

# Spausdinamų elementų formatai

- Išmoksite nurodyti formatus spausdinamoms kintamųjų reikšmėms.

## TEORINĖ DALIS

Programos pradiniai duomenys ir skaičiavimo rezultatai gali būti pateikti pačia įvairiausia forma arba formatu. Jie turi tenkinti vartotojo keliamus reikalavimus, kaip pavyzdžiui tikslumas. Paimkite paprasčiausią skaičiuotuvą ir padalinkite 1 iš 17 (tegu bus tokios dviejų kintamųjų reikšmės). Skaičiuotuvas parodė rezultatą 0,0588235294117647 (16 skaitmenų po kablelio). Tačiau gal toks tikslumas nėra reikalingas, gal pakanka parodyti tik 4 skaitmenis po kablelio.

Geometriniuose skaičiavimuose dažnai prisieina naudoti konstantą  $\pi$  (apskritimo ilgio ir skersmens santykis) 3.141592653589.. Daugelis žino apytikslę šios skaičiaus reikšmę 3.14. Vieniems skaičiavimams pakanka 4 skaitmenų po kablelio, kitiems, kaip pvz., kosminiams skaičiavimams reikia ir keturiasdešimties skaitmenų po kablelio tikslumo. Taigi, programoje yra galimybė pateikti realiuosius skaičius norimu tikslumu. Spausdinant realiuosius skaičius reikia įvertinti, kiek visam skaičiui reikia išskirti vietos, įskaitant skaičiaus ženklą (minus), sveikąją dalį, trupmeninę dalį (tikslumas) ir kablelį (arba tašką).

Su sveikaisiais skaičiais yra paprasčiau, čia reikia įvertinti kokiose ribose galės būti skaičiavimo rezultatas, taip pat įskaitant ženklą (minus).

Spausdinant sveikąjį skaičių ar kintamojo reikšmę (pvz., sveikasis) naudojamas toks formatas: tarp dvigubų kabučių (" ") riestiniuose skliaustuose ({ }) pirmiausiai nurodoma spausdinamo kintamojo vieta, kablelio simbolis , ir pozicijų skaičius n už kurio seka dvitaškis ir simbolis d:

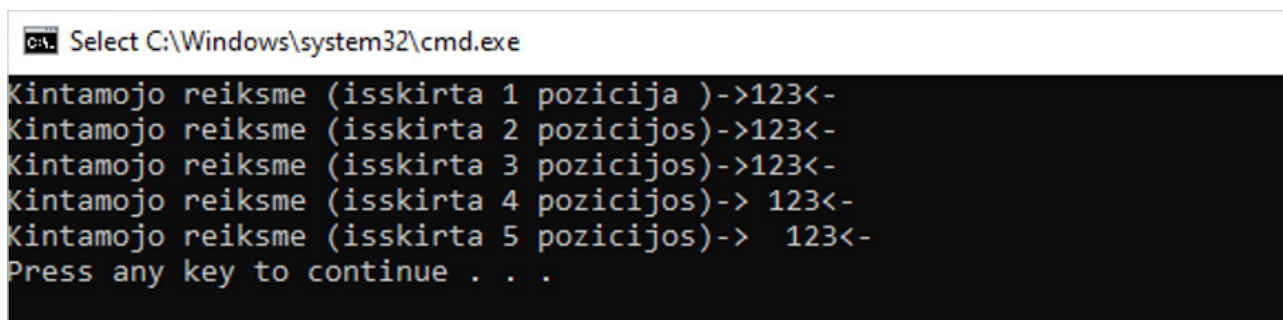
```
int sveikas = 123;  
Console.WriteLine("{0, 3:d}", sveikas);
```

Kintamojo vietos numeracija visuomet prasideda nuo 0 (nulinio)! Kintamajam skiriamų pozicijų skaičius n nurodomas pagal poreikį: 1, 2, 3, 4, 5 ir t. t.

Pasižiūrėkite, kaip atrodo sveikąjo kintamojo reikšmė išspausdinta naudojant penkis skirtingus formatus:

```
Console.WriteLine("Kintamojo reikšmė (išskirta 1 pozicija )->{0, 1:d}<-", sveikas);  
Console.WriteLine("Kintamojo reikšmė (išskirta 2 pozicijos)->{0, 2:d}<-", sveikas);  
Console.WriteLine("Kintamojo reikšmė (išskirta 3 pozicijos)->{0, 3:d}<-", sveikas);  
Console.WriteLine("Kintamojo reikšmė (išskirta 4 pozicijos)->{0, 4:d}<-", sveikas);  
Console.WriteLine("Kintamojo reikšmė (išskirta 5 pozicijos)->{0, 5:d}<-", sveikas);
```

Įvykdę tokį programos fragmentą ekrane matysite:



```
C:\Windows\system32\cmd.exe  
Kintamojo reiksme (isskirta 1 pozicija )->123<-  
Kintamojo reiksme (isskirta 2 pozicijos)->123<-  
Kintamojo reiksme (isskirta 3 pozicijos)->123<-  
Kintamojo reiksme (isskirta 4 pozicijos)-> 123<-  
Kintamojo reiksme (isskirta 5 pozicijos)->  123<-  
Press any key to continue . . .
```

Jeigu skaičius netelpa nurodytame formate jis vis tiek išspausdinamas ir užima tiek pozicijų kiek skaičius turi skaitmenų (plius ženklas, jeigu skaičius neigimas). Tai matosi išspausdintose pirmose dviejose eilutėse.

Kaip pastebite išspausdintas skaičius glaudžiamas prie dešinės pusės. Jeigu norite, kad spausdinamas skaičius būtų glaudžiamas prie kairės pusės, reikia formate prieš nurodomų pozicijų skaičių užrašyti minuso ženklą (-). Pvz., penktoje eilutėje padėkite minus ženklą ir pasižiūrėkite kaip pasikeis 5-os eilutės rezultatas įvykdžius programos fragmentą:

```
Console.WriteLine("Kintamojo reikšmė (išskirta 5 pozicijos)->{0, -5:d}<- ", sveikas);
```

```
C:\Windows\system32\cmd.exe
Kintamojo reiksme (isskirta 1 pozicija )->123<-
Kintamojo reiksme (isskirta 2 pozicijos)->123<-
Kintamojo reiksme (isskirta 3 pozicijos)->123<-
Kintamojo reiksme (isskirta 4 pozicijos)-> 123<-
Kintamojo reiksme (isskirta 5 pozicijos)->123 <-
Press any key to continue . . .
```

Išspausdinkime trijų sveikojo tipo kintamųjų reikšmes vienoje eilutėje skirtingais formatais:

```
int sveikas0 = -255,
    sveikas1 = 0,
    sveikas2 = 17;

Console.WriteLine("Kintamųjų reikšmės->{0, -7:d}<-->{1, 3:d}<-->{2, 5:d}<- ",
    sveikas0, sveikas1, sveikas2);
```

Rezultatas ekrane:

```
C:\Windows\system32\cmd.exe
Kintamųjų reikšmės->-255 <--> 0<--> 17<-
Press any key to continue . . .
```

Kintamojo sveikas0 reikšmė spausdinama naudojant formatą `{0, -7:d}`, kintamojo sveikas1 reikšmė - `{1, 3:d}`, kintamojo sveikas2 - `{2, 5:d}`.

Spausdinant realųjį skaičių ar kintamojo reikšmę (pvz., `realus`) naudojamas toks formatas (panašus į sveikojo skaičiaus spausdinimo formatą): tarp dvigubų kabučių (" ") riestiniuose skliaustuose ({ }) pirmiausiai nurodoma spausdinamo kintamojo vieta, kabelio simbolis , ir pozicijų skaičius n, už kurio seka dvitaškis, simbolis f ir skaičius m:

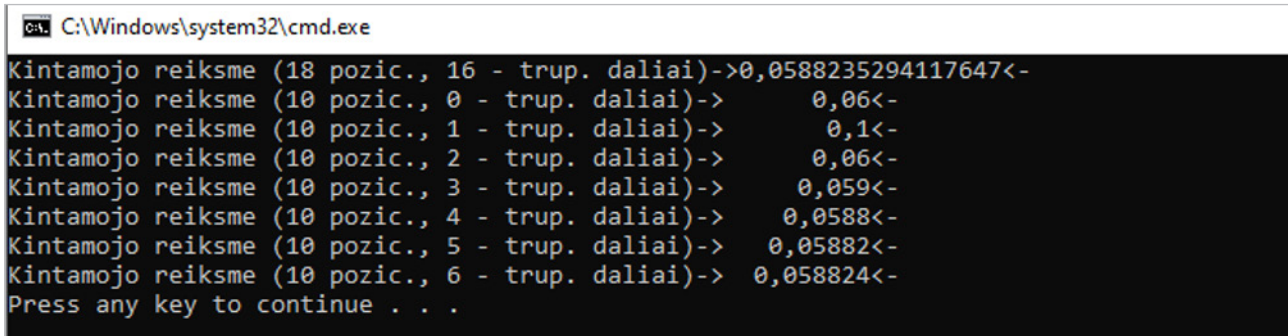
```
int realus = 1.0 / 17;
Console.WriteLine("{0, 9:f5}", realus);
```

Skaičius n nurodo, kiek iš viso pozicijų skiriama realiajam skaičiui, skaičius m – kiek pozicijų skiriama skaičiaus trupmeninei daliai. Šiuo atveju n yra 9, o m – 5.

Pasižiūrėkite, kaip atrodo realiojo tipo kintamojo reikšmė išspausdinta naudojant aštuonis skirtingus formatus:

```
double realus = 1.0 / 17; // 0,0588235294117647 - suskaičiuota skaičiuotuvu
Console.WriteLine("Kintamojo reikšmė (18 pozic.,16 - trup. daliai)->{0, 18:f16}<- ", realus);
Console.WriteLine("Kintamojo reikšmė (10 pozic., 0 - trup. daliai)->{0, 10:f}<- ", realus);
Console.WriteLine("Kintamojo reikšmė (10 pozic., 1 - trup. daliai)->{0, 10:f1}<- ", realus);
Console.WriteLine("Kintamojo reikšmė (10 pozic., 2 - trup. daliai)->{0, 10:f2}<- ", realus);
Console.WriteLine("Kintamojo reikšmė (10 pozic., 3 - trup. daliai)->{0, 10:f3}<- ", realus);
Console.WriteLine("Kintamojo reikšmė (10 pozic., 4 - trup. daliai)->{0, 10:f4}<- ", realus);
Console.WriteLine("Kintamojo reikšmė (10 pozic., 5 - trup. daliai)->{0, 10:f5}<- ", realus);
Console.WriteLine("Kintamojo reikšmė (10 pozic., 6 - trup. daliai)->{0, 10:f6}<- ", realus);
```

Įvykdę tokį programos fragmentą ekrane matysite:



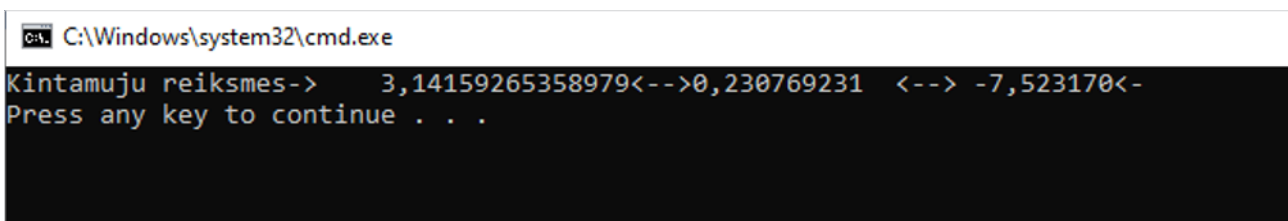
```
C:\Windows\system32\cmd.exe
Kintamojo reiksme (18 pozic., 16 - trup. daliai)->0,0588235294117647<-
Kintamojo reiksme (10 pozic., 0 - trup. daliai)-> 0,06<-
Kintamojo reiksme (10 pozic., 1 - trup. daliai)-> 0,1<-
Kintamojo reiksme (10 pozic., 2 - trup. daliai)-> 0,06<-
Kintamojo reiksme (10 pozic., 3 - trup. daliai)-> 0,059<-
Kintamojo reiksme (10 pozic., 4 - trup. daliai)-> 0,0588<-
Kintamojo reiksme (10 pozic., 5 - trup. daliai)-> 0,05882<-
Kintamojo reiksme (10 pozic., 6 - trup. daliai)-> 0,058824<-
Press any key to continue . . .
```

Atkreipkite dėmesį, kad spausdinamas skaičius apvalinamas. Be to, jeigu nėra nurodyta, kiek pozicijų skiriama skaičiaus trupmeninei daliai, numatytoji reikšmė yra dvi pozicijos, t. y. spausdinami du skaitmenys po kablelio.

Išspausdinkime trijų realaus tipo kintamųjų reikšmes vienoje eilutėje skirtingais formatais:

```
double realus0 = Math.PI,
       realus1 = 3 / 13.0,
       realus2 = -7.52317;
Console.WriteLine("Kintamųjų reikšmės->{0, 20:f14}<-->{1, -13:f9}<-->{2, 10:f6}<- ",
                 realus0, realus1, realus2);
```

Rezultatas ekrane:



```
C:\Windows\system32\cmd.exe
Kintamųjų reiksmes-> 3,14159265358979<-->0,230769231 <--> -7,523170<-
Press any key to continue . . .
```

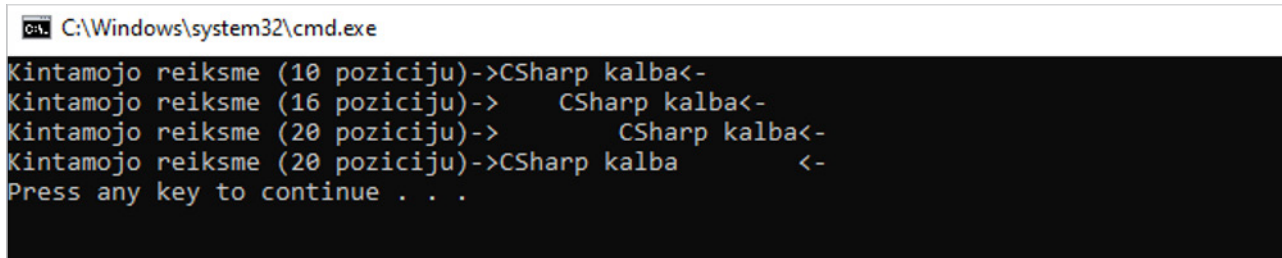
Kintamojo `realus0` reikšmė spausdinama naudojant formatą `{0, 20:f14}`, kintamojo `realus1` reikšmė - `{1, -13:f9}`, kintamojo `realus2` - `{2, 10:f6}`.

Spausdinant simbolių eilutę ar kintamojo reikšmę (pvz., eilute) naudojamas toks formatas (panašus į sveiką skaičiaus spausdinimo formatą): tarp dvigubų kabučių (" ") riestiniuose skliaustuose ({ }) pirmiausiai nurodoma spausdinamo kintamojo vieta, kablelio simbolis , ir pozicijų skaičius n:

Pasižiūrėkite, kaip atrodo eilutės tipo kintamojo reikšmė išspausdinta naudojant keturis skirtingus formatus:

```
string eilute = "CSharp kalba";
Console.WriteLine("Kintamojo reikšmė (10 pozicijų)->{0, 10}<- ", eilute);
Console.WriteLine("Kintamojo reikšmė (16 pozicijų)->{0, 16}<- ", eilute);
Console.WriteLine("Kintamojo reikšmė (20 pozicijų)->{0, 20}<- ", eilute);
Console.WriteLine("Kintamojo reikšmė (20 pozicijų)->{0, -20}<- ", eilute);
```

Įvykdę tokį programos fragmentą ekrane matysite:

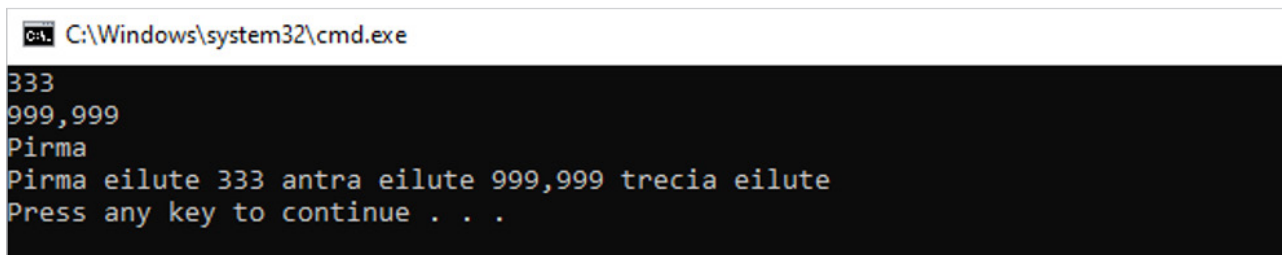


```
C:\Windows\system32\cmd.exe
Kintamojo reiksme (10 poziciju)->CSharp kalba<-
Kintamojo reiksme (16 poziciju)-> CSharp kalba<-
Kintamojo reiksme (20 poziciju)-> CSharp kalba<-
Kintamojo reiksme (20 poziciju)->CSharp kalba <-
Press any key to continue . . .
```

Galima spausdinti kintamųjų reikšmes nenaudojant formatavimo.

```
int n = 333;
Console.WriteLine(n);
double x = 999.999;
Console.WriteLine(x);
string eilute = "Pirma";
Console.WriteLine(eilute);
Console.WriteLine(eilute + " eilutė " + n + " antra eilutė " + x + " trečia eilutė");
```

Įvykdę tokį programos fragmentą ekrane matysite:



```
C:\Windows\system32\cmd.exe
333
999,999
Pirma
Pirma eilute 333 antra eilute 999,999 trecia eilute
Press any key to continue . . .
```

**UŽDUOTIS.** Žvejys pagavo 3 žuvis ir jas pasvėrė (svoris gramais). Parašykite programą, kuri suskaičiuotų kiek gramų ir kiek kilogramų vidutiniškai svėrė žvejo pagautos žuvis ir kiek gramų ir kiek kilogramų iš viso svėrė žvejo laimikis.

Užduoties sprendimui bus reikalingi kintamieji:

- trys sveikojo tipo kintamieji žuvų svoriui gramais saugoti;
- sveikojo tipo kintamasis visų žuvų svoriui saugoti.
- realiojo tipo kintamasis vidurkiui saugoti;

Pradiniai duomenys (trijų žuvų svoriai) programai bus nurodomi (įvedami) dialogo būdu.

Programos tekstas/kodas:

```
static void Main(string[] args)
{
    // Programos kintamieji - pradiniai duomenys
    int svoris1,      // gramais
        svoris2,
        svoris3;
    int suma;        // gramais
    double vidurkis; // gramais

    // Pradinių duomenų įvedimas dialogo būdu
    Console.Write("Užrašykite 1-os žuvies svorį (sveikas skaičius, gramais): ");
    svoris1 = int.Parse(Console.ReadLine());

    Console.Write("Užrašykite 2-os žuvies svorį (sveikas skaičius, gramais): ");
    svoris2 = int.Parse(Console.ReadLine());

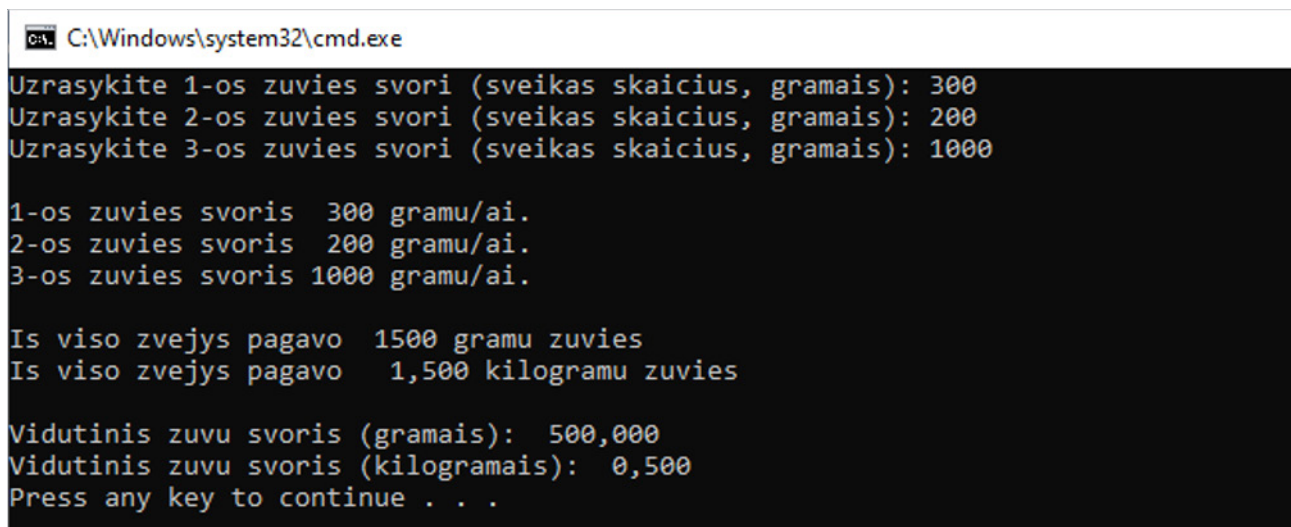
    Console.Write("Užrašykite 3-os žuvies svorį (sveikas skaičius, gramais): ");
    svoris3 = int.Parse(Console.ReadLine());

    // Pradinių duomenų spausdinimas
    Console.WriteLine();
    Console.WriteLine("1-os žuvies svoris {0, 4:d} gramų/ai.", svoris1);
    Console.WriteLine("2-os žuvies svoris {0, 4:d} gramų/ai.", svoris2);
    Console.WriteLine("3-os žuvies svoris {0, 4:d} gramų/ai.", svoris3);
    Console.WriteLine();

    // Skaičiavimai (spausdinimas)
    suma = svoris1 + svoris2 + svoris3;
    Console.WriteLine("Iš viso žvejys pagavo {0, 5:d} gramų žuvies", suma);
    Console.WriteLine("Iš viso žvejys pagavo {0, 7:f3} kilogramų žuvies", suma * 1.0 / 1000);
    Console.WriteLine();

    vidurkis = suma * 1.0 / 3;
    Console.WriteLine("Vidutinis žuvų svoris (gramais): {0, 8:f3}", vidurkis);
    Console.WriteLine("Vidutinis žuvų svoris (kilogramais): {0, 6:f3}", vidurkis / 1000);
}
```

Sukompiliuokite ir įvykdysite programą. Rezultatas ekrane:



```
C:\Windows\system32\cmd.exe
Uzrasykite 1-os zuvies svori (sveikas skaicius, gramais): 300
Uzrasykite 2-os zuvies svori (sveikas skaicius, gramais): 200
Uzrasykite 3-os zuvies svori (sveikas skaicius, gramais): 1000

1-os zuvies svoris 300 gramu/ai.
2-os zuvies svoris 200 gramu/ai.
3-os zuvies svoris 1000 gramu/ai.

Is viso zvejys pagavo 1500 gramu zuvies
Is viso zvejys pagavo 1,500 kilogramu zuvies

Vidutinis zuvu svoris (gramais): 500,000
Vidutinis zuvu svoris (kilogramais): 0,500
Press any key to continue . . .
```