

# Programavimas

Šaltinis: [Informacinių technologijų valstybinio brandos egzamino programa](#)

# 4.1. Pradiniams duomenims ir rezultatams laikyti (saugoti) naudoti tekstinius failus.



4.1.1. Žinoti tekstinių failų paskirtį.

4.1.2. Parengti failą duomenų įvedimui ir rašymui.

4.1.3. Skaityti duomenis iš failo.

4.1.4. Rašyti duomenis į failą.

# Skaitymas iš failo, rašymas į failą.

## IT VBE vertintojų pastebėjimai

Nurodomas pilnas kelias (jau rečiau, 2019 m. egzamine buvo tik keli atvejai):

```
ifstream fd("C:/Egzaminas/U1.txt");
```

```
ofstream fr("C:/Egzaminas/U1rez.txt");
```

Turėtų būti:

```
ifstream fd("U1.txt");
```

```
ofstream fr("U1rez.txt");
```

# Skaitymas iš failo, kai yra eilutės (string) tipo duomenų

15		
Vilnius	Vilniaus	541278
Dusetos	Utenos	4211
Alytus	Alytaus	69859
Druskininkai	Alytaus	16890
Ignalina	Utenos	6307
Kavarskas	Utenos	753
Lazdijai	Alytaus	5027
Simnas	Alytaus	1940
.....	.....	.....

```
void Skaitymas(Miestas M[], int &n)
{
    char eil1[21];
    char eil2[14];
    ifstream fd("U2.txt");
    fd >> n;
    fd.ignore();
    for(int i = 0; i < n; i++)
    {
        fd.get(eil1, 20); M[i].pav = eil1; fd.get(eil2, 14);
        M[i].apskr = eil2; fd >> M[i].gs;
        fd.ignore();
    }
    fd.close();
}
```

## 4.2. Aprašyti veiksmus procedūromis arba funkcijomis, taikyti jas sudarant programą.



4.2.1. Analizuoti užduotį, skaidyti ją į struktūrinės dalis.

4.2.2. Paaiškinti procedūras ir / ar funkcijos sąvoką, paskirtį.

4.2.3. Išskirti uždavinio dalis, kurias galima užrašyti procedūromis arba funkcijomis.

4.2.4. Naudoti parametrus rašant procedūras arba funkcijas.

4.2.5. Rašant programą naudoti procedūras ir / ar funkcijas.

4.2.6. Žinoti programos, procedūrų ir / ar funkcijų vardų galiojimo sritis.

# 2019 m. egzamino 2 užduotis – puikus pavyzdys, iliustruojantis funkcijų galią



- Užduoties sąlyga
- Netinkamo sprendimo pavyzdys
- Tinkamo sprendimo pavyzdys



# Dažniausios klaidos užrašant sumos (kiekio) skaičiavimo funkciją, gražinančią apskaičiuotą reikšmę per funkcijos vardą

```
int Suma (int A[], int a)
{
    int s = 0;
    for (int i = 0; i < a; i++)
        s = s + A[i];
    return s;
}
```

# Dažniausios klaidos užrašant vidurkio skaičiavimo funkciją, gražinančią apskaičiuotą reikšmę per funkcijos vardą

```
double Vidurkis (int A[], int a)
{
    int s = 0;
    for (int i = 0; i < a; i++)
        s = s + A[i];
    return (double) s / a;
}
```



# Vietos, kurioje yra didžiausia / mažiausia reikšmė, paieškos funkcija

Jei reikia rasti pirmą eilęje

```
int Vieta (int A[], int a)
{
    int k = 0;
    for (int i = 0; i < a; i++)
        if (A[i] > A[k]) k = i;
    return k;
}
```

```
int Vieta (int A[], int a)
{
    int k = 0;
    for (int i = 0; i < a; i++)
        if (A[i] < A[k]) k = i;
    return k;
}
```

Jei reikia rasti paskutinį eilęje

```
int Vieta (int A[], int a)
{
    int k = 0;
    for (int i = 0; i < a; i++)
        if (A[i] >= A[k]) k = i;
    return k;
}
```

```
int Vieta (int A[], int a)
{
    int k = 0;
    for (int i = 0; i < a; i++)
        if (A[i] <= A[k]) k = i;
    return k;
}
```

# Funkcija su parametrais-nuorodomis: k-tojo elemento reikšmės šalinimas iš masyvo

```
void Salinti (int A[], int k, int & a)  
{  
    for (int i = k; i < a; i++)  
        A[i] = A[i-1];  
    a = a - 1;  
}
```

# Rikiavimo funkcija

```
void Rikiavimas (int A[], int a)
{
    int m; int keitimas;
    for (int i = 0; i < a - 1; i++)
    {
        m = i;
        for (int j = i + 1; j < a; j++)
            if (A[j] < A[m]) m = j;
        keitimas = A[i]; A[i] = A[m]; A[m] = keitimas;
    }
}
```

```
void Rikiavimas (int A[], int a)
{
    int m;
    for (int i = 0; i < a - 1; i++)
    {
        m = i;
        for (int j = i + 1; j < a; j++)
            if (A[j] < A[m]) m = j;
        swap(A[i], A[m]);
    }
}
```

## 4.3. Suformuoti masyvą ir atlikti veiksmus su jo duomenimis.



4.3.1. Skirti masyvo elemento indeksą ir masyvo elemento reikšmę.

4.3.2. Aprašyti skaičių ir simbolių masyvo duomenų tipą, kintamiesiems aprašyti naudoti masyvo tipo kintamuosius.

4.3.3. Suteikti masyvo elementams reikšmes (priskyrimas, skaitymas iš failo).

4.3.4. Atlikti veiksmus su masyvo elementų reikšmėmis.

4.3.5. Naudoti masyvo tipo parametrus procedūrose arba funkcijose.

## 4.4. Atlikti veiksmus su simbolių eilutėmis.



4.4.1. Aprašyti simbolių eilutės duomenų tipo kintamuosius.

4.4.2. Atlikti veiksmus su simbolių eilučių kintamųjų reikšmėmis: įvesti, išvesti, palyginti.

## 4.5. Atlikti veiksmus su įrašais (struktūromis) ir masyvais su įrašo (struktūros) tipo elementais.

4.5.1. Aprašyti įrašo (struktūros) duomenų tipą, masyvo su įrašo (struktūros) tipo elementais tipą.

4.5.2. Suteikti įrašo (struktūros) tipo kintamiesiems reikšmes (įvesti duomenis), išvesti.

4.5.3. Atlikti veiksmus su įrašo (struktūros) tipo kintamųjų reikšmėmis.

4.5.4. Atlikti veiksmus su įrašo (struktūros) tipo masyvo elementų reikšmėmis.

4.5.5. Sudaryti nesudėtingas duomenų struktūras, panaudoti įrašo (struktūros) ir masyvo duomenų tipus.

# Į ką atkreipti dėmesį aprašant duomenis? IT VBE vertintojų pastebėjimai



- Neatidžiai perskaityta užduoties sąlyga (o gal mokinys nesupranta kuo skiriasi **double**, **int** duomenų tipai?) ir apibrėžiant pasirenkamas netinkamas duomenų tipas.
- Masyvo apraše nurodomas per mažas masyvo elementų reikšmių skaičius.
- Struktūros apraše: netinkamai nurodomi kintamųjų rinkinio elementų tipai, sintaksės klaidos.

## 4.6. Sudaryti uždavinių programos.

4.6.1. Parinkti uždaviniui spręsti tinkamas duomenų struktūras ir algoritmus.

4.6.2. Programose naudoti algoritmus: sumos, sandaugos, kiekio, aritmetinio vidurkio skaičiavimo; rikiavimo; didžiausios (mažiausios) reikšmės radimo, paieškos surikiuotame ir nerikiuotame masyve.

4.6.3. Parinktą, modifikuotą ar sudarytą algoritmą užrašyti programa.

4.6.4. Kurti programas įvairiems taikomiesiems uždaviniams spręsti.



## 4.7. Įvaldyti programavimo technologijos (metodikos) elementus.

4.7.1. Žinoti programos sudarymo etapus, laikytis jų rašant programas.

4.7.2. Parengti kontrolinius duomenis programos darbo teisingumui patikrinti.

4.7.3. Testuoti programą ir taisyti pastebėtas klaidas.

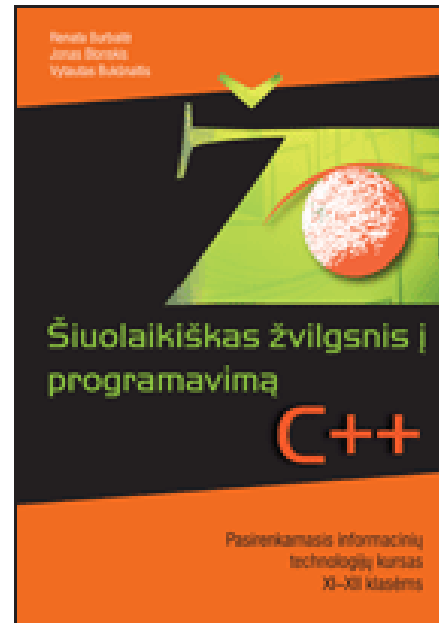
4.7.4. Komentuoti programos dalis.

# Į ką svarbiausia atkreipti dėmesį atliekant programavimo užduotis?

Programa = Duomenų struktūros + algoritmai

- Labai gerai apgalvoti, kokias duomenų struktūras ir algoritmus naudosite.
- Labai atidžiai išnagrinėti užduoties sąlygą ir reikalavimus.
- **Pirmiausia samprotaudami iš pradinių duomenų gaukite rezultatus, aiškiai apibrėžkite sprendimo žingsnius ir tik tada pradėkite rašyti programą.**

# Programavimo užduočių rinkiniai



Pradinių duomenų failai:

<http://it.vadoveliai.lt/pem/>

arba iš archyvo:

[IT\\_VBE\\_kartojimas](#)

- [KTU programavimo pamokėlių 2013-2014 m.m. užduočių rinkinys](#)
- [KTU programavimo pamokėlių 2014-2015 m.m. užduočių rinkinys](#)
- [KTU programavimo pamokėlių 2015-2016 m.m. užduočių rinkinys](#)
- [KTU programavimo pamokėlių 2016-2017 m.m. užduočių rinkinys](#)
- [KTU programavimo pamokėlių 2017-2018 m.m. užduočių rinkinys](#)
- [KTU programavimo pamokėlių 2018-2019 m.m. užduočių rinkinys](#)

← → ↻ aity.pikta.lt/t/

Apps Bookmarks Nonlinear predictiv... „Gmail“: el. pa

**Uždaviniai** Laidos Visuotinė ataskaita

- + Įkelti uždavinius
- 11s\_f\_1lygis\_kateriai
- 11s\_f\_1lygis\_kiek\_yra\_tokiu
- 11s\_f\_1lygis\_mokytojos\_pac
- 11s\_f\_1lygis\_obelu\_sodas
- 11s\_f\_1lygis\_pazymiu\_vidur

## Uždavinio aprašymas

Sprendimo kodas

Komentaras

Įkelti sprendimą

# Pasitreniruojam?



- [Pirmos užduoties pavyzdžiai](#)
- [Antros užduoties pavyzdžiai](#)

