

My Project

Generated by Doxygen 1.12.0

1 Objektinis	1
1.1 v1.5	1
1.2 v1.2	2
1.2.1 Aprasymas:	2
1.2.2 destruktorius:	2
1.2.3 kopijavimo operatorius:	2
1.2.4 kopijavimo konstruktorius:	2
1.2.5 << operatorius:	3
1.2.6 >> operatorius:	3
1.3 Spartos analize (v0.3)	3
1.3.1 Vector:	3
1.3.2 List:	4
1.3.2.1 Isvados:	4
1.4 Spartos analize (v1.0)	4
1.4.0.1 1 STRATEGIJA:	5
1.4.0.2 2 STRATEGIJA:	5
1.4.0.3 3 STRATEGIJA:	5
1.4.0.4 Isvados:	5
1.5 Spartos analize (v1.1)	6
1.5.1 class :	6
1.5.2 struct :	6
1.5.2.1 Isvada:	6
1.5.3 Optimizavimo flagu analize :	6
1.5.4 O1 :	6
1.5.5 O2 :	6
1.5.6 O3 :	6
1.5.6.1 Isvada:	6
1.5.7 Instaliavimas bei paleidimas:	7
2 Hierarchical Index	9
2.1 Class Hierarchy	9
3 Class Index	11
3.1 Class List	11
4 File Index	13
4.1 File List	13
5 Class Documentation	15
5.1 stud Class Reference	15
5.1.1 Constructor & Destructor Documentation	16
5.1.1.1 stud() [1/4]	16
5.1.1.2 stud() [2/4]	17
5.1.1.3 ~stud() [1/2]	17

5.1.1.4 stud() [3/4]	17
5.1.1.5 stud() [4/4]	17
5.1.1.6 ~stud() [2/2]	17
5.1.2 Member Function Documentation	17
5.1.2.1 addNd() [1/2]	17
5.1.2.2 addNd() [2/2]	17
5.1.2.3 getEgz() [1/2]	17
5.1.2.4 getEgz() [2/2]	17
5.1.2.5 getMed() [1/2]	18
5.1.2.6 getMed() [2/2]	18
5.1.2.7 getNd() [1/2]	18
5.1.2.8 getNd() [2/2]	18
5.1.2.9 getVid() [1/2]	18
5.1.2.10 getVid() [2/2]	18
5.1.2.11 operator=() [1/2]	18
5.1.2.12 operator=() [2/2]	18
5.1.2.13 setEgz() [1/2]	18
5.1.2.14 setEgz() [2/2]	18
5.1.2.15 setEgzFromLastNd() [1/2]	19
5.1.2.16 setEgzFromLastNd() [2/2]	19
5.1.2.17 setMed() [1/2]	19
5.1.2.18 setMed() [2/2]	19
5.1.2.19 setNd() [1/2]	19
5.1.2.20 setNd() [2/2]	19
5.1.2.21 setRez() [1/2]	19
5.1.2.22 setRez() [2/2]	19
5.1.2.23 setVid() [1/2]	19
5.1.2.24 setVid() [2/2]	20
5.1.2.25 skaiciuotiAmziu() [1/2]	20
5.1.2.26 skaiciuotiAmziu() [2/2]	20
5.1.3 Friends And Related Symbol Documentation	20
5.1.3.1 operator<< [1/2]	20
5.1.3.2 operator<< [2/2]	20
5.1.3.3 operator>> [1/2]	20
5.1.3.4 operator>> [2/2]	20
5.2 zmogus Class Reference	21
5.2.1 Constructor & Destructor Documentation	21
5.2.1.1 zmogus() [1/2]	21
5.2.1.2 ~zmogus() [1/2]	21
5.2.1.3 zmogus() [2/2]	21
5.2.1.4 ~zmogus() [2/2]	22
5.2.2 Member Function Documentation	22

5.2.2.1 getPavarde() [1/2]	22
5.2.2.2 getPavarde() [2/2]	22
5.2.2.3 getVardas() [1/2]	22
5.2.2.4 getVardas() [2/2]	22
5.2.2.5 setPavarde() [1/2]	22
5.2.2.6 setPavarde() [2/2]	22
5.2.2.7 setVardas() [1/2]	22
5.2.2.8 setVardas() [2/2]	22
5.2.2.9 skaiciuotiAmziu() [1/2]	23
5.2.2.10 skaiciuotiAmziu() [2/2]	23
5.2.3 Member Data Documentation	23
5.2.3.1 pavarde	23
5.2.3.2 vardas	23
6 File Documentation	25
6.1 include/lib.h File Reference	25
6.2 lib.h	25
6.3 Project2/lib.h File Reference	26
6.4 lib.h	26
6.5 include/stud.h File Reference	27
6.5.1 Function Documentation	28
6.5.1.1 arvargsiukas()	28
6.5.1.2 generavimas()	28
6.5.1.3 isvedimas()	28
6.5.1.4 isvedimasLst()	28
6.5.1.5 ived()	28
6.5.1.6 kokiastrategija()	28
6.5.1.7 kontNustatymas()	29
6.5.1.8 measureTimeLst()	29
6.5.1.9 measureTimeLst2str()	29
6.5.1.10 measureTimeVec()	29
6.5.1.11 measureTimeVec2str()	29
6.5.1.12 measureTimeVec3str()	29
6.5.1.13 mediana()	30
6.5.1.14 outputfile()	30
6.5.1.15 outputmed()	30
6.5.1.16 outputvid()	30
6.5.1.17 pasirinkti()	30
6.5.1.18 pasirinktikont()	30
6.5.1.19 random_number()	30
6.5.1.20 readFromFile()	30
6.5.1.21 readFromFileLst()	30

6.5.1.22 skirstymas()	31
6.5.1.23 skirstymas2str()	31
6.5.1.24 skirstymas3str()	31
6.5.1.25 skirstymasLst()	31
6.5.1.26 skirstymasLst2str()	31
6.5.1.27 strategija()	31
6.5.1.28 val() [1/2]	31
6.5.1.29 val() [2/2]	31
6.5.1.30 vidurkis()	32
6.6 stud.h	32
6.7 Project2/stud.h File Reference	33
6.7.1 Function Documentation	34
6.7.1.1 arvargsiukas()	34
6.7.1.2 generavimas()	34
6.7.1.3 isvedimas()	35
6.7.1.4 isvedimasLst()	35
6.7.1.5 ived()	35
6.7.1.6 kokiastrategija()	35
6.7.1.7 kontNustatymas()	35
6.7.1.8 measureTimeLst()	35
6.7.1.9 measureTimeLst2str()	35
6.7.1.10 measureTimeVec()	36
6.7.1.11 measureTimeVec2str()	36
6.7.1.12 measureTimeVec3str()	36
6.7.1.13 mediana()	36
6.7.1.14 outputfile()	36
6.7.1.15 outputmed()	36
6.7.1.16 outputvid()	36
6.7.1.17 pasirinkti()	37
6.7.1.18 pasirinktikont()	37
6.7.1.19 random_number()	37
6.7.1.20 readFromFile()	37
6.7.1.21 readFromFileLst()	37
6.7.1.22 skirstymas()	37
6.7.1.23 skirstymas2str()	37
6.7.1.24 skirstymas3str()	37
6.7.1.25 skirstymasLst()	38
6.7.1.26 skirstymasLst2str()	38
6.7.1.27 strategija()	38
6.7.1.28 val() [1/2]	38
6.7.1.29 val() [2/2]	38
6.7.1.30 vidurkis()	38

6.8 stud.h	38
6.9 Project2/listui.cpp File Reference	40
6.9.1 Function Documentation	40
6.9.1.1 isvedimasLst()	40
6.9.1.2 kontNustatymas()	41
6.9.1.3 measureTimeLst()	41
6.9.1.4 pasirinkti()	41
6.9.1.5 pasirinktikont()	41
6.9.1.6 readFromFileLst()	41
6.9.1.7 skirstymasLst()	41
6.10 src/listui.cpp File Reference	41
6.10.1 Function Documentation	42
6.10.1.1 isvedimasLst()	42
6.10.1.2 kontNustatymas()	42
6.10.1.3 measureTimeLst()	42
6.10.1.4 pasirinkti()	42
6.10.1.5 pasirinktikont()	42
6.10.1.6 readFromFileLst()	42
6.10.1.7 skirstymasLst()	43
6.11 Project2/main.cpp File Reference	43
6.11.1 Function Documentation	43
6.11.1.1 main()	43
6.12 src/main.cpp File Reference	43
6.12.1 Function Documentation	43
6.12.1.1 main()	43
6.13 Project2/strategijos.cpp File Reference	43
6.13.1 Function Documentation	44
6.13.1.1 arvargsiukas()	44
6.13.1.2 kokiastategija()	44
6.13.1.3 measureTimeLst2str()	44
6.13.1.4 measureTimeVec2str()	44
6.13.1.5 measureTimeVec3str()	44
6.13.1.6 skirstymas2str()	45
6.13.1.7 skirstymas3str()	45
6.13.1.8 skirstymasLst2str()	45
6.14 src/strategijos.cpp File Reference	45
6.14.1 Function Documentation	45
6.14.1.1 arvargsiukas()	45
6.14.1.2 kokiastategija()	46
6.14.1.3 measureTimeLst2str()	46
6.14.1.4 measureTimeVec2str()	46
6.14.1.5 measureTimeVec3str()	46

6.14.1.6 skirstymas2str()	46
6.14.1.7 skirstymas3str()	46
6.14.1.8 skirstymasLst2str()	47
6.15 Project2/stud.cpp File Reference	47
6.15.1 Function Documentation	47
6.15.1.1 generavimas()	47
6.15.1.2 isvedimas()	47
6.15.1.3 ived()	47
6.15.1.4 measureTimeVec()	48
6.15.1.5 mediana()	48
6.15.1.6 outputfile()	48
6.15.1.7 outputmed()	48
6.15.1.8 outputvid()	48
6.15.1.9 random_number()	48
6.15.1.10 readFromFile()	48
6.15.1.11 skirstymas()	48
6.15.1.12 strategija()	49
6.15.1.13 val()	49
6.15.1.14 vidurkis()	49
6.16 src/stud.cpp File Reference	49
6.16.1 Function Documentation	49
6.16.1.1 generavimas()	49
6.16.1.2 isvedimas()	50
6.16.1.3 ived()	50
6.16.1.4 measureTimeVec()	50
6.16.1.5 mediana()	50
6.16.1.6 outputfile()	50
6.16.1.7 outputmed()	50
6.16.1.8 outputvid()	50
6.16.1.9 random_number()	50
6.16.1.10 readFromFile()	51
6.16.1.11 skirstymas()	51
6.16.1.12 strategija()	51
6.16.1.13 val()	51
6.16.1.14 vidurkis()	51
6.17 README.md File Reference	51
Index	53

