v2.0

Alpha

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

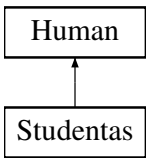
Class Documentation

4.1 Human Class Reference

Bazinė klasė Human reprezentuoja žmogų.

`#include <zmogus.h>`

Inheritance diagram for Human:



Public Member Functions

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

  Virtualus destruktorius.
- virtual void setVardas (const std::string &vardas)=0

  Nustato žmogaus vardą.
- virtual std::string getVardas () const =0

  Gauna žmogaus vardą.
- virtual void setPavarde (const std::string &pavarde)=0

  Nustato žmogaus pavardę.
- virtual std::string getPavarde () const =0

  Gauna žmogaus pavardę.

4.1.1 Detailed Description

Bazinė klasė Human reprezentuoja žmogų.

Ši klasė apibrėžia abstrakčius metodus, kurie turi būti implementuoti vaikų klasėse.

4.1.2 Member Function Documentation

4.1.2.1 getPavarde()

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

Gauna žmogaus pavardę.

Returns

Žmogaus pavardė.

Implemented in Studentas.

4.1.2.2 getVardas()

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

Gauna žmogaus vardą.

Returns

Žmogaus vardas.

Implemented in Studentas.

4.1.2.3 setPavarde()

```
virtual void Human::setPavarde (
            const std::string & pavarde )  [pure virtual]
```

Nustato žmogaus pavardę.

Parameters

pavarde	Nustatoma pavardė.

Implemented in Studentas.

4.1.2.4 setVardas()

```
virtual void Human::setVardas (
            const std::string & vardas )  [pure virtual]
```

Nustato žmogaus vardą.

Parameters

vardas	Nustatomas vardas.

Implemented in Studentas.

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

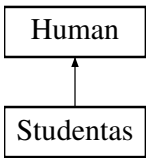
- zmogus.h

4.2 Studentas Class Reference

Klasė Studentas reprezentuoja studentą.

`#include <studentas.h>`

Inheritance diagram for Studentas:



Public Member Functions

- **Studentas** ()
  Numatytasis konstruktorius.
- Studentas (const std::string &vardas, const std::string &pavarde)
  Konstruktorius su parametrais vardui ir pavardėi.
- **Studentas** (const Studentas &other)
  Kopijavimo konstruktorius.
- Studentas & **operator=** (const Studentas &other)
  Priskyrimo operatorius.
- **Studentas** (Studentas &&other) noexcept
  Perkėlimo konstruktorius.
- Studentas & **operator=** (Studentas &&other) noexcept
  Perkėlimo priskyrimo operatorius.
- ~**Studentas** () override
  Destruktorius.
- void setVardas (const std::string &vardas) override
  Nustato studento vardą.
- std::string getVardas () const override
  Gauna studento vardą.
- void setPavarde (const std::string &pavarde) override
  Nustato studento pavardę.
- std::string getPavarde () const override
  Gauna studento pavardę.
- void **setNamuDarbai** (const std::vector< int > &nd)

Nustato studento namų darbų įvertinimus.

- std::vector< int > **getNamuDarbai** () const

  Gauna studento namų darbų įvertinimus.
- void **addNamuDarbas** (int pazymys)

  Prideda namų darbų įvertinimą studentui.
- void **setEgzaminas** (int egzaminas)

  Nustato studento egzamino įvertinimą.
- int **getEgzaminas** () const

  Gauna studento egzamino įvertinimą.
- double **skaiciuotiVidurki** () const

  Skaičiuoja studento vidurkį.
- double **skaiciuotiMediana** () const

  Skaičiuoja studento medianą.
- double skaiciuotiGalutini (bool naudotiVidurki) const

  Skaičiuoja studento galutinį įvertinimą.
- void **atsitiktiniai** ()

  Sugeneruoja atsitiktinius įvertinimus studento namų darbams ir egzaminui.
- void **atsitiktiniaiStudentai** ()

  Sugeneruoja atsitiktinius įvertinimus keliems studentų namų darbams ir egzaminams.

