

OBJEKTINIS

Generated by Doxygen 1.13.2

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Human Class Reference	7
4.2 student Class Reference	7
4.2.1 Member Function Documentation	8
4.2.1.1 skaiciuotiGalutini()	8
5 File Documentation	9
5.1 C:/Users/Lenovo/Desktop/2-uzduotis/vector/include/functions.h File Reference	9
5.1.1 Detailed Description	10
5.2 functions.h	10
5.3 C:/Users/Lenovo/Desktop/2-uzduotis/vector/include/student.h File Reference	10
5.3.1 Detailed Description	11
5.4 student.h	11
5.5 C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/functions.cpp File Reference	13
5.5.1 Detailed Description	14
5.6 C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/main.cpp File Reference	14
5.6.1 Detailed Description	15
5.7 C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/student.cpp File Reference	15
5.7.1 Detailed Description	15
5.8 C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/utils.h File Reference	16
5.8.1 Detailed Description	16
5.9 utils.h	16
Index	17

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Human	7
student	7

Chapter 2

Class Index

2.1 Class List

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

Human	7
student	7

Chapter 3

File Index

3.1 File List

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

C:/Users/Lenovo/Desktop/2-uzduotis/vector/include/ functions.h	
This file contains functions declarations	9
C:/Users/Lenovo/Desktop/2-uzduotis/vector/include/ student.h	
This file contains Human and Student classes	10
C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/ functions.cpp	
This file contains all functions	13
C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/ main.cpp	
This file handles program's flow	14
C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/ student.cpp	
This file contains class methods implementation	15
C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/ utils.h	
This file contains templates	16

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.](#)

```
00001
00011
00012 #ifndef FUNCTIONS_H
00013 #define FUNCTIONS_H
00014
00015 #include <vector>
00016 #include "student.h"
00017
00018 void rusiuotiOutput(std::vector<student>& grupe, char rusiavimoBudas, char galutinioBudas);
00019 void spausdinimasTerminale(const std::vector<student>& grupe, char galutinioBudas);
00020 void spausdinimasFaile(const std::vector<student>& grupe, char galutinioBudas);
00021 void generuotiFaila(int pKiekis, int studentuKiekis, const std::string& failoPavadinimas);
00022 void spausdinimas(char spausBudas, char rusiavimoBudas, char galutinioBudas,
00023 std::vector<student>&grupe);
00023 void sortedStudentSpausdinimas(std::string lowGradeFailas, std::string highGradeFailas,
00024 std::vector<student>&nepazangus, std::vector<student>&normalus, char galutinioBudas);
00024 void nuskaitytiGeneruotusFailus(const std::string& failoPavadinimas, std::vector<student> & grupe, int
00025 pKiekis, char galutinioBudas);
00025 void skirstytiStudentus(std::vector<student> & grupe, std::vector<student>& nepazangus,
00026 std::vector<student>& normalus, char galutinioBudas);
00026
00027 void pirmas(std::vector<student>& grupe, char spausBudas, char rusiavimoBudas, char galutinioBudas,
00028 int pKiekis);
00028 void antras(std::vector<student>& grupe, char spausBudas, char rusiavimoBudas, char galutinioBudas,
00029 int pKiekis);
00029 void trecias(std::vector<student>& grupe, const std::vector<std::string>& vardai, const
00030 std::vector<std::string>& pavardes, char spausBudas, char rusiavimoBudas, char galutinioBudas, int
00031 pKiekis);
00030 void ketvirtas(std::vector<student>& grupe, int pKiekis, char galutinioBudas);
00031 void penktas(int pKiekis);
00032 void sextas(std::vector<student>& grupe, std::vector<student>& testGrupe, std::vector<student>&
00033 nepazangus, std::vector<student>& normalus, char galutinioBudas, char rusiavimoBudas, int pKiekis);
00033 void septintas(char galutinioBudas);
00034 #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.](#)

```
00001 #ifndef STUDENT_H
00002 #define STUDENT_H
00003
00004
00005 #include <vector>
00006 #include <string>
00007 #include <iostream>
00008
00009 class Human{
00010     protected:
00011
00012         std::string vardas;
00013         std::string pavarde;
00014
00015     public:
00016
00017     Human() : vardas(""), pavarde("") {}
00018     Human(const std::string& v, const std::string& p) : vardas(v), pavarde(p) {}
00019
00020     //getters
00021     const std::string& getVardas() const { return vardas; }
00022     const std::string& getPavarde() const { return pavarde; }
00023     //setters
00024     void setVardas(const std::string& v) { vardas = v; }
00025     void setPavarde(const std::string& p) { pavarde = p; }
00026
00027 }
```

