFailas, std::vector< student > &nepazangus, std::vector< student > &normalus, char galutinioBudas)
- void **nuskaitytiGeneruotusFailus** (const std::string &failoPavadinimas, std::vector< student > &grupe, int pKiekis, char galutinioBudas)
- void **skirstytiStudentus** (std::vector< student > &grupe, std::vector< student > &nepazangus, std::vector< student > &normalus, char galutinioBudas)
- void **pirmas** (std::vector< student > &grupe, char spausBudas, char rusiavimoBudas, char galutinioBudas, int pKiekis)
- void **antras** (std::vector< student > &grupe, char spausBudas, char rusiavimoBudas, char galutinioBudas, int pKiekis)
- void **trecias** (std::vector< student > &grupe, const std::vector< std::string > &vardai, const std::vector< std::string > &pavardes, char spausBudas, char rusiavimoBudas, char galutinioBudas, int pKiekis)
- void **ketvirtas** (std::vector< student > &grupe, int pKiekis, char galutinioBudas)
- void **penktas** (int pKiekis)
- void **sestas** (std::vector< student > &grupe, std::vector< student > &testGrupe, std::vector< student > &nepazangus, std::vector< student > &normalus, char galutinioBudas, char rusiavimoBudas, int pKiekis)
- void **septintas** (char galutinioBudas)

5.1.1 Detailed Description

this file contains functions declarations

Author

Narbas

Version

v2.0

Date

2025-05-07

Copyright

Copyright (c) 2025

5.2 functions.h

Go to the documentation of this file.

```


#ifndef FUNCTIONS_H
#define FUNCTIONS_H

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

void rusiuotiOutput(std::vector<student>& grupe, char rusiavimoBudas, char galutinioBudas);
void spausdinimasTerminale(const std::vector<student>& grupe, char galutinioBudas);
void spausdinimasFaile(const std::vector<student>& grupe, char galutinioBudas);
void generuotiFaila(int pKiekis, int studentuKiekis, const std::string& failoPavadinimas);
void spausdinimas(char spausBudas, char rusiavimoBudas, char galutinioBudas, std::vector<student>&grupe);
void sortedStudentSpausdinimas(std:: string lowGradeFailas, std::string highGradeFailas, std::vector<student>&nepazangus, std::vector<student>&normalus, char galutinioBudas);
void nuskaitytiGeneruotusFailus(const std::string& failoPavadinimas, std::vector<student> & grupe, int pKiekis, char galutinioBudas);
void skirstytiStudentus(std::vector<student> & grupe, std::vector<student>& nepazangus, std::vector<student>& normalus, char galutinioBudas);

void pirmas(std::vector<student>& grupe, char spausBudas, char rusiavimoBudas, char galutinioBudas, int pKiekis);
void antras(std::vector<student>& grupe, char spausBudas, char rusiavimoBudas, char galutinioBudas, int pKiekis);
void trecias(std::vector<student>& grupe, const std::vector<std::string>& vardai, const std::vector<std::string>& pavardes, char spausBudas, char rusiavimoBudas, char galutinioBudas, int pKiekis);
void ketvirtas(std::vector<student>& grupe, int pKiekis, char galutinioBudas);
void penktas(int pKiekis);
void sestas(std::vector<student>& grupe, std::vector<student>& testGrupe, std::vector<student>& nepazangus, std::vector<student>& normalus, char galutinioBudas, char rusiavimoBudas, int pKiekis);
void septintas(char galutinioBudas);
#endif
```

5.3 C:/Users/Lenovo/Desktop/2-uzduotis/vector/include/student.h File Reference

this file contains Human and Student classes

```
#include <vector>
#include <string>
#include <iostream>
```

Include dependency graph for student.h: This graph shows which files directly or indirectly include this file:

Classes

- class Human
- class student

Functions

- std::ostream & **operator**<< (std::ostream &os, const student &studentas)
- std::istream & **operator**>> (std::istream &in, student &studentas)