Chapter 1

Objektinis1

Projektas objektinio v0.1

1.1 v1.5

Klase zmogus:

Klase stud :

cpp

```
class stud : public zmogus{ private: vector<double>nd; double vid, med, egz, rez;
```

```
public:
```

```
void skaiciuotiAmziu() override {  
    cout << "skaiciuojama"<<endl;
```

```
}
```

```
// Numatytasis konstruktorius stud() : nd(), vid(0.0), med(0.0), egz(0.0), rez(0.0) {}
```

```
//getteriai vector<double> getNd() const { return nd; } double getEgz() const { return egz; } double getVid() const {  
return vid; } double getMed() const { return med; }
```

```
//seteriai void setNd(vector<double> nd) { this->nd = nd; } void setEgz(double egz) { this->egz = egz; } void  
setVid(double vid) { this->vid = vid; } void setMed(double med) { this->med = med; } void setRez(double rez) {  
this->rez = rez; }
```

```
void addNd(double grade) { nd.push_back(grade); } //kopijavimo operatorius stud& operator=(const stud& a) { if  
(this == &a) return *this; this->vardas = a.vardas; this->pavarde = a.pavarde; this->nd = a.nd; this->egz = a.egz;  
this->vid = a.vid; this->med = a.med; this->rez = a.rez; return *this; }
```

```
// Kopijavimo konstruktorius stud(const stud& other) { vardas = other.vardas; pavarde = other.pavarde; nd = other.nd;  
egz = other.egz; vid = other.vid; med = other.med; rez = other.rez; }
```

```

void setEgzFromLastNd() {
    if (!nd.empty()) {
        egz = nd.back();
        nd.pop_back();
    } else {
        std::cerr << "Error: No grades in nd to assign to egz." << std::endl;
    }
}

friend std::istream& operator>>(std::istream& in, stud& student) {
    std::cout << "enter name: ";
    in >> student.vardas;
    std::cout << "enter surname: ";
    in >> student.pavarde;
    return in;
}

// void clearData() { // vardas.clear(); // pavarde.clear(); // nd.clear(); // egz = 0.0; // vid = 0.0; // med = 0.0; // rez = 0.0; // }

friend std::ostream& operator<<(std::ostream& out, const stud& student) { out << std::left << std::setw(15) <<
student.vardas << std::left << std::setw(15) << student.pavarde << std::left << std::setw(15) << std::fixed
<< std::setprecision(2) << student.vid; return out; }

~stud(){} // destruktorius;
bandymas sukurti zmogaus objekta ismes klaida :

markdown error: cannot declare variable 'temp' to be of abstract type 'zmogus'because the following virtual functions
are pure within 'zmogus':'virtual void zmogus::skaiciuotiAmziu\(\)'|
...

```

1.2 v1.2

1.2.1 Aprasymas:

Igyvendinta rule of three, isvesties ir ivesties operatoriai.

1.2.2 destruktorius:

1.2.3 kopijavimo operatorius:

Kopijavimo operatorius suveikia tada, kai jau egzistuojantis objektas priskiria kito objekto reikšmes. Objektas, kuris yra kairiajame lygybės zenklo (=) kraste, perima desiniajame kraste esancio objekto atributu reikšmes. Parametras Stud a yra const ir perduodamas kaip nuoroda (reference type). Taip daroma, kad parametras butu tik skaitomas ir nereiketu kurti nereikalingu kopiju, taupant atminties resursus.

1.2.4 kopijavimo konstruktorius:

Kopijavimo konstruktorius yra iskvieciamas, kai naujai sukuriamas objektas igyja jau egzistuojancio objekto reikšmes.

1.2.5 << operatorius:

padedą sutaupyti kodo eilutes:

1.2.6 >> operatorius:

padedą sutaupyti kodo eilutes:

1.3 Spartos analize (v0.3)

1.3.1 Vector:

1000 Failo kurimas uztruko: 0.008931 s
vector 1000 Duomenų nuskaitymas uztruko: 0.020995 s
vector 1000 Studentai surusiuoti pagal vardus per 0.008931 s
vector 1000 Studentų rusiavimas uztruko: 0.000000 s
vector 1000 Duomenų išvedimas uztruko: 0.008744 s
vector 1000 išviso uztruko 0.044425 s

10 000 Failo kurimas uztruko: 0.066196 s
vector 10 000 Duomenų nuskaitymas uztruko: 0.196940 s
vector 10 000 Studentai surusiuoti pagal vardus per 0.066196 s
vector 10 000 Studentų rusiavimas uztruko: 0.004815 s
vector 10 000 Duomenų išvedimas uztruko: 0.076159 s
vector 10 000 išviso uztruko 0.364299 s

100 000 Failo kurimas uztruko: 0.673784 s
vector 100 000 Duomenų nuskaitymas uztruko: 2.038001 s
vector 100 000 Studentai surusiuoti pagal vardus per 0.673784 s
vector 100 000 Studentų rusiavimas uztruko: 0.034068 s
vector 100 000 Duomenų išvedimas uztruko: 0.719583 s
vector 100 000 išviso uztruko 3.662217 s

1 000 000 Failo kurimas uztruko: 6.486274 s
vector 1 000 000 Duomenų nuskaitymas uztruko: 20.344804 s
vector 1 000 000 Studentai surusiuoti pagal vardus per 6.486274 s
vector 1 000 000 Studentų rusiavimas uztruko: 0.399872 s
vector 1 000 000 Duomenų išvedimas uztruko: 7.014264 s
vector 1 000 000 išviso uztruko 36.670251 s

10 000 000 Failo kurimas uztruko: 65.061046 s
vector 10 000 000 Duomenų nuskaitymas uztruko: 199.550376 s
vector 10 000 000 Studentai surusiuoti pagal vardus per 65.061046 s
vector 10 000 000 Studentų rusiavimas uztruko: 3.424919 s
vector 10 000 000 Duomenų išvedimas uztruko: 69.298156 s
vector 10 000 000 išviso uztruko 368.115694 s

1.3.2 List:

1000 Failo kurimas uztruko: 0.008738 s
list 1000 Duomenu nuskaitymas uztruko: 0.019995 s
list 1000 studentai surusiuoti pagal vardus per 0.000000 s
list 1000 Studentu rusiavimas uztruko: 0.000000 s
list 1000 Duomenu isvedimas uztruko: 0.008877 s
list 1000 isviso uztruko 0.043050 s

10 000 Failo kurimas uztruko: 0.063990 s
list 10 000 Duomenu nuskaitymas uztruko: 0.192936 s
list 10 000 studentai surusiuoti pagal vardus per 0.004001 s
list 10 000 Studentu rusiavimas uztruko: 0.002995 s
list 10 000 Duomenu isvedimas uztruko: 0.078867 s
list 10 000 isviso uztruko 0.346756 s

100 000 Failo kurimas uztruko: 0.642807 s
list 100 000 Duomenu nuskaitymas uztruko: 1.988361 s
list 100 000 studentai surusiuoti pagal vardus per 0.048984 s
list 100 000 Studentu rusiavimas uztruko: 0.035992 s
list 100 000 Duomenu isvedimas uztruko: 0.696766 s
list 100 000 isviso uztruko 3.417449 s

1 000 000 Failo kurimas uztruko: 6.341986 s
list 1 000 000 Duomenu nuskaitymas uztruko: 19.773706 s
list 1 000 000 studentai surusiuoti pagal vardus per 0.554823 s
list 1 000 000 Studentu rusiavimas uztruko: 0.344901 s
list 1 000 000 Duomenu isvedimas uztruko: 7.058292 s
list 1 000 000 isviso uztruko 34.076979 s

10 000 000 Failo kurimas uztruko: 64.174002 s
list 10 000 000 Duomenu nuskaitymas uztruko: 203.791442 s
list 10 000 000 studentai surusiuoti pagal vardus per 6.642581 s
list 10 000 000 Studentu rusiavimas uztruko: 3.806869 s
list 10 000 000 Duomenu isvedimas uztruko: 76.000803 s
list 10 000 000 isviso uztruko 354.420880 s

CPU: i7-4790 RAM:16gb

1.3.2.1 Isvados:

Didziausi skirtumai matomi studentus rusiuojant pagal vardus arba pavardes, kas listuose daroma zymiai greiciau nei vektoriuose.