Public Member Functions inherited from Human

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

  Virtualus destruktorius.

Friends

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

  Išvesti studento duomenis į srautą.
- std::istream & operator>> (std::istream &is, Studentas &student)

  Įvesti studento duomenis iš srauto.

4.2.1 Detailed Description

Klasė Studentas reprezentuoja studentą.

Ši klasė paveldi funkcijas iš žmogaus klasės ir prideda funkcionalumą, specifinį studentams, kaip namų darbų ir egzaminų įvertinimų valdymas.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Studentas()

```
Studentas::Studentas (
            const std::string & vardas,
            const std::string & pavarde )
```

Konstruktorius su parametrais vardui ir pavardėi.

Parameters

vardas	Studento vardas.
pavarde	Studento pavardė.

4.2.3 Member Function Documentation

4.2.3.1 getPavarde()

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

Gauna studento pavardę.

Implements Human.

4.2.3.2 getVardas()

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

Gauna studento vardą.

Implements Human.

4.2.3.3 setPavarde()

```
void Studentas::setPavarde (
            const std::string & pavarde )  [override], [virtual]
```

Nustato studento pavardę.

Implements Human.

4.2.3.4 setVardas()

```
void Studentas::setVardas (
            const std::string & vardas )  [override], [virtual]
```

Nustato studento vardą.

Implements Human.

4.2.3.5 skaiciuotiGalutini()

```
double Studentas::skaiciuotiGalutini (
            bool naudotiVidurki ) const
```

Skaičiuoja studento galutinį įvertinimą.

Parameters

naudotiVidurki	Ar naudoti vidurkį (true) ar medianą (false).

4.2.4 Friends And Related Symbol Documentation

4.2.4.1 operator<<

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

Išvesti studento duomenis į srautą.

Parameters

os	Išvesties srautas.
student	Studento objektas, kurio duomenys išvedami.

Returns

Išvesties srauto nuoroda.

4.2.4.2 operator>>

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

Įvesti studento duomenis iš srauto.

Parameters

is	Įvesties srautas.
student	Studento objektas, į kurį įvedami duomenys.

Returns

Įvesties srauto nuoroda.

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

- studentas.h
- studentas.cpp

Chapter 5

File Documentation

5.1 app.h

```
00001 #ifndef FUNCTIONS_H
00002 #define FUNCTIONS_H
00003
00004 #include <vector>
00005 #include <string>
00006 #include "studentas.h"
00007 #include "funkcijos.h"
00008
00012 enum class ContainerType { None, Vector };
00013
00017 enum class Action { None, Generate, Sort };
00018
00024 ContainerType getContainerChoice();
00025
00031 Action getActionChoice();
00032
00040 void performAction(ContainerType containerChoice, Action actionChoice, const std::vector<int>& sizes);
00041
00045 void runApp();
00046
00047 #endif // FUNCTIONS_H
```

5.2 funkcijos.h

```
00001 #ifndef FUNKCIJOSVECTOR_H
00002 #define FUNKCIJOSVECTOR_H
00003
00004 #include "studentas.h"
00005 #include <vector>
00006
00013 void readDataVector(std::vector<Studentas>& studentai, const std::string& failoVardas);
00014
00020 void generateStudentFilesVector(int size);
00021
00027 void rusiuotStudentusVector(const std::string& failoVardas);
00028
00034 void rusiuotStudentusVector2(const std::string& failoVardas);
00035
00041 void rusiuotStudentusVector3(const std::string& failoVardas);
00042
00043 #endif // FUNKCIJOSVECTOR_H
```