```

00037     virtual ~Human() = default;
00038
00039     virtual void skaiciuotiGalutini(char galutinioBudas) = 0;
00040
00041 };
00042
00043 class student : public Human{
00044
00045     private:
00046
00047         std::vector<float> pazymiai{};
00048         float egzaminoRezultatas = 0.0f;
00049         float galutinisM = 0.0f;
00050         float galutinisV = 0.0f;
00051
00052     public:
00053
00054         student() = default;
00055
00056         //rule of 5-----
00057         //copy
00058         student(const student &other) noexcept : Human(other.getVardas(), other.getPavarde()),
00059             pazymiai(other.pazymiai),
00059             egzaminoRezultatas(other.egzaminoRezultatas),
00060             galutinisM(other.galutinisM), galutinisV(other.galutinisV){};
00061         //copy asg
00062         student& operator=(const student &other) noexcept {
00063             if(this != &other){
00064                 setVardas(other.getVardas());
00065                 setPavarde(other.getPavarde());
00066                 pazymiai = other.pazymiai;
00067                 egzaminoRezultatas = other.egzaminoRezultatas;
00068                 galutinisM = other.galutinisM;
00069                 galutinisV = other.galutinisV;
00070             }
00071             return *this;
00072         };
00073         //move
00074         student(student &&other) noexcept : Human(std::move(other.vardas), std::move(other.pavarde)),
00075             pazymiai(std::move(other.pazymiai)),
00076             egzaminoRezultatas(other.egzaminoRezultatas),
00077             galutinisM(other.galutinisM),
00078             galutinisV(other.galutinisV) {}
00079         //move asg
00080         student& operator=(student &&other) noexcept {
00081             if(this != &other){
00082                 vardas = std::move(other.vardas);
00083                 pavarde = std::move(other.pavarde);
00084                 pazymiai = std::move(other.pazymiai);
00085                 egzaminoRezultatas = other.egzaminoRezultatas;
00086                 galutinisM = other.galutinisM;
00087                 galutinisV = other.galutinisV;
00088             }
00089             return *this;
00090         };
00091         // -----
00092
00093         //isvestis, ivestis overloads
00094         friend std::ostream& operator < (std::ostream& os, const student& studentas);
00095         friend std::istream& operator > (std::istream& in, student& studentas);
00096
00097         //parametrizuotas ctor
00098         student(std::string v, std::string p, std::vector<float> pazymiai, float egz) noexcept : Human(v,
00099             p), pazymiai(std::move(pazymiai)), egzaminoRezultatas(egz) {}
00100
00101         //setters
00102         void setPazymiai(std::vector<float> paz) noexcept { pazymiai = std::move(paz); }
00103         void setEgzaminoRezultatas(float egz) noexcept { egzaminoRezultatas = egz; }
00104         void setGalutinisV(float V) noexcept { galutinisV = V; }
00105         void setGalutinisM(float M) noexcept { galutinisM = M; }
00106
00107         //getters
00108         const std::vector<float>& getPazymiai() const { return pazymiai; }
00109         float getEgzaminoRezultatas() const { return egzaminoRezultatas; }
00110         float getGalutinisV() const { return galutinisV; }
00111         float getGalutinisM() const { return galutinisM; }
00112
00113         //methods
00114         float skaiciuotiVid() const;
00115         float skaiciuotiMed() const;
00116         void skaiciuotiGalutini(char galutinioBudas) override;
00117         void addPazymys(float pazymys) { pazymiai.push_back(pazymys); }
00118     };
00119     std::ostream& operator<(std::ostream& os, const student& studentas);
00120     std::istream& operator>(std::istream& in, student& studentas);

```



```
00121
00122 #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 **generuotiFaile** (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.](#)

```
00001 #ifndef UTILS_H
00002 #define UTILS_H
00003
00004
00015 #include <iostream>
00016 #include <limits>
00017 #include <string>
00018
00019 template<typename T>
00020 T tikrintiInput(const std::string& prompt, const std::string& klaida) {
00021     T value;
00022     while (true) {
00023         std::cout << prompt;
00024         std::cin >> value;
00025         if (!std::cin.fail()) {
00026             std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
00027             return value;
00028         }
00029         std::cerr << klaida << std::endl;
00030         std::cin.clear();
00031         std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
00032     }
00033 }
00034
00035 #endif
```

Index

C:/Users/Lenovo/Desktop/2-uzduotis/vector/include/functions.h,
9, 10
C:/Users/Lenovo/Desktop/2-uzduotis/vector/include/student.h,
10, 11
C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/functions.cpp,
13
C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/main.cpp,
14
C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/student.cpp,
15
C:/Users/Lenovo/Desktop/2-uzduotis/vector/source/utils.h,
16

Human, 7

skaiciuotiGalutini
student, 8
student, 7
skaiciuotiGalutini, 8