3. C++ DUOMENŲ TIPAI

Šiame skyriuje skaitytojas susipažįsta su C++ kalbos baziniais duomenų tipais, jų modifikacijomis bei kintamųjų charakteristikomis. Skyriuje nagrinėjama kintamųjų aprašų struktūra ir pateikiami įvairūs jų pavyzdžiai. Be to, apžvelgiami įvairūs konstantų tipai ir pateikiamas vardinės konstantos aprašas.

➢ Mokymosi tikslai

Mokydamasis ir baigęs šį skyrių skaitytojas turi:

- žinoti bazinius duomenų tipus ir jų modifikacijas;
- žinoti kintamųjų paskirtį ir jų charakteristikas;
- mokėti aprašyti įvairių tipų kintamuosius;
- mokėti aprašyti skaičiavimus;
- žinoti konstantų tipus;
- mokėti naudotis kintamaisiais ir konstantomis;
- gebėti aprašyti vardinę konstantą.

3.1 Paprastieji duomenų tipai ir jų modifikacijos

C++ programą sudaro duomenys ir komandos, rodančios kokie veiksmai atliekami su duomenimis. Pastaruosius sudaro kintamieji ir konstantos.

Kintamuosius apibūdina šios charakteristikos:
1. atminties adresas;
2. vardas;
3. tipas, rodantis kokius veiksmus galima atlikti su kintamuoju;
4. reikšmė.

C++ kalboje yra trys skaitmeninių duomenų tipų grupės: sveikiesiems skaičiams, sveikiesiems skaičiams be ženklo ir realiesiems skaičiams vaizduoti. Sveikųjų skaičių su ženklais grupė gali būti suprantama kaip paprastųjų sveikųjų skaičių bazinės grupės sinonimas. Šiose grupėse skiriasi tik modifikacijos: **char** ir **unsigned char**. Baziniai duomenų tipai ir jų modifikacijos rodomos 3.1 ir 3.2 lentelėse.

C++ duomenų tipai

3.1 lentelė

Tipas	Aprašymas	Reikšmių atkarpa
char	simbolis	-128 - 127
unsigned char	simbolis be ženklo	0 - 255
signed char	simbolis su ženklu (kaip char)	-128 - 127
int	sveikasis skaičius	-32768 - 32767
unsigned int	sveikasis be ženklo	0 - 65535
signed int	sveikasis su ženklu (kaip int)	-32768 - 32767
short int	trumpasis sveikasis skaičius	-32768 - 32767
unsigned short int	trumpasis sveikasis skaičius be ženklo	0 - 65535
signed short int	trumpasis sveikasis skaičius su ženklu	-32768 - 32767
long int	ilgasis sveikasis skaičius	-2147483648 - 2147483648
signed long int	ilgasis sveikasis su ženklu	-2147483648 - 2147483648
unsigned long int	ilgasis sveikasis skaičius be ženklo	0 - 4294967295
float	slankaus kablelio skaičius	-3.4E+38 - 3.4E+38
double	dvigubo tikslumo slankaus kablelio skaičius	-1.7E+308 - 1.7E+308
long double	ilgasis dvigubo tikslumo slankaus kablelio skaičius	-1.7E+308 - 1.7E+308

Bazinės modifikacijos

3.2 lentelė

Sveikųjų skaičių bazinės modifikacijos	**char, short int, int, long int**.
Realiųjų skaičių bazinės modifikacijos	**float, double, long double.**
Sveikųjų skaičių be ženklo bazinės modifikacijos	**unsigned char, unsigned short int, unsigned int, unsigned long int.**

Sveikieji skaičiai be ženklo taip pat gali būti vartojami loginiams dydžiams, bitų masyvams, ir kitokiems panašaus pobūdžio duomenims saugoti. Dažniausiai naudojami **int**, **char**, **float** kintamųjų tipai. Kintamieji, kurių didelės reikšmės, aprašomi ilgaisiais tipais. Simboliniams kintamiesiems leidžiama priskirti skaitmenines reikšmes. Žinome, kad kiekvienas simbolis ASCII kodų lentelėje turi numerį, atitinkantį šio simbolio eilės numerį lentelėje. Simboliniam kintamajam priskiriant skaičių, priskiriamas atitinkamas simbolis iš ASCII lentelės. Todėl sveikojo tipo kintamajam galima priskirti bet kurį simbolį.