5.3 funkcijosVECTOR.h

```
00001 #ifndef FUNKCIJOSVECTOR_H
00002 #define FUNKCIJOSVECTOR_H
00003
00004 #include "studentas.h"
00005 #include <vector>
```

```
00006
00013 void readDataVector(std::vector<Studentas>& studentai, const std::string& failoVardas);
00014
00020 void generateStudentFilesVector(int size);
00021
00027 void rusiuotStudentusVector(const std::string& failoVardas);
00028
00034 void rusiuotStudentusVector2(const std::string& failoVardas);
00035
00041 void rusiuotStudentusVector3(const std::string& failoVardas);
00042
00043 #endif // FUNKCIJOSVECTOR_H
```

5.4 studentas.h

```
00001 #ifndef STUDENTAS_H
00002 #define STUDENTAS_H
00003
00004 #include "zmogus.h"
00005 #include <vector>
00006 #include <iostream>
00007
00014 class Studentas : public Human {
00015 public:
00019     Studentas();
00020
00027     Studentas(const std::string &vardas, const std::string &pavarde);
00028
00029     // Rule of Five
00033     Studentas(const Studentas &other);                // Kopijavimo konstruktorius
00034
00038     Studentas &operator=(const Studentas &other);     // Priskyrimo operatorius
00039
00043     Studentas(Studentas &&other) noexcept;            // Perkėlimo konstruktorius
00044
00048     Studentas &operator=(Studentas &&other) noexcept; // Perkėlimo priskyrimo operatorius
00049
00050     // Destruktorius
00054     ~Studentas() override;
00055
00056     // Implementacija abstrakčių klasės funkcijų
00060     void setVardas(const std::string &vardas) override;
00061
00065     std::string getVardas() const override;
00066
00070     void setPavarde(const std::string &pavarde) override;
00071
00075     std::string getPavarde() const override;
00076
00080     void setNamuDarbai(const std::vector<int> &nd);
00081
00085     std::vector<int> getNamuDarbai() const;
00086
00090     void addNamuDarbas(int pazymys);
00091
00095     void setEgzaminas(int egzaminas);
00096
00100     int getEgzaminas() const;
00101
00105     double skaiciuotiVidurki() const;
00106
00110     double skaiciuotiMediana() const;
00111
00117     double skaiciuotiGalutini(bool naudotiVidurki) const;
00118
00122     void atsitiktiniai();
00123
00127     void atsitiktiniaiStudentai();
00128
00136     friend std::ostream &operator<<(std::ostream &os, const Studentas &student);
00137
00145     friend std::istream &operator>>(std::istream &is, Studentas &student);
00146
00147
00148 private:
00149     std::string vardas;
00150     std::string pavarde;
00151     std::vector<int> nd_rezultatai;
00152     int egzaminas;
00153 };
00154
00155 #endif
```

5.5 zmogus.h

```
00001 #ifndef ZMOGUS_H
00002 #define ZMOGUS_H
00003
00004 #include <string>
00005 #include <vector>
00006
00012 class Human
00013 {
00014 public:
00018     virtual ~Human() = default;
00019
00025     virtual void setVardas(const std::string &vardas) = 0;
00026
00032     virtual std::string getVardas() const = 0;
00033
00039     virtual void setPavarde(const std::string &pavarde) = 0;
00040
00046     virtual std::string getPavarde() const = 0;
00047
00048     // Abstraktūs metodai, kurie gali būti įgyvendinti vaikinėse klasėse:
00049     // virtual void setNamuDarbai(const std::vector<int> &nd) = 0;
00050     // virtual std::vector<int> getNamuDarbai() const = 0;
00051     // virtual void addNamuDarbas(int pazymys) = 0;
00052     // virtual void setEgzaminas(int egzaminas) = 0;
00053     // virtual int getEgzaminas() const = 0;
00054     // virtual double skaiciuotiVidurki() const = 0;
00055     // virtual double skaiciuotiMediana() const = 0;
00056     // virtual double skaiciuotiGalutini(bool naudotiVidurki) const = 0;
00057     // virtual void atsitiktiniai() = 0;
00058     // virtual void atsitiktiniaiStudentai() = 0;
00059 };
00060
00061 #endif
```

Index