1.4 Spartos analize (v1.0)

Testuojam studentu rusiavima i dvi grupes 3 strategijas:

1.4.0.1 1 STRATEGIJA:

vector 1000 Studentu rusiavimas uztruko: 0.000000 s
vector 10 000 Studentu rusiavimas uztruko: 0.001998 s
vector 100 000 Studentu rusiavimas uztruko: 0.019245 s
vector 1 000 000 Studentu rusiavimas uztruko: 0.178742 s
vector 10 000 000 Studentu rusiavimas uztruko: 1.799656 s
Vidurkis: 0.3999282 s

list 1000 Studentu rusiavimas uztruko: 0.000751 s
list 10 000 Studentu rusiavimas uztruko: 0.001999 s
list 100 000 Studentu rusiavimas uztruko: 0.032069 s
list 1 000 000 Studentu rusiavimas uztruko: 0.203321 s
list 10 000 000 Studentu rusiavimas uztruko: 2.324669 s
Vidurkis: 0.5125618 s

1.4.0.2 2 STRATEGIJA:

vector 1000 Studentu rusiavimas uztruko: 0.002998 s
vector 10 000 Studentu rusiavimas uztruko: 0.296418 s
vector 100 000 Studentu rusiavimas uztruko: 32.306840 s
vector 1 000 000 Studentu rusiavimas uztruko: - (labai ilgai)
vector 10 000 000 Studentu rusiavimas uztruko: -
Vidurkis: -

list 1000 Studentu rusiavimas uztruko: 0.000000 s
list 10 000 Studentu rusiavimas uztruko: 0.000999 s
list 100 000 Studentu rusiavimas uztruko: 0.009869 s
list 1 000 000 Studentu rusiavimas uztruko: 0.094055 s
list 10 000 000 Studentu rusiavimas uztruko: 0.987879 s
Vidurkis: 0.2185604 s

1.4.0.3 3 STRATEGIJA:

vector 1000 Studentu rusiavimas uztruko: 0.001001 s
vector 10 000 Studentu rusiavimas uztruko: 0.000997 s
vector 100 000 Studentu rusiavimas uztruko: 0.016252 s
vector 1 000 000 Studentu rusiavimas uztruko: 0.159152 s
vector 10 000 000 Studentu rusiavimas uztruko: 1.508008 s
Vidurkis: 0.337082 s

1.4.0.4 Isvados:

2 strategija, listu atzvilgiu, yra zymiai geresne nei pirma, bet vektoriams tragiska, o pritaikius funkcijas is STL isprendzia sia problema vektoriams(zr. 3 strategijos rezultatus).

3 strategijos rezultatu foto:

CPU: i7-4790 RAM:16gb

1.5 Spartos analize (v1.1)

1.5.1 class :

100000 Failo kurimas uztruko: 0.657203 s
vector 100000 Duomenu nuskaitymas uztruko: 1.903391 s
vector 100000 Studentai surusiuoti pagal vardus per 0.657203 s
vector 100000 Studentu rusiavimas uztruko: 0.019994 s
vector 100000 Duomenu isvedimas uztruko: 0.863130 s
vector 100000 isviso uztruko 3.609665 s

1000000 Failo kurimas uztruko: 6.477034 s
vector 1000000 Duomenu nuskaitymas uztruko: 19.289833 s
vector 1000000 Studentai surusiuoti pagal vardus per 6.477034 s
vector 1000000 Studentu rusiavimas uztruko: 0.276924 s
vector 1000000 Duomenu isvedimas uztruko: 8.619423 s
vector 1000000 isviso uztruko 36.359636 s

1.5.2 struct :

100000 Failo kurimas uztruko: 0.653854 s
vector 100000 Duomenu nuskaitymas uztruko: 1.860402 s
vector 100000 Studentai surusiuoti pagal vardus per 0.653854 s
vector 100000 Studentu rusiavimas uztruko: 0.014010 s
vector 100000 Duomenu isvedimas uztruko: 0.860581 s
vector 100000 isviso uztruko 3.442403 s

1000000 Failo kurimas uztruko: 6.471268 s
vector 1000000 Duomenu nuskaitymas uztruko: 19.039902 s
vector 1000000 Studentai surusiuoti pagal vardus per 6.471268 s
vector 1000000 Studentu rusiavimas uztruko: 0.175948 s
vector 1000000 Duomenu isvedimas uztruko: 8.439341 s
vector 1000000 isviso uztruko 34.759877 s

1.5.2.1 Isvada:

kazkokio labai akivaizdaus greitumo laiko prasme tarp abieju nepastebejau(labai nezymus).

1.5.3 Optimizavimo flagu analize :

1.5.4 O1 :

1.5.5 O2 :

1.5.6 O3 :

1.5.6.1 Isvada:

class exe failas maziau sveria nei struct,o kazkoks skirtumas pakeitus flagus nesimato kazkodel, kazkuriam turejo sumazeti exe failo dydis, bet nesumazejo (buvo daroma per codeblocks).

1.5.7 Instaliavimas bei paleidimas:

Atsisiunciat V0.1 release, kadangi naudojama cmake paleidimui reikes ne mazesnes kaip 3.31 versijos cmake windows sistemai, o kompiliavimui uzteks tiesiog du kartus paspausti ant run.bat faila.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

zmogus	21
stud	15
stud	15

Chapter 3

Class Index

3.1 Class List

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

stud	15
zmogus	21

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

include/lib.h	25
include/stud.h	27
Project2/lib.h	26
Project2/listui.cpp	40
Project2/main.cpp	43
Project2/strategijos.cpp	43
Project2/stud.cpp	47
Project2/stud.h	33
src/listui.cpp	41
src/main.cpp	43
src/strategijos.cpp	45
src/stud.cpp	49

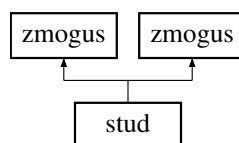
Chapter 5

Class Documentation

5.1 stud Class Reference

```
#include <stud.h>
```

Inheritance diagram for stud:



Public Member Functions

- void `skaiciuotiAmziu ()` override
- `stud ()`
- `vector< double > getNd () const`
- `double getEgz () const`
- `double getVid () const`
- `double getMed () const`
- `void setNd (vector< double > nd)`
- `void setEgz (double egz)`
- `void setVid (double vid)`
- `void setMed (double med)`
- `void setRez (double rez)`
- `void addNd (double grade)`
- `stud & operator= (const stud &a)`
- `stud (const stud &other)`
- `void setEgzFromLastNd ()`
- `~stud ()`
- void `skaiciuotiAmziu ()` override
- `stud ()`
- `vector< double > getNd () const`
- `double getEgz () const`
- `double getVid () const`
- `double getMed () const`

- void [setNd](#) (vector< double > nd)
- void [setEgz](#) (double egz)
- void [setVid](#) (double vid)
- void [setMed](#) (double med)
- void [setRez](#) (double rez)
- void [addNd](#) (double grade)
- [stud](#) & [operator=](#) (const [stud](#) &a)
- [stud](#) (const [stud](#) &other)
- void [setEgzFromLastNd](#) ()
- [~stud](#) ()

Public Member Functions inherited from [zmogus](#)

- string [getVardas](#) () const
- string [getPavarde](#) () const
- void [setVardas](#) (string [vardas](#))
- void [setPavarde](#) (string [pavarde](#))
- [zmogus](#) ()
- virtual [~zmogus](#) ()
- string [getVardas](#) () const
- string [getPavarde](#) () const
- void [setVardas](#) (string [vardas](#))
- void [setPavarde](#) (string [pavarde](#))
- [zmogus](#) ()
- virtual [~zmogus](#) ()

Friends

- std::istream & [operator>>](#) (std::istream &in, [stud](#) &student)
- std::ostream & [operator<<](#) (std::ostream &out, const [stud](#) &student)
- std::istream & [operator>>](#) (std::istream &in, [stud](#) &student)
- std::ostream & [operator<<](#) (std::ostream &out, const [stud](#) &student)

Additional Inherited Members

Protected Attributes inherited from [zmogus](#)

- string [vardas](#)
- string [pavarde](#)

5.1.1 Constructor & Destructor Documentation

5.1.1.1 [stud](#)() [1/4]

```
stud::stud () [inline]
```


5.1.1.2 stud() [2/4]

```
stud::stud (  
    const stud & other) [inline]
```

5.1.1.3 ~stud() [1/2]

```
stud::~stud () [inline]
```

5.1.1.4 stud() [3/4]

```
stud::stud () [inline]
```

5.1.1.5 stud() [4/4]

```
stud::stud (  
    const stud & other) [inline]
```

5.1.1.6 ~stud() [2/2]

```
stud::~stud () [inline]
```

5.1.2 Member Function Documentation

5.1.2.1 addNd() [1/2]

```
void stud::addNd (  
    double grade) [inline]
```

5.1.2.2 addNd() [2/2]

```
void stud::addNd (  
    double grade) [inline]
```

5.1.2.3 getEgz() [1/2]

```
double stud::getEgz () const [inline]
```

5.1.2.4 getEgz() [2/2]

```
double stud::getEgz () const [inline]
```

5.1.2.5 getMed() [1/2]

```
double stud::getMed () const [inline]
```

5.1.2.6 getMed() [2/2]

```
double stud::getMed () const [inline]
```

5.1.2.7 getNd() [1/2]

```
vector< double > stud::getNd () const [inline]
```

5.1.2.8 getNd() [2/2]

```
vector< double > stud::getNd () const [inline]
```

5.1.2.9 getVid() [1/2]

```
double stud::getVid () const [inline]
```

5.1.2.10 getVid() [2/2]

```
double stud::getVid () const [inline]
```

5.1.2.11 operator=() [1/2]

```
stud & stud::operator= (  
    const stud & a) [inline]
```

5.1.2.12 operator=() [2/2]

```
stud & stud::operator= (  
    const stud & a) [inline]
```

5.1.2.13 setEgz() [1/2]

```
void stud::setEgz (  
    double egz) [inline]
```

5.1.2.14 setEgz() [2/2]

```
void stud::setEgz (  
    double egz) [inline]
```

5.1.2.15 setEgzFromLastNd() [1/2]

```
void stud::setEgzFromLastNd () [inline]
```

5.1.2.16 setEgzFromLastNd() [2/2]

```
void stud::setEgzFromLastNd () [inline]
```

5.1.2.17 setMed() [1/2]

```
void stud::setMed (  
    double med) [inline]
```

5.1.2.18 setMed() [2/2]

```
void stud::setMed (  
    double med) [inline]
```

5.1.2.19 setNd() [1/2]

```
void stud::setNd (  
    vector< double > nd) [inline]
```

5.1.2.20 setNd() [2/2]

```
void stud::setNd (  
    vector< double > nd) [inline]
```

5.1.2.21 setRez() [1/2]

```
void stud::setRez (  
    double rez) [inline]
```

5.1.2.22 setRez() [2/2]

```
void stud::setRez (  
    double rez) [inline]
```

5.1.2.23 setVid() [1/2]

```
void stud::setVid (  
    double vid) [inline]
```

5.1.2.24 setVid() [2/2]

```
void stud::setVid (  
    double vid) [inline]
```

5.1.2.25 skaiciuotiAmziu() [1/2]

```
void stud::skaiciuotiAmziu () [inline], [override], [virtual]
```

Implements [zmogus](#).

5.1.2.26 skaiciuotiAmziu() [2/2]

```
void stud::skaiciuotiAmziu () [inline], [override], [virtual]
```

Implements [zmogus](#).

5.1.3 Friends And Related Symbol Documentation

5.1.3.1 operator<< [1/2]

```
std::ostream & operator<< (  
    std::ostream & out,  
    const stud & student) [friend]
```

5.1.3.2 operator<< [2/2]

```
std::ostream & operator<< (  
    std::ostream & out,  
    const stud & student) [friend]
```

5.1.3.3 operator>> [1/2]

```
std::istream & operator>> (  
    std::istream & in,  
    stud & student) [friend]
```

5.1.3.4 operator>> [2/2]

```
std::istream & operator>> (  
    std::istream & in,  
    stud & student) [friend]
```

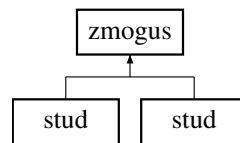
The documentation for this class was generated from the following files:

- include/[stud.h](#)
- Project2/[stud.h](#)

5.2 zmogus Class Reference

```
#include <stud.h>
```

Inheritance diagram for zmogus:



Public Member Functions

- string [getVardas](#) () const
- string [getPavarde](#) () const
- void [setVardas](#) (string [vardas](#))
- void [setPavarde](#) (string [pavarde](#))
- [zmogus](#) ()
- virtual [~zmogus](#) ()
- virtual void [skaiciuotiAmziu](#) ()=0
- string [getVardas](#) () const
- string [getPavarde](#) () const
- void [setVardas](#) (string [vardas](#))
- void [setPavarde](#) (string [pavarde](#))
- [zmogus](#) ()
- virtual [~zmogus](#) ()
- virtual void [skaiciuotiAmziu](#) ()=0

Protected Attributes

- string [vardas](#)
- string [pavarde](#)

5.2.1 Constructor & Destructor Documentation

5.2.1.1 [zmogus\(\)](#) [1/2]

```
zmogus::zmogus () [inline]
```

5.2.1.2 [~zmogus\(\)](#) [1/2]

```
virtual zmogus::~~zmogus () [inline], [virtual]
```

5.2.1.3 [zmogus\(\)](#) [2/2]

```
zmogus::zmogus () [inline]
```

5.2.1.4 ~zmogus() [2/2]

```
virtual zmogus::~zmogus () [inline], [virtual]
```

5.2.2 Member Function Documentation

5.2.2.1 getPavarde() [1/2]

```
string zmogus::getPavarde () const [inline]
```

5.2.2.2 getPavarde() [2/2]

```
string zmogus::getPavarde () const [inline]
```

5.2.2.3 getVardas() [1/2]

```
string zmogus::getVardas () const [inline]
```

5.2.2.4 getVardas() [2/2]

```
string zmogus::getVardas () const [inline]
```

5.2.2.5 setPavarde() [1/2]

```
void zmogus::setPavarde (  
    string pavarde) [inline]
```

5.2.2.6 setPavarde() [2/2]

```
void zmogus::setPavarde (  
    string pavarde) [inline]
```

5.2.2.7 setVardas() [1/2]

```
void zmogus::setVardas (  
    string vardas) [inline]
```

5.2.2.8 setVardas() [2/2]

```
void zmogus::setVardas (  
    string vardas) [inline]
```

5.2.2.9 skaiciuotiAmziu() [1/2]

```
virtual void zmogus::skaiciuotiAmziu () [pure virtual]
```

Implemented in [stud](#), and [stud](#).

5.2.2.10 skaiciuotiAmziu() [2/2]

```
virtual void zmogus::skaiciuotiAmziu () [pure virtual]
```

Implemented in [stud](#), and [stud](#).

5.2.3 Member Data Documentation

5.2.3.1 pavarde

```
string zmogus::pavarde [protected]
```

5.2.3.2 vardas

```
string zmogus::vardas [protected]
```

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

- include/[stud.h](#)
- Project2/[stud.h](#)

Chapter 6

File Documentation

6.1 include/lib.h File Reference

```
#include <iostream>
#include <string>
#include <vector>
#include <iomanip>
#include <algorithm>
#include <cstdlib>
#include <ctime>
#include <fstream>
#include <sstream>
#include <stdexcept>
#include <chrono>
#include <list>
```

6.2 lib.h

[Go to the documentation of this file.](#)

```
00001 #ifndef LIB_H_INCLUDED
00002 #define LIB_H_INCLUDED
00003
00004 #include <iostream>
00005 #include <string>
00006 #include <vector>
00007 #include <iomanip>
00008 #include <algorithm>
00009 #include <cstdlib>
00010 #include <ctime>
00011 #include <fstream>
00012 #include <sstream>
00013 #include <stdexcept>
00014 #include <chrono>
00015 #include <list>
00016
00017
00018 using std::endl;
00019 using std::cout;
00020 using std::cin;
00021 using std::left;
00022 using std::right;
00023 using std::setw;
00024 using std::setprecision;
00025 using std::fixed;
00026 using std::string;
00027 using std::vector;
```

```
00028 using std::sort;
00029 using std::stringstream;
00030 using std::ifstream;
00031 using std::getline;
00032 using std::exception;
00033 using std::runtime_error;
00034 using std::streamsize;
00035 using std::numeric_limits;
00036 using std::cerr;
00037 using std::istringstream;
00038 using std::runtime_error;
00039 using std::ofstream;
00040 using namespace std::chrono;
00041 using std::to_string;
00042 using std::list;
00043 using std::remove;
00044 using std::partition;
00045 using std::copy;
00046 using std::remove_if;
00047 using std::back_inserter;
00048
00049 #endif // LIB_H_INCLUDED
```

6.3 Project2/lib.h File Reference

```
#include <iostream>
#include <string>
#include <vector>
#include <iomanip>
#include <algorithm>
#include <cstdlib>
#include <ctime>
#include <fstream>
#include <sstream>
#include <stdexcept>
#include <chrono>
#include <list>
```

6.4 lib.h

[Go to the documentation of this file.](#)

```
00001 #ifndef LIB_H_INCLUDED
00002 #define LIB_H_INCLUDED
00003
00004 #include <iostream>
00005 #include <string>
00006 #include <vector>
00007 #include <iomanip>
00008 #include <algorithm>
00009 #include <cstdlib>
00010 #include <ctime>
00011 #include <fstream>
00012 #include <sstream>
00013 #include <stdexcept>
00014 #include <chrono>
00015 #include <list>
00016
00017
00018 using std::endl;
00019 using std::cout;
00020 using std::cin;
00021 using std::left;
00022 using std::right;
00023 using std::setw;
00024 using std::setprecision;
00025 using std::fixed;
00026 using std::string;
00027 using std::vector;
```

```

00028 using std::sort;
00029 using std::stringstream;
00030 using std::ifstream;
00031 using std::getline;
00032 using std::exception;
00033 using std::runtime_error;
00034 using std::streamsize;
00035 using std::numeric_limits;
00036 using std::cerr;
00037 using std::istringstream;
00038 using std::runtime_error;
00039 using std::ofstream;
00040 using namespace std::chrono;
00041 using std::to_string;
00042 using std::list;
00043 using std::remove;
00044 using std::partition;
00045 using std::copy;
00046 using std::remove_if;
00047 using std::back_inserter;
00048
00049 #endif // LIB_H_INCLUDED

```

6.5 include/stud.h File Reference

```
#include "lib.h"
```

Classes

- class [zmogus](#)
- class [stud](#)

Functions

- void [ived](#) ([stud](#) &lok)
- void [outputvid](#) ([stud](#) lok)
- double [vidurkis](#) ([stud](#) &lok)
- void [outputmed](#) ([stud](#) lok)
- double [mediana](#) ([stud](#) &lok)
- void [readFromFile](#) (const string &failas, vector< [stud](#) > &vec1)
- void [outputfile](#) ([stud](#) lok)
- void [isvedimas](#) (const string pavadinimas, const vector< [stud](#) > &vec1)
- int [random_number](#) ()
- void [skirstymas](#) (const vector< [stud](#) > &vec1, vector< [stud](#) > &kietiakai, vector< [stud](#) > &vargsiukai)
- void [generavimas](#) (const string filename, int stud_kiekis)
- void [measureTimeVec](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [measureTimeLst](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [pasirinkti](#) (int &pasirinkimas)
- void [pasirinktikont](#) (int &kont)
- void [kontNustatymas](#) (string &konteineris, int &kont)
- void [readFromFileLst](#) (const string &failas, list< [stud](#) > &lst1)
- void [skirstymasLst](#) (const list< [stud](#) > &lst1, list< [stud](#) > &kietiakai, list< [stud](#) > &vargsiukai)
- void [isvedimasLst](#) (const string pavadinimas, const list< [stud](#) > &lst1)
- void [skirstymas2str](#) (vector< [stud](#) > &vec, vector< [stud](#) > &vargsiukai)
- void [measureTimeVec2str](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [skirstymasLst2str](#) (list< [stud](#) > &lst1, list< [stud](#) > &vargsiukai)
- void [measureTimeLst2str](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [strategija](#) (int &strat)

- void `kokiastrategija` (string filename, int strat, int kont, int pasirinkimas, const string &kontaineris, int numeris)
- bool `arvargsiukas` (const `stud` &s)
- void `skirstymas3str` (vector< `stud` > &vec1, vector< `stud` > &vargsiukai)
- void `measureTimeVec3str` (const string filename, int stud_num, int pasirinkimas, string kontaineris)
- void `val` (vector< `stud` > &vec)
- void `val` (`stud` &student)

6.5.1 Function Documentation

6.5.1.1 `arvargsiukas()`

```
bool arvargsiukas (  
    const stud & s)
```

6.5.1.2 `generavimas()`

```
void generavimas (  
    const string filename,  
    int stud_kiekis)
```

6.5.1.3 `isvedimas()`

```
void isvedimas (  
    const string pavadinimas,  
    const vector< stud > & vec1)
```

6.5.1.4 `isvedimasLst()`

```
void isvedimasLst (  
    const string pavadinimas,  
    const list< stud > & lst1)
```

6.5.1.5 `ived()`

```
void ived (  
    stud & lok)
```

6.5.1.6 `kokiastrategija()`

```
void kokiastrategija (  
    string filename,  
    int strat,  
    int kont,  
    int pasirinkimas,  
    const string & kontaineris,  
    int numeris)
```

6.5.1.7 kontNustatymas()

```
void kontNustatymas (
    string & konteineris,
    int & kont)
```

6.5.1.8 measureTimeLst()

```
void measureTimeLst (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.5.1.9 measureTimeLst2str()

```
void measureTimeLst2str (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.5.1.10 measureTimeVec()

```
void measureTimeVec (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.5.1.11 measureTimeVec2str()

```
void measureTimeVec2str (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.5.1.12 measureTimeVec3str()

```
void measureTimeVec3str (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.5.1.13 mediana()

```
double mediana (  
    stud & lok)
```

6.5.1.14 outputfile()

```
void outputfile (  
    stud lok)
```

6.5.1.15 outputmed()

```
void outputmed (  
    stud lok)
```

6.5.1.16 outputvid()

```
void outputvid (  
    stud lok)
```

6.5.1.17 pasirinkti()

```
void pasirinkti (  
    int & pasirinkimas)
```

6.5.1.18 pasirinktikont()

```
void pasirinktikont (  
    int & kont)
```

6.5.1.19 random_number()

```
int random_number ()
```

6.5.1.20 readFromFile()

```
void readFromFile (  
    const string & failas,  
    vector< stud > & vec1)
```

6.5.1.21 readFromFileLst()

```
void readFromFileLst (  
    const string & failas,  
    list< stud > & lst1)
```

6.5.1.22 skirstymas()

```
void skirstymas (
    const vector< stud > & vec1,
    vector< stud > & kietiakai,
    vector< stud > & vargsiukai)
```

6.5.1.23 skirstymas2str()

```
void skirstymas2str (
    vector< stud > & vec,
    vector< stud > & vargsiukai)
```

6.5.1.24 skirstymas3str()

```
void skirstymas3str (
    vector< stud > & vec1,
    vector< stud > & vargsiukai)
```

6.5.1.25 skirstymasLst()

```
void skirstymasLst (
    const list< stud > & lst1,
    list< stud > & kietiakai,
    list< stud > & vargsiukai)
```

6.5.1.26 skirstymasLst2str()

```
void skirstymasLst2str (
    list< stud > & lst1,
    list< stud > & vargsiukai)
```

6.5.1.27 strategija()

```
void strategija (
    int & strat)
```

6.5.1.28 val() [1/2]

```
void val (
    stud & student)
```

6.5.1.29 val() [2/2]

```
void val (
    vector< stud > & vec)
```

6.5.1.30 vidurkis()

```
double vidurkis (
    stud & lok)
```

6.6 stud.h

[Go to the documentation of this file.](#)

```
00001 #ifndef STUD_H_INCLUDED
00002 #define STUD_H_INCLUDED
00003 #include "lib.h"
00004
00005 class zmogus {
00006     protected:
00007     string vardas,pavarde;
00008     public:
00009         string getVardas() const { return vardas; };
00010         string getPavarde() const { return pavarde; };
00011         void setVardas(string vardas) { this->vardas = vardas; };
00012         void setPavarde(string pavarde) { this->pavarde = pavarde; };
00013
00014         zmogus() {} //cout << "Zmogus sukurtas" << endl; kad pamatyti kurima zmogaus
00015         virtual ~zmogus() {}
00016
00017         virtual void skaiciuotiAmziu() = 0;
00018 };
00019
00020 class stud : public zmogus{
00021     private:
00022         vector<double>nd;
00023         double vid, med, egz, rez;
00024
00025     public:
00026
00027         void skaiciuotiAmziu() override {
00028             cout << "skaiciuojama" << endl;
00029         }
00030
00031         // Numatytais konstruktorius
00032         stud() : nd(), vid(0.0), med(0.0), egz(0.0), rez(0.0) {}
00033
00034         //getteriai
00035         vector<double> getNd() const { return nd; }
00036         double getEgz() const { return egz; }
00037         double getVid() const { return vid; }
00038         double getMed() const { return med; }
00039
00040         //seteriai
00041         void setNd(vector<double> nd) { this->nd = nd; }
00042         void setEgz(double egz) { this->egz = egz; }
00043         void setVid(double vid) { this->vid = vid; }
00044         void setMed(double med) { this->med = med; }
00045         void setRez(double rez) { this->rez = rez; }
00046
00047         void addNd(double grade) {
00048             nd.push_back(grade);
00049         }
00050         //kopiavimo operatorius
00051         stud& operator=(const stud& a) {
00052             if (this == &a) return *this;
00053             this->vardas = a.vardas;
00054             this->pavarde = a.pavarde;
00055             this->nd = a.nd;
00056             this->egz = a.egz;
00057             this->vid = a.vid;
00058             this->med = a.med;
00059             this->rez = a.rez;
00060             return *this;
00061         }
00062
00063         // Kopijavimo konstruktorius
00064         stud(const stud& other) {
00065             vardas = other.vardas;
00066             pavarde = other.pavarde;
00067             nd = other.nd;
00068             egz = other.egz;
00069             vid = other.vid;
00070             med = other.med;
00071             rez = other.rez;
```



```

00072     }
00073
00074
00075     void setEgzFromLastNd() {
00076         if (!nd.empty()) {
00077             egz = nd.back();
00078             nd.pop_back();
00079         } else {
00080             std::cerr << "Error: No grades in nd to assign to egz." << std::endl;
00081         }
00082     }
00083
00084
00085     friend std::istream& operator>>(std::istream& in, stud& student) {
00086         std::cout << "enter name: ";
00087         in >> student.vardas;
00088         std::cout << "enter surname: ";
00089         in >> student.pavarde;
00090         return in;
00091     }
00092
00093     // void clearData() {
00094     //     vardas.clear();
00095     //     pavarde.clear();
00096     //     nd.clear();
00097     //     egz = 0.0;
00098     //     vid = 0.0;
00099     //     med = 0.0;
00100     //     rez = 0.0;
00101     // }
00102
00103
00104     friend std::ostream& operator<<(std::ostream& out, const stud& student) {
00105         out << std::left << std::setw(15) << student.vardas
00106             << std::left << std::setw(15) << student.pavarde
00107             << std::left << std::setw(15) << std::fixed << std::setprecision(2) << student.vid;
00108         return out;
00109     }
00110
00111     ~stud(){} // destruktorius
00112 };
00113
00114 void ived(stud &lok);
00115 void outputvid(stud lok);
00116 double vidurkis(stud &lok);
00117 void outputmed(stud lok);
00118 double mediana(stud &lok);
00119 void readFromFile(const string& failas, vector<stud>& vec1);
00120 void outputFile(stud lok);
00121 void isvedimas(const string pavadinimas, const vector<stud>& vec1);
00122 int random_number();
00123 void skirstymas(const vector<stud>& vec1, vector<stud>& kietiakai, vector<stud>& vargsiukai);
00124 void generavimas(const string filename, int stud_kiekis);
00125 void measureTimeVec(const string filename, int stud_num, int pasirinkimas, string konteineris);
00126 void measureTimeLst(const string filename, int stud_num, int pasirinkimas, string konteineris);
00127 void pasirinkti(int &pasirinkimas);
00128 void pasirinktikont(int &kont);
00129 void kontNustatymas(string &konteineris, int &kont);
00130 void readFromFileLst(const string &failas, list<stud>& lst1);
00131 void skirstymasLst(const list<stud>& lst1, list<stud>& kietiakai, list<stud>& vargsiukai);
00132 void isvedimasLst(const string pavadinimas, const list<stud>& lst1);
00133 void skirstymas2str(vector<stud>& vec, vector<stud>& vargsiukai);
00134 void measureTimeVec2str(const string filename, int stud_num, int pasirinkimas, string konteineris);
00135 void skirstymasLst2str(list<stud>& lst1, list<stud>& vargsiukai);
00136 void measureTimeLst2str(const string filename, int stud_num, int pasirinkimas, string konteineris);
00137 void strategija(int &strat);
00138 void kokiasstrategija(string filename,int strat, int kont, int pasirinkimas, const string& konteineris,
    int numeris);
00139 bool arvargsiukas(const stud& s);
00140 void skirstymas3str(vector<stud>& vec1, vector<stud>& vargsiukai);
00141 void measureTimeVec3str(const string filename, int stud_num, int pasirinkimas, string konteineris);
00142 void val(vector<stud>& vec);
00143 void val(stud& student);
00144 #endif // STUD_H_INCLUDED

```

6.7 Project2/stud.h File Reference

```
#include "lib.h"
```

Classes

- class [zmogus](#)
- class [stud](#)

Functions

- void [ived](#) ([stud](#) &lok)
- void [outputvid](#) ([stud](#) lok)
- double [vidurkis](#) ([stud](#) &lok)
- void [outputmed](#) ([stud](#) lok)
- double [mediana](#) ([stud](#) &lok)
- void [readFromFile](#) (const string &failas, vector< [stud](#) > &vec1)
- void [outputfile](#) ([stud](#) lok)
- void [isvedimas](#) (const string pavadinimas, const vector< [stud](#) > &vec1)
- int [random_number](#) ()
- void [skirstymas](#) (const vector< [stud](#) > &vec1, vector< [stud](#) > &kietiakai, vector< [stud](#) > &vargsiukai)
- void [generavimas](#) (const string filename, int stud_kiekis)
- void [measureTimeVec](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [measureTimeLst](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [pasirinkti](#) (int &pasirinkimas)
- void [pasirinktikon](#) (int &kont)
- void [kontNustatymas](#) (string &konteineris, int &kont)
- void [readFromFileLst](#) (const string &failas, list< [stud](#) > &lst1)
- void [skirstymasLst](#) (const list< [stud](#) > &lst1, list< [stud](#) > &kietiakai, list< [stud](#) > &vargsiukai)
- void [isvedimasLst](#) (const string pavadinimas, const list< [stud](#) > &lst1)
- void [skirstymas2str](#) (vector< [stud](#) > &vec, vector< [stud](#) > &vargsiukai)
- void [measureTimeVec2str](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [skirstymasLst2str](#) (list< [stud](#) > &lst1, list< [stud](#) > &vargsiukai)
- void [measureTimeLst2str](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [strategija](#) (int &strat)
- void [kokiastategija](#) (string filename, int strat, int kont, int pasirinkimas, const string &konteineris, int numeris)
- bool [arvargsiukas](#) (const [stud](#) &s)
- void [skirstymas3str](#) (vector< [stud](#) > &vec1, vector< [stud](#) > &vargsiukai)
- void [measureTimeVec3str](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [val](#) (vector< [stud](#) > &vec)
- void [val](#) ([stud](#) &student)

6.7.1 Function Documentation

6.7.1.1 arvargsiukas()