3.2 Kintamųjų aprašai

Kintamojo aprašo sintaksė:

tipas kint.vardas [= pr. reikšmė] [,…, kint. vardas[= pr. reikšmė]];

> *Kintamojo apraše būtina nurodyti vardą ir tipą, o pradinė reikšmė priskiriama kintamajam pagal uždavinio sąlygą. Atminties adresą nustato pats kompiuteris.*

Programos tekste kintamųjų aprašai skiriami vienas nuo kito kabliataškiais. Vieno tipo kintamieji gali būti aprašomi viename apraše. Be to, priskyrimo sakiniais galima nustatyti pradines kintamųjų reikšmes.

1 Pavyzdys

```
int i;
char c;
float x;
int k, j, l;
int i=1, j=2;
```

Kintamųjų vardai

Kiekvienam kintamajam parenkamas unikalus vardas. Pirmas simbolis varde turi būti raidė arba pabraukimo simbolis (_). Už pirmojo simbolio gali būti rašomos raidės, skaičiai ir kiti pabraukimo simboliai. Kintamųjų vardai rašomi mažosiomis raidėmis. Tik klasių ir globaliųjų masyvų vardai pradedami didžiąja raide.

darbas dARBAS DARBAS - tai skirtingų kintamųjų vardai.

Vardas kintamajam duodamas pagal kintamajame saugomos informacijos prasmę. Labai dažnai žodžių santrumpos varduose yra jungiamos pabraukimo simboliu.
Kintamuosius galima aprašyti įvairiose programos vietose:

1. Prieš funkciją *main()*. Toks kintamasis yra globalusis ir gali būti naudojamas visuose programos moduliuose.
2. Funkcijos *main()* viduje. Toks kintamasis - lokalusis funkcijos *main()* kintamasis.

1 Pratimas

```
// programa skaičiuoja PVM nuo pardavimo sumos
#include <iostream.h>
main()
{
float  ps, pvm;                          // dviejų realaus tipo kintamųjų aprašas
const float  mok_k = 0.18;               // vardinė konstanta
        ps = 22.54;                      // priskyrimo sakinys
        pvm = ps*mok_k;                  // skaičiavimai
        cout << ”Mokestis: “  <<  pvm << endl; // rezultatų išvedimas
return 0;
}
```

3.3 Konstantos

Konstantos – tai pastovūs duomenys, kurių reikšmės atliekant programą, nekinta.

Konstantos C++ kalboje gali būti tokių pat tipų, kaip kintamieji. Dažniausiai konstantos nurodomos reikšmėmis, o ne vardais, nors C++ kalboje leidžiama naudotis ir vardinėmis konstantomis.

Vardinės konstantos aprašo sintaksė:

const [tipas] vardas = reikšmė;

Bazinio tipo vardą (char, int, float) apraše galima ir praleisti.

2 Pavyzdys

```
const int y=10;
const float x1=3.5;     const x1=3.5;
```

Sveikojo tipo konstantų reikšmės gali būti papildomos specialiaisiais simboliais, kurie rodo kokioje skaičiavimo sistemoje jos užrašytos:

- aštuntainėje skaičiavimo sistemoje prieš skaičių rašomas nulis 012.
- šešioliktainėje skaičiavimo sistemoje prieš skaičių rašomas nulis ir x 0x10.

Realiojo tipo konstantų reikšmės gali būti užrašomos dešimtaine su fiksuotu tašku ir eksponentine formomis:

eksponentinė forma $aEn=a*10^n$

a - *mantisė (skaičiaus reikšminiai skaitmenys)*

n - *eilė (per kiek pozicijų reikia perkelti tašką mantisėje)*

jei n teigiamas, kablelis keliamas į dešinę.

jei n neigiamas, kablelis keliamas į kairę.

Pervedant į dešimtainę skaičiavimo sistemą, dešimtainis taškas perkeliamas už pirmo reikšminio skaičiaus ir pataisoma skaičiaus eilė.

3 Pavyzdys

```
254.6   pertvarkomas        2.546E+02
-0.01   pertvarkomas        -1.0E-02
```

Eilutinė konstanta - tai simbolių grupė, rašoma tarp kabučių (Pvz. “kalba”).

Kompiuterio atