

Objektinis Programavimas

2-oji praktinė užduotis



© Remigijus Paulavičius [@RemigPau](#), Vilniaus Universitetas - 2018



Darbas su duomenimis

Praktinės užduoties formuliuotė

- ☞ Parašykite programą, kuri nuskaito šiuos studentų duomenis:
 - ☞ **vardą** ir **pavardę**
 - ☞ n atliktų **namų darbų** (nd) rezultatus (10-balėje sistemoje), o taip pat galutinio **egzamino** (egz) rezultataj.
 - ☞ Tuomet iš šių duomenų, suskaičiuoja galutinį balą (galutinis):

$$\text{galutinis} = 0.4 \times \frac{\sum_{i=1}^n \text{nd}_i}{n} + 0.6 \times \text{egz}$$

Reikalavimai (v0.1) versijai (1)

- ☞ Pagal užduoties reikalavimus relizuokite programą ir atspausdinkite ekrane aktualią informaciją:
 - ☞ studento vardą ir pavardę,
 - ☞ namų darbų ir egzamino rezultatus
 - ☞ bei galutinį balą **dviejų skaičių po kablelio tikslumu**
- ☞ Papildykite programą, kad vietoj **vidurkio** būtų galima naudoti **medianą**.

Reikalavimai (v0.1) versijai (2)

- ☞ Papildykite programą taip, kad ji veiktu ir tokiu atveju, kai namų darbuų skaičius (n) yra nežinomas iš anksto, t.y. tik įvedimo metu vartotojas nusprendžia kuomet jis jau įvedė visu namų darbuų rezultatus. Šią užduotį realizuokite dviem būdais:
- ☞ naudojant C masyvus.
- ☞ naudojant `vector` ar kito tipo konteinerį.
- ☞ Papildykite, kad būtų galimybė, jog mokinio gautieji balai už namų darbus bei egzaminą būtų generuojami atsitiktinai.

Reikalavimai (v0.2) versijai (1)

Terminas: 2018-02-27

- ☞ Papildykite (v0.1) taip, kad būtų galima duomenis ne tik ivesti bet ir nuskaityti iš failų, t.y., sukurkite ir užpildykite failą **kursiokai.txt**, kurio (pleriminari) struktūra:

Pavardė	Vardas	ND1	ND2	ND3	ND4	ND5	Egzaminas
Pocius	Paulius	8	9	10	6	10	9
Makevičius	Augustinas	7	10	8	5	4	6
...							

Reikalavimai (v0.2) versijai (2)

Terminas: 2018-02-27

- ☞ Papildykite programą taip, nuskaičiuos duomenis iš failo, galutinis rezultatas (output'as) pleriminariai atrodytų taip:

Pavardė	Vardas	Gal. – vidurkis	Gal. – mediana
Makevičius	Augustinas	x . xx	y . yy
Pocius	Paulius	z . zz	q . qq
...			

Reikalavimai output'ui: studentai turi būtui surūšiuoti pagal vardus (ar pavardes) ir stulpeliai būtų gražiai "išlygiuoti".

Reikalavimai (v0.3) versijai (1)

Preliminarus terminas: 2018-03-01

- 👉 Atlikite versijos (v0.2) kodo reorganizavimą (refactoring'ą):
 - 👉 Kur tikslina, programoje naudokite (jeigu dar nenaudojote) struct'ūras
 - 👉 Funkcijas, naujus duomenų tipus (struct'ūras) perkelkite į antraštinius (**header** (*.h)) failus, t.y. tokiu būdu turėtumete projekte turéti kelis *.cpp failus, kaip ir kelis *.h failus.

Reikalavimai (v0.3) versijai (2)

Preliminarus terminas: 2018-03-01

👉 Panaudokite išimčių valdymą (**Exception Handling**):

```
try { // išimtys yra apdorojamos žemiau
      // kodas, kuris atlieka tam tikras užduotis
} catch (std::exception& e) {
      // kodas, kuris apdoroja išimtis
}
```

Kam viso to reikia? Kas atsitiks, jei failas, kuri bandote atidaryti neegzistuoja; arba bandote gauti masyvo elementą, kurio nėra?

Reikalavimai (v0.4) versijai (1)

Preliminarus terminas: 2018-03-08

- 👉 Programos veikimo greičio (spartos) analizė:
- 👉 Sūrušiuokite (padalinkite) studentus į dvi kategorijas:
 - 👉 Studentai, kurie **surinko < 60%** už namų darbų užduotis ("vargšiukai", "nuskriaustukai" ir pan.)
 - 👉 Studentai, kurie **surinko >= 60%** ("kietiakai", "galvočiai", "vizunčikai" ir pan.) ir buvo prileisti prie egzamino.

Versijos (v0.1) kūrimas (1)

Skaičių po kablelio tikslumo valdymas

- 👉 <iomanip> antraštės (header) faile deklaruotas manipulatorius setprecision, kuris leidžia kontroliuoti kiek reikšmingų skaitmenų išvedima informacija naudotų.
- 👉 Kai naudojame endl, kuris taip pat yra manipulatorius, mums nereikia ištraukti <iomanip> antraštės. **Kodėl?**

Versijos (v0.1) kūrimas (2)

std::setprecision pavyzdys¹

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <limits>
int main() {
    const long double pi = std::acos(-1.L);
    std::cout << "default precision (6): " << pi << '\n'
        << "std::setprecision(10): " << std::setprecision(10) << pi << '\n'
        << "max precision: "
        << std::setprecision(std::numeric_limits<long double>::digits10 + 1) << pi << '\n';
}
```

```
default precision (6): 3.14159
std::setprecision(10): 3.141592654
max precision:      3.141592653589793239
```

¹ <http://en.cppreference.com/w/cpp/io/manip/setprecision>

Versijos (v0.1) kūrimas (3)

Rezultatų spausdinimas 2 skaičių po kablelio tikslumu

```
// išsaugome numatytaį (default) tikslumą
streamszie prec = cout.precision(); // C++11: auto prec
cout << "Galutinis balas yra " << setprecision(3)
      << 0.6 * egzaminas + 0.4 * suma / n
      << setprecision(prec) << endl;
```

Versijos (v0.1) kūrimas (4)

EOF (end-of-file) signalas

- 👉 Vienas iš būdų, kaip užbaigti duomenų įvedimą (cikle) yra **end-of-file (EOF)** signalo pasiuntimas.
- 👉 Skirtingos C++ realizacijos siūlo skirtinges būdus kaip tokį signalą į programą pasiūsti. Labiausiai paplitęs būdas yra:
 - 👉 pradėti naują eilutę (spaudžiame Enter)
 - 👉 tuomet spaudžiame Ctrl+z (Windows) arba Ctrl+d (Unix).

Versijos (v0.1) kūrimas (5)

EOF (end-of-file) pavyzdžiai²

```
while( !cin.eof() ) { // EOF false čia
    cin >> x;          // kai nuskaityti nepavyksta, EOF tampa true
    // use x           // naudojame x, nors nieko nenuskaitėme
}
```

VS.

```
while(cin >> x) { // Bando nuskaityti į x, grąžina false kai nepavyksta
                    // ciklą vykdys tik kai nuskaityti pavyko
}
```

² <https://stackoverflow.com/questions/4533063/how-does-ifstreams-eof-work>

Versijos (v0.1) kūrimas (6)

while (cin >> x) reiškinio analizė

Norint nuskaityti reikšmę į **x** ir patikrinti ar pavyko:

```
if (cin >> x) { /*...*/ }
```

Tai ekvivalentu:

```
cin >> x;  
if (cin) { /* ... */ }
```

Todėl cin naudojimas **if** ar **while** sąlygose ekvivalentus: ar paskutinis bandymas nuskaityti iš `cin buvo sėkmingas.

Versijos (v0.1) kūrimas (7)

Kada skaitymas iš srauto (cin) gali būti nesėkminges?

- 👉 Mes galėjome pasiekti įvesties failo pabaigą.
- 👉 Galbūt susidūrēme su įvestimi, nesuderinama su kintamojo tipu, kurių bandome perskaityti, pvz., taip gali atsitikti, jei mes bandome perskaityti int bet reikšmę yra nėra skaičius.
- 👉 Sistemoje gali būti aptikta įvesties įrangos gedimas.

Visais šiai atvejais: naudojant sąlygoje cin reikšmę reiks false.

Versijos (v0.1) kūrimas (8)

Nežinomo skaičiaus namų darbų (nd) nuskaitymas

```
int n = 0;           // nd skaitiklis
double suma = 0.;   // nd įverčiu suma
double x;           // kintamasis i kuri nuskaityti
/* invariantas: iki šiol nuskaitėme `n' nd rezultatu,
   ir jų įverčiu suma lygi `suma' */
while (cin >> x) {
    ++n;             // reikalauja pirmoji dalis invarianto
    suma += x;        // reikalauja antroji dalis invarianto
}
```

Klausimai!

!?