```
bool arvargsiukas (
    const stud & s)
```

6.7.1.2 generavimas()

```
void generavimas (
    const string filename,
    int stud_kiekis)
```

6.7.1.3 isvedimas()

```
void isvedimas (
    const string pavadinimas,
    const vector< stud > & vec1)
```

6.7.1.4 isvedimasLst()

```
void isvedimasLst (
    const string pavadinimas,
    const list< stud > & lst1)
```

6.7.1.5 ived()

```
void ived (
    stud & lok)
```

6.7.1.6 kokiastrategija()

```
void kokiastrategija (
    string filename,
    int strat,
    int kont,
    int pasirinkimas,
    const string & konteineris,
    int numeris)
```

6.7.1.7 kontNustatymas()

```
void kontNustatymas (
    string & konteineris,
    int & kont)
```

6.7.1.8 measureTimeLst()

```
void measureTimeLst (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.7.1.9 measureTimeLst2str()

```
void measureTimeLst2str (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.7.1.10 measureTimeVec()

```
void measureTimeVec (  
    const string filename,  
    int stud_num,  
    int pasirinkimas,  
    string konteineris)
```

6.7.1.11 measureTimeVec2str()

```
void measureTimeVec2str (  
    const string filename,  
    int stud_num,  
    int pasirinkimas,  
    string konteineris)
```

6.7.1.12 measureTimeVec3str()

```
void measureTimeVec3str (  
    const string filename,  
    int stud_num,  
    int pasirinkimas,  
    string konteineris)
```

6.7.1.13 mediana()

```
double mediana (  
    stud & lok)
```

6.7.1.14 outputfile()

```
void outputfile (  
    stud lok)
```

6.7.1.15 outputmed()

```
void outputmed (  
    stud lok)
```

6.7.1.16 outputvid()

```
void outputvid (  
    stud lok)
```

6.7.1.17 pasirinkti()

```
void pasirinkti (
    int & pasirinkimas)
```

6.7.1.18 pasirinktikont()

```
void pasirinktikont (
    int & kont)
```

6.7.1.19 random_number()

```
int random_number ()
```

6.7.1.20 readFromFile()

```
void readFromFile (
    const string & failas,
    vector< stud > & vec1)
```

6.7.1.21 readFromFileLst()

```
void readFromFileLst (
    const string & failas,
    list< stud > & lst1)
```

6.7.1.22 skirstymas()

```
void skirstymas (
    const vector< stud > & vec1,
    vector< stud > & kietiakai,
    vector< stud > & vargsiukai)
```

6.7.1.23 skirstymas2str()

```
void skirstymas2str (
    vector< stud > & vec,
    vector< stud > & vargsiukai)
```

6.7.1.24 skirstymas3str()

```
void skirstymas3str (
    vector< stud > & vec1,
    vector< stud > & vargsiukai)
```

6.7.1.25 skirstymasLst()

```
void skirstymasLst (
    const list< stud > & lst1,
    list< stud > & kietiakai,
    list< stud > & vargsiukai)
```

6.7.1.26 skirstymasLst2str()

```
void skirstymasLst2str (
    list< stud > & lst1,
    list< stud > & vargsiukai)
```

6.7.1.27 strategija()

```
void strategija (
    int & strat)
```

6.7.1.28 val() [1/2]

```
void val (
    stud & student)
```

6.7.1.29 val() [2/2]

```
void val (
    vector< stud > & vec)
```

6.7.1.30 vidurkis()

```
double vidurkis (
    stud & lok)
```

6.8 stud.h

[Go to the documentation of this file.](#)

```
00001 #ifndef STUD_H_INCLUDED
00002 #define STUD_H_INCLUDED
00003 #include "lib.h"
00004
00005 class zmogus {
00006     protected:
00007         string vardas, pavarde;
00008     public:
00009         string getVardas() const { return vardas; };
00010         string getPavarde() const { return pavarde; };
00011         void setVardas(string vardas) { this->vardas = vardas; };
00012         void setPavarde(string pavarde) { this->pavarde = pavarde; };
00013
00014         zmogus() {} //cout << "Zmogus sukurtas" << endl; kad pamatyti kurima zmogaus
00015         virtual ~zmogus() {}
00016 }
```

```

00017         virtual void skaiciuotiAmziu() = 0;
00018     };
00019
00020     class stud : public zmogus{
00021     private:
00022         vector<double>nd;
00023         double vid, med, egz, rez;
00024
00025     public:
00026
00027         void skaiciuotiAmziu() override {
00028             cout << "skaiciuojama"«endl;
00029         }
00030
00031         // Numatytais konstruktorius
00032         stud() : nd(), vid(0.0), med(0.0), egz(0.0), rez(0.0) {}
00033
00034         //getteriai
00035         vector<double> getNd() const { return nd; }
00036         double getEgz() const { return egz; }
00037         double getVid() const { return vid; }
00038         double getMed() const { return med; }
00039
00040         //seteriai
00041         void setNd(vector<double> nd) { this->nd = nd; }
00042         void setEgz(double egz) { this->egz = egz; }
00043         void setVid(double vid) { this->vid = vid; }
00044         void setMed(double med) { this->med = med; }
00045         void setRez(double rez) { this->rez = rez; }
00046
00047         void addNd(double grade) {
00048             nd.push_back(grade);
00049         }
00050         //kopiavimo operatorius
00051         stud& operator=(const stud& a) {
00052             if (this == &a) return *this;
00053             this->vardas = a.vardas;
00054             this->pavarde = a.pavarde;
00055             this->nd = a.nd;
00056             this->egz = a.egz;
00057             this->vid = a.vid;
00058             this->med = a.med;
00059             this->rez = a.rez;
00060             return *this;
00061         }
00062
00063         // Kopijavimo konstruktorius
00064         stud(const stud& other) {
00065             vardas = other.vardas;
00066             pavarde = other.pavarde;
00067             nd = other.nd;
00068             egz = other.egz;
00069             vid = other.vid;
00070             med = other.med;
00071             rez = other.rez;
00072         }
00073
00074         void setEgzFromLastNd() {
00075             if (!nd.empty()) {
00076                 egz = nd.back();
00077                 nd.pop_back();
00078             } else {
00079                 std::cerr << "Error: No grades in nd to assign to egz." << std::endl;
00080             }
00081         }
00082     }
00083
00084
00085     friend std::istream& operator>>(std::istream& in, stud& student) {
00086         std::cout << "enter name: ";
00087         in >> student.vardas;
00088         std::cout << "enter surname: ";
00089         in >> student.pavarde;
00090         return in;
00091     }
00092
00093     // void clearData() {
00094     //     vardas.clear();
00095     //     pavarde.clear();
00096     //     nd.clear();
00097     //     egz = 0.0;
00098     //     vid = 0.0;
00099     //     med = 0.0;
00100     //     rez = 0.0;
00101     // }
00102
00103

```

```

00104     friend std::ostream& operator<<(std::ostream& out, const stud& student) {
00105         out << std::left << std::setw(15) << student.vardas
00106             << std::left << std::setw(15) << student.pavarde
00107             << std::left << std::setw(15) << std::fixed << std::setprecision(2) << student.vid;
00108         return out;
00109     }
00110
00111     ~stud(){} // destruktorius
00112 };
00113
00114 void ived(stud &lok);
00115 void outputvid(stud lok);
00116 double vidurkis(stud &lok);
00117 void outputmed(stud lok);
00118 double mediana(stud &lok);
00119 void readFromFile(const string& failas, vector<stud>& vec1);
00120 void outputfile(stud lok);
00121 void isvedimas(const string pavadinimas, const vector<stud>& vec1);
00122 int random_number();
00123 void skirstymas(const vector<stud>& vec1, vector<stud>& kietiakai, vector<stud>& vargsiukai);
00124 void generavimas(const string filename, int stud_kiekis);
00125 void measureTimeVec(const string filename, int stud_num, int pasirinkimas, string konteineris);
00126 void measureTimeLst(const string filename, int stud_num, int pasirinkimas, string konteineris);
00127 void pasirinkti(int &pasirinkimas);
00128 void pasirinktikont(int &kont);
00129 void kontNustatymas(string &konteineris, int &kont);
00130 void readFromFileLst(const string &failas, list<stud>& lst1);
00131 void skirstymasLst(const list<stud>& lst1, list<stud>& kietiakai, list<stud>& vargsiukai);
00132 void isvedimasLst(const string pavadinimas, const list<stud>& lst1);
00133 void skirstymas2str(vector<stud>& vec, vector<stud>& vargsiukai);
00134 void measureTimeVec2str(const string filename, int stud_num, int pasirinkimas, string konteineris);
00135 void skirstymasLst2str(list<stud>& lst1, list<stud>& vargsiukai);
00136 void measureTimeLst2str(const string filename, int stud_num, int pasirinkimas, string konteineris);
00137 void strategija(int &strat);
00138 void kokiastrategija(string filename,int strat, int kont, int pasirinkimas, const string& konteineris,
    int numeris);
00139 bool arvargsiukas(const stud& s);
00140 void skirstymas3str(vector<stud>& vec1, vector<stud>& vargsiukai);
00141 void measureTimeVec3str(const string filename, int stud_num, int pasirinkimas, string konteineris);
00142 void val(vector<stud>& vec);
00143 void val(stud& student);
00144 #endif // STUD_H_INCLUDED

```

6.9 Project2/listui.cpp File Reference