5.3.1 Detailed Description

this file contains Human and Student classes

Author

Narbas

Version

v2.0

Date

2025-05-07

Copyright

Copyright (c) 2025

5.4 student.h

Go to the documentation of this file.

```
#ifndef STUDENT_H
#define STUDENT_H


#include <vector>
#include <string>
#include <iostream>

class Human{
    protected:

        std::string vardas;
        std::string pavarde;

    public:

    Human() : vardas(""), pavarde("") {}
    Human(const std::string& v, const std::string& p) : vardas(v), pavarde(p) {}

    //getters
    const std::string& getVardas() const { return vardas; }
    const std::string& getPavarde() const { return pavarde; }
    //setters
    void setVardas(const std::string& v)   { vardas = v; }
    void setPavarde(const std::string& p)  { pavarde = p; }

    virtual ~Human() = default;

    virtual void skaiciuotiGalutini(char galutinioBudas) = 0;

};

class student : public Human{

    private:

        std::vector<float> pazymiai{};
        float egzaminoRezultatas = 0.0f;
        float galutinisM = 0.0f;
        float galutinisV = 0.0f;

    public:

    student() = default;

    //rule of 5------------------------------------------------------
    //copy
    student(const student &other) noexcept : Human(other.getVardas(), other.getPavarde()), pazymiai(other.pazymiai),
                                    egzaminoRezultatas(other.egzaminoRezultatas), galutinisM(other.galutinisM), galutinisV(other.galutinisV){};
    //copy asg
    student& operator=(const student &other) noexcept {
        if(this != &other){
            setVardas(other.getVardas());
            setPavarde(other.getPavarde());
            pazymiai = other.pazymiai;
            egzaminoRezultatas = other.egzaminoRezultatas;
            galutinisM = other.galutinisM;
            galutinisV = other.galutinisV;
        }

        return *this;
    };
    //move
    student(student &&other) noexcept : Human(std::move(other.vardas), std::move(other.pavarde)),
      pazymiai(std::move(other.pazymiai)),
      egzaminoRezultatas(other.egzaminoRezultatas),
      galutinisM(other.galutinisM),
      galutinisV(other.galutinisV) {}
    //move asg
    student& operator=(student &&other) noexcept {
        if(this != &other){
            vardas = std::move(other.vardas);
            pavarde = std::move(other.pavarde);
            pazymiai = std::move(other.pazymiai);
            egzaminoRezultatas = other.egzaminoRezultatas;
            galutinisM = other.galutinisM;
            galutinisV = other.galutinisV;
        }
        return *this;
    };
    // ------------------------------------------------------------

    //isvestis, ivestis overloads
    friend std::ostream& operator « (std::ostream& os, const student& studentas);
    friend std::istream& operator » (std::istream& in, student& studentas);

    //parametrizuotas ctor
    student(std::string v, std::string p, std::vector<float> pazymiai, float egz) noexcept : Human(v, p), pazymiai(std::move(pazymiai)), egzaminoRezultatas(egz) {}

    //setters
    void setPazymiai(std::vector<float> paz) noexcept{ pazymiai = std::move(paz); }
    void setEgzaminoRezultatas(float egz) noexcept { egzaminoRezultatas = egz; }
    void setGalutinisV(float V) noexcept { galutinisV = V; }
    void setGalutinisM(float M) noexcept { galutinisM = M; }

    //getters
    const std::vector<float>& getPazymiai() const { return pazymiai; }
    float getEgzaminoRezultatas() const { return egzaminoRezultatas; }
    float getGalutinisV() const { return galutinisV; }
    float getGalutinisM() const { return galutinisM; }

    //methods
    float skaiciuotiVid() const;
    float skaiciuotiMed() const;
    void skaiciuotiGalutini(char galutinioBudas) override;
    void addPazymys(float pazymys) { pazymiai.push_back(pazymys); }
};

std::ostream& operator«(std::ostream& os, const student& studentas);
std::istream& operator»(std::istream& in, student& studentas);
```

```

#endif
```

5.5 C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/functions.cpp File Reference

this file contains all functions

```
#include "functions.h"
#include <numeric>
#include <algorithm>
#include <iomanip>
#include <fstream>
#include <iostream>
#include <chrono>
#include <sstream>
#include <random>
#include "student.h"
```

Include dependency graph for functions.cpp:

Functions

- void **rusiuotiOutput** (std::vector< student > &grupe, char rusiavimoBudas, char galutinioBudas)
- void **spausdinimasTerminale** (const std::vector< student > &grupe, char galutinioBudas)
- void **spausdinimasFaile** (const std::vector< student > &grupe, char galutinioBudas)
- void **generuotiFaila** (int pKiekis, int studentuKiekis, const std::string &failoPavadinimas)
- void **spausdinimas** (char spausBudas, char rusiavimoBudas, char galutinioBudas, std::vector< student > &grupe)
- void **sortedStudentSpausdinimas** (std::string lowGradeFailas, std::string highGradeFailas, std::vector< student > &nepazangus, std::vector< student > &normalus, char galutinioBudas)
- void **nuskaitytiGeneruotusFailus** (const std::string &failoPavadinimas, std::vector< student > &grupe, int pKiekis, char galutinioBudas)
- void **skirstytiStudentus** (std::vector< student > &grupe, std::vector< student > &nepazangus, std::vector< student > &normalus, char galutinioBudas)
- void **pirmas** (std::vector< student > &grupe, char spausBudas, char rusiavimoBudas, char galutinioBudas, int pKiekis)
- void **antras** (std::vector< student > &grupe, char spausBudas, char rusiavimoBudas, char galutinioBudas, int pKiekis)
- void **trecias** (std::vector< student > &grupe, const std::vector< std::string > &vardai, const std::vector< std::string > &pavardes, char spausBudas, char rusiavimoBudas, char galutinioBudas, int pKiekis)
- void **ketvirtas** (std::vector< student > &grupe, int pKiekis, char galutinioBudas)
- void **penktas** (int pKiekis)
- void **sestas** (std::vector< student > &grupe, std::vector< student > &testGrupe, std::vector< student > &nepazangus, std::vector< student > &normalus, char galutinioBudas, char rusiavimoBudas, int pKiekis)
- void **septintas** (char galutinioBudas)

Variables

- double **programosLaikas**

5.5.1 Detailed Description

this file contains all functions

Author

Narbas

Version

v2.0

Date

2025-05-07

Copyright

Copyright (c) 2025

5.6 C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/main.cpp File Reference

This file handles program's flow.

```
#include <iostream>
#include <vector>
#include <ctime>
#include <limits>
#include <cstdlib>
#include <chrono>
#include <sstream>
#include <fstream>
#include <algorithm>
#include "student.h"
#include "functions.h"
#include <iomanip>
#include "utils.h"
```

Include dependency graph for main.cpp:

Functions

- int **main** ()

Variables

- double **programosLaikas** = 0.0

5.6.1 Detailed Description

This file handles program's flow.

Author

Narbas

Version

v2.0

Date

2025-05-07

Copyright

Copyright (c) 2025

5.7 C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/student.cpp File Reference

this file contains class methods implementation

```
#include "student.h"
#include <algorithm>
#include <numeric>
#include <iomanip>
```

Include dependency graph for student.cpp:

Functions

- std::ostream & **operator**<< (std::ostream &os, const student &studentas)
- std::istream & **operator**>> (std::istream &in, student &studentas)

5.7.1 Detailed Description

this file contains class methods implementation

Author

Narbas

Version

v2.0

Date

2025-05-07

Copyright

Copyright (c) 2025

5.8 C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/utils.h File Reference

this file contains templates

```
#include <iostream>
#include <limits>
#include <string>
```

Include dependency graph for utils.h: This graph shows which files directly or indirectly include this file:

Functions

- template<typename T>
  T **tikrintiInput** (const std::string &prompt, const std::string &klaida)

5.8.1 Detailed Description

this file contains templates

Author

Narbas

Version

v2.0

Date

2025-05-07

Copyright

Copyright (c) 2025

5.9 utils.h

Go to the documentation of this file.

```
#ifndef UTILS_H
#define UTILS_H


#include <iostream>
#include <limits>
#include <string>

template<typename T>
T tikrintiInput(const std::string& prompt, const std::string& klaida) {
    T value;
    while (true) {
        std::cout « prompt;
        std::cin » value;
        if (!std::cin.fail()) {
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
            return value;
        }
        std::cerr « klaida « std::endl;
        std::cin.clear();
        std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
    }
}

#endif
```

Index