```

#include "stud.h"
#include "lib.h"

```

Functions

- void [pasirinkti](#) (int &pasirinkimas)
- void [pasirinktikont](#) (int &kont)
- void [kontNustatymas](#) (string &konteineris, int &kont)
- void [readFromFileLst](#) (const string &failas, list< [stud](#) > &lst1)
- void [skirstymasLst](#) (const list< [stud](#) > &lst1, list< [stud](#) > &kietiakai, list< [stud](#) > &vargsiukai)
- void [isvedimasLst](#) (const string pavadinimas, const list< [stud](#) > &lst1)
- void [measureTimeLst](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)

6.9.1 Function Documentation

6.9.1.1 isvedimasLst()

```

void isvedimasLst (
    const string pavadinimas,
    const list< stud > & lst1)

```


6.9.1.2 kontNustatymas()

```
void kontNustatymas (
    string & konteineris,
    int & kont)
```

6.9.1.3 measureTimeLst()

```
void measureTimeLst (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.9.1.4 pasirinkti()

```
void pasirinkti (
    int & pasirinkimas)
```

6.9.1.5 pasirinktikont()

```
void pasirinktikont (
    int & kont)
```

6.9.1.6 readFromFileLst()

```
void readFromFileLst (
    const string & failas,
    list< stud > & lst1)
```

6.9.1.7 skirstymasLst()

```
void skirstymasLst (
    const list< stud > & lst1,
    list< stud > & kietiakai,
    list< stud > & vargsiukai)
```

6.10 src/listui.cpp File Reference

```
#include "stud.h"
#include "lib.h"
```

Functions

- void `pasirinkti` (int &pasirinkimas)
- void `pasirinktikont` (int &kont)
- void `kontNustatymas` (string &konteineris, int &kont)
- void `readFromFileLst` (const string &failas, list< `stud` > &lst1)
- void `skirstymasLst` (const list< `stud` > &lst1, list< `stud` > &kietiakai, list< `stud` > &vargsiukai)
- void `isvedimasLst` (const string pavadinimas, const list< `stud` > &lst1)
- void `measureTimeLst` (const string filename, int stud_num, int pasirinkimas, string konteineris)

6.10.1 Function Documentation

6.10.1.1 isvedimasLst()

```
void isvedimasLst (  
    const string pavadinimas,  
    const list< stud > & lst1)
```

6.10.1.2 kontNustatymas()

```
void kontNustatymas (  
    string & konteineris,  
    int & kont)
```

6.10.1.3 measureTimeLst()

```
void measureTimeLst (  
    const string filename,  
    int stud_num,  
    int pasirinkimas,  
    string konteineris)
```

6.10.1.4 pasirinkti()

```
void pasirinkti (  
    int & pasirinkimas)
```

6.10.1.5 pasirinktikont()

```
void pasirinktikont (  
    int & kont)
```

6.10.1.6 readFromFileLst()

```
void readFromFileLst (  
    const string & failas,  
    list< stud > & lst1)
```

6.10.1.7 skirstymasLst()

```
void skirstymasLst (  
    const list< stud > & lst1,  
    list< stud > & kietiakai,  
    list< stud > & vargsiukai)
```

6.11 Project2/main.cpp File Reference

```
#include "lib.h"  
#include "stud.h"
```

Functions

- int [main](#) ()

6.11.1 Function Documentation

6.11.1.1 main()

```
int main ()
```

6.12 src/main.cpp File Reference

```
#include "lib.h"  
#include "stud.h"
```

Functions

- int [main](#) ()

6.12.1 Function Documentation

6.12.1.1 main()

```
int main ()
```

6.13 Project2/strategijos.cpp File Reference

```
#include "stud.h"  
#include "lib.h"
```

Functions

- void `measureTimeLst2str` (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void `skirstymasLst2str` (list< `stud` > &lst1, list< `stud` > &vargsiukai)
- void `measureTimeVec2str` (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void `skirstymas2str` (vector< `stud` > &vec1, vector< `stud` > &vargsiukai)
- void `kokiastrategija` (string filename, int strat, int kont, int pasirinkimas, const string &konteineris, int numeris)
- void `measureTimeVec3str` (const string filename, int stud_num, int pasirinkimas, string konteineris)
- bool `arvargsiukas` (const `stud` &s)
- void `skirstymas3str` (vector< `stud` > &vec1, vector< `stud` > &vargsiukai)

6.13.1 Function Documentation

6.13.1.1 arvargsiukas()

```
bool arvargsiukas (  
    const stud & s)
```

6.13.1.2 kokiastrategija()

```
void kokiastrategija (  
    string filename,  
    int strat,  
    int kont,  
    int pasirinkimas,  
    const string & konteineris,  
    int numeris)
```

6.13.1.3 measureTimeLst2str()

```
void measureTimeLst2str (  
    const string filename,  
    int stud_num,  
    int pasirinkimas,  
    string konteineris)
```

6.13.1.4 measureTimeVec2str()

```
void measureTimeVec2str (  
    const string filename,  
    int stud_num,  
    int pasirinkimas,  
    string konteineris)
```

6.13.1.5 measureTimeVec3str()

```
void measureTimeVec3str (  
    const string filename,  
    int stud_num,  
    int pasirinkimas,  
    string konteineris)
```

6.13.1.6 skirstymas2str()

```
void skirstymas2str (
    vector< stud > & vec1,
    vector< stud > & vargsiukai)
```

6.13.1.7 skirstymas3str()

```
void skirstymas3str (
    vector< stud > & vec1,
    vector< stud > & vargsiukai)
```

6.13.1.8 skirstymasLst2str()

```
void skirstymasLst2str (
    list< stud > & lst1,
    list< stud > & vargsiukai)
```

6.14 src/strategijos.cpp File Reference

```
#include "stud.h"
#include "lib.h"
```

Functions

- void [measureTimeLst2str](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [skirstymasLst2str](#) (list< [stud](#) > &lst1, list< [stud](#) > &vargsiukai)
- void [measureTimeVec2str](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void [skirstymas2str](#) (vector< [stud](#) > &vec1, vector< [stud](#) > &vargsiukai)
- void [kokiastategija](#) (string filename, int strat, int kont, int pasirinkimas, const string &konteineris, int numeris)
- void [measureTimeVec3str](#) (const string filename, int stud_num, int pasirinkimas, string konteineris)
- bool [arvargsiukas](#) (const [stud](#) &s)
- void [skirstymas3str](#) (vector< [stud](#) > &vec1, vector< [stud](#) > &vargsiukai)

6.14.1 Function Documentation

6.14.1.1 arvargsiukas()

```
bool arvargsiukas (
    const stud & s)
```

6.14.1.2 kokiastrategija()

```
void kokiastrategija (
    string filename,
    int strat,
    int kont,
    int pasirinkimas,
    const string & konteineris,
    int numeris)
```

6.14.1.3 measureTimeLst2str()

```
void measureTimeLst2str (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.14.1.4 measureTimeVec2str()

```
void measureTimeVec2str (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.14.1.5 measureTimeVec3str()

```
void measureTimeVec3str (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.14.1.6 skirstymas2str()

```
void skirstymas2str (
    vector< stud > & vec1,
    vector< stud > & vargsiukai)
```

6.14.1.7 skirstymas3str()

```
void skirstymas3str (
    vector< stud > & vec1,
    vector< stud > & vargsiukai)
```

6.14.1.8 skirstymasLst2str()

```
void skirstymasLst2str (
    list< stud > & lst1,
    list< stud > & vargsiukai)
```

6.15 Project2/stud.cpp File Reference

```
#include "stud.h"
#include "lib.h"
```

Functions

- void `ived` (`stud` &`lok`)
- void `outputvid` (`stud` `lok`)
- void `outputmed` (`stud` `lok`)
- void `outputfile` (`stud` `lok`)
- void `val` (vector< `stud` > &`vec`)
- double `vidurkis` (`stud` &`lok`)
- double `mediana` (`stud` &`lok`)
- void `readFromFile` (const string &`failas`, vector< `stud` > &`vec1`)
- void `generavimas` (const string `filename`, int `stud_kiekis`)
- int `random_number` ()
- void `skirstymas` (const vector< `stud` > &`vec1`, vector< `stud` > &`kietiakai`, vector< `stud` > &`vargsiukai`)
- void `isvedimas` (const string `pavadinimas`, const vector< `stud` > &`vec1`)
- void `measureTimeVec` (const string `filename`, int `stud_num`, int `pasirinkimas`, string `konteineris`)
- void `strategija` (int &`strat`)

6.15.1 Function Documentation

6.15.1.1 generavimas()

```
void generavimas (
    const string filename,
    int stud_kiekis)
```

6.15.1.2 isvedimas()

```
void isvedimas (
    const string pavadinimas,
    const vector< stud > & vec1)
```

6.15.1.3 ived()

```
void ived (
    stud & lok)
```

6.15.1.4 measureTimeVec()

```
void measureTimeVec (  
    const string filename,  
    int stud_num,  
    int pasirinkimas,  
    string konteineris)
```

6.15.1.5 mediana()

```
double mediana (  
    stud & lok)
```

6.15.1.6 outputfile()

```
void outputfile (  
    stud lok)
```

6.15.1.7 outputmed()

```
void outputmed (  
    stud lok)
```

6.15.1.8 outputvid()

```
void outputvid (  
    stud lok)
```

6.15.1.9 random_number()

```
int random_number ()
```

6.15.1.10 readFromFile()

```
void readFromFile (  
    const string & failas,  
    vector< stud > & vec1)
```

6.15.1.11 skirstymas()

```
void skirstymas (  
    const vector< stud > & vec1,  
    vector< stud > & kietiakai,  
    vector< stud > & vargsiukai)
```


6.15.1.12 strategija()

```
void strategija (
    int & strat)
```

6.15.1.13 val()

```
void val (
    vector< stud > & vec)
```

6.15.1.14 vidurkis()

```
double vidurkis (
    stud & lok)
```

6.16 src/stud.cpp File Reference

```
#include "stud.h"
#include "lib.h"
```

Functions

- void `ived` (stud &lok)
- void `outputvid` (stud lok)
- void `outputmed` (stud lok)
- void `outputfile` (stud lok)
- void `val` (vector< stud > &vec)
- double `vidurkis` (stud &lok)
- double `mediana` (stud &lok)
- void `readFromFile` (const string &failas, vector< stud > &vec1)
- void `generavimas` (const string filename, int stud_kiekis)
- int `random_number` ()
- void `skirstymas` (const vector< stud > &vec1, vector< stud > &kietiakai, vector< stud > &vargsiukai)
- void `isvedimas` (const string pavadinimas, const vector< stud > &vec1)
- void `measureTimeVec` (const string filename, int stud_num, int pasirinkimas, string konteineris)
- void `strategija` (int &strat)

6.16.1 Function Documentation

6.16.1.1 generavimas()

```
void generavimas (
    const string filename,
    int stud_kiekis)
```

6.16.1.2 isvedimas()

```
void isvedimas (
    const string pavadinimas,
    const vector< stud > & vec1)
```

6.16.1.3 ived()

```
void ived (
    stud & lok)
```

6.16.1.4 measureTimeVec()

```
void measureTimeVec (
    const string filename,
    int stud_num,
    int pasirinkimas,
    string konteineris)
```

6.16.1.5 mediana()

```
double mediana (
    stud & lok)
```

6.16.1.6 outputfile()

```
void outputfile (
    stud lok)
```

6.16.1.7 outputmed()

```
void outputmed (
    stud lok)
```

6.16.1.8 outputvid()

```
void outputvid (
    stud lok)
```

6.16.1.9 random_number()

```
int random_number ()
```

6.16.1.10 readFromFile()

```
void readFromFile (
    const string & failas,
    vector< stud > & vec1)
```

6.16.1.11 skirstymas()

```
void skirstymas (
    const vector< stud > & vec1,
    vector< stud > & kietiakai,
    vector< stud > & vargsiukai)
```

6.16.1.12 strategija()

```
void strategija (
    int & strat)
```

6.16.1.13 val()

```
void val (
    vector< stud > & vec)
```

6.16.1.14 vidurkis()

```
double vidurkis (
    stud & lok)
```

6.17 README.md File Reference

Index

- ~stud
 - stud, [17](#)
- ~zmogus
 - zmogus, [21](#)
- addNd
 - stud, [17](#)
- arvargsiukas
 - strategijos.cpp, [44](#), [45](#)
 - stud.h, [28](#), [34](#)
- generavimas
 - stud.cpp, [47](#), [49](#)
 - stud.h, [28](#), [34](#)
- getEgz
 - stud, [17](#)
- getMed
 - stud, [17](#), [18](#)
- getNd
 - stud, [18](#)
- getPavarde
 - zmogus, [22](#)
- getVardas
 - zmogus, [22](#)
- getVid
 - stud, [18](#)
- include/lib.h, [25](#)
- include/stud.h, [27](#), [32](#)
- isvedimas
 - stud.cpp, [47](#), [49](#)
 - stud.h, [28](#), [34](#)
- isvedimasLst
 - listui.cpp, [40](#), [42](#)
 - stud.h, [28](#), [35](#)
- ived
 - stud.cpp, [47](#), [50](#)
 - stud.h, [28](#), [35](#)
- kokiastategija
 - strategijos.cpp, [44](#), [45](#)
 - stud.h, [28](#), [35](#)
- kontNustatymas
 - listui.cpp, [40](#), [42](#)
 - stud.h, [28](#), [35](#)
- listui.cpp
 - isvedimasLst, [40](#), [42](#)
 - kontNustatymas, [40](#), [42](#)
 - measureTimeLst, [41](#), [42](#)
 - pasirinkti, [41](#), [42](#)
 - pasirinktikont, [41](#), [42](#)
 - readFromFileLst, [41](#), [42](#)
 - skirstymasLst, [41](#), [42](#)
- main
 - main.cpp, [43](#)
- main.cpp
 - main, [43](#)
- measureTimeLst
 - listui.cpp, [41](#), [42](#)
 - stud.h, [29](#), [35](#)
- measureTimeLst2str
 - strategijos.cpp, [44](#), [46](#)
 - stud.h, [29](#), [35](#)
- measureTimeVec
 - stud.cpp, [47](#), [50](#)
 - stud.h, [29](#), [35](#)
- measureTimeVec2str
 - strategijos.cpp, [44](#), [46](#)
 - stud.h, [29](#), [36](#)
- measureTimeVec3str
 - strategijos.cpp, [44](#), [46](#)
 - stud.h, [29](#), [36](#)
- mediana
 - stud.cpp, [48](#), [50](#)
 - stud.h, [29](#), [36](#)
- Objektinis1, [1](#)
- operator<<
 - stud, [20](#)
- operator>>
 - stud, [20](#)
- operator=
 - stud, [18](#)
- outputfile
 - stud.cpp, [48](#), [50](#)
 - stud.h, [30](#), [36](#)
- outputmed
 - stud.cpp, [48](#), [50](#)
 - stud.h, [30](#), [36](#)
- outputvid
 - stud.cpp, [48](#), [50](#)
 - stud.h, [30](#), [36](#)
- pasirinkti
 - listui.cpp, [41](#), [42](#)
 - stud.h, [30](#), [36](#)
- pasirinktikont
 - listui.cpp, [41](#), [42](#)
 - stud.h, [30](#), [37](#)

- pavarde
 - zmogus, 23
- Project2/lib.h, 26
- Project2/listui.cpp, 40
- Project2/main.cpp, 43
- Project2/strategijos.cpp, 43
- Project2/stud.cpp, 47
- Project2/stud.h, 33, 38
- random_number
 - stud.cpp, 48, 50
 - stud.h, 30, 37
- readFromFile
 - stud.cpp, 48, 50
 - stud.h, 30, 37
- readFromFileLst
 - listui.cpp, 41, 42
 - stud.h, 30, 37
- README.md, 51
- setEgz
 - stud, 18
- setEgzFromLastNd
 - stud, 18, 19
- setMed
 - stud, 19
- setNd
 - stud, 19
- setPavarde
 - zmogus, 22
- setRez
 - stud, 19
- setVardas
 - zmogus, 22
- setVid
 - stud, 19
- skaiciuotiAmziu
 - stud, 20
 - zmogus, 22, 23
- skirstymas
 - stud.cpp, 48, 51
 - stud.h, 30, 37
- skirstymas2str
 - strategijos.cpp, 44, 46
 - stud.h, 31, 37
- skirstymas3str
 - strategijos.cpp, 45, 46
 - stud.h, 31, 37
- skirstymasLst
 - listui.cpp, 41, 42
 - stud.h, 31, 37
- skirstymasLst2str
 - strategijos.cpp, 45, 46
 - stud.h, 31, 38
- src/listui.cpp, 41
- src/main.cpp, 43
- src/strategijos.cpp, 45
- src/stud.cpp, 49
- strategija
 - stud.cpp, 48, 51
 - stud.h, 31, 38
- strategijos.cpp
 - arvargsiukas, 44, 45
 - kokiastategija, 44, 45
 - measureTimeLst2str, 44, 46
 - measureTimeVec2str, 44, 46
 - measureTimeVec3str, 44, 46
 - skirstymas2str, 44, 46
 - skirstymas3str, 45, 46
 - skirstymasLst2str, 45, 46
- stud, 15
 - ~stud, 17
 - addNd, 17
 - getEgz, 17
 - getMed, 17, 18
 - getNd, 18
 - getVid, 18
 - operator<<, 20
 - operator>>, 20
 - operator=, 18
 - setEgz, 18
 - setEgzFromLastNd, 18, 19
 - setMed, 19
 - setNd, 19
 - setRez, 19
 - setVid, 19
 - skaiciuotiAmziu, 20
 - stud, 16, 17
- stud.cpp
 - generavimas, 47, 49
 - isvedimas, 47, 49
 - ived, 47, 50
 - measureTimeVec, 47, 50
 - mediana, 48, 50
 - outputfile, 48, 50
 - outputmed, 48, 50
 - outputvid, 48, 50
 - random_number, 48, 50
 - readFromFile, 48, 50
 - skirstymas, 48, 51
 - strategija, 48, 51
 - val, 49, 51
 - vidurkis, 49, 51
- stud.h
 - arvargsiukas, 28, 34
 - generavimas, 28, 34
 - isvedimas, 28, 34
 - isvedimasLst, 28, 35
 - ived, 28, 35
 - kokiastategija, 28, 35
 - kontNustatymas, 28, 35
 - measureTimeLst, 29, 35
 - measureTimeLst2str, 29, 35
 - measureTimeVec, 29, 35
 - measureTimeVec2str, 29, 36
 - measureTimeVec3str, 29, 36
 - mediana, 29, 36

- outputfile, [30](#), [36](#)
- outputmed, [30](#), [36](#)
- outputvid, [30](#), [36](#)
- pasirinkti, [30](#), [36](#)
- pasirinktikon, [30](#), [37](#)
- random_number, [30](#), [37](#)
- readFromFile, [30](#), [37](#)
- readFromFileLst, [30](#), [37](#)
- skirstymas, [30](#), [37](#)
- skirstymas2str, [31](#), [37](#)
- skirstymas3str, [31](#), [37](#)
- skirstymasLst, [31](#), [37](#)
- skirstymasLst2str, [31](#), [38](#)
- strategija, [31](#), [38](#)
- val, [31](#), [38](#)
- vidurkis, [31](#), [38](#)

val

- stud.cpp, [49](#), [51](#)
- stud.h, [31](#), [38](#)

vardas

- zmogus, [23](#)

vidurkis

- stud.cpp, [49](#), [51](#)
- stud.h, [31](#), [38](#)

zmogus, [21](#)

- ~zmogus, [21](#)
- getPavarde, [22](#)
- getVardas, [22](#)
- pavarde, [23](#)
- setPavarde, [22](#)
- setVardas, [22](#)
- skaiciuotiAmziu, [22](#), [23](#)
- vardas, [23](#)
- zmogus, [21](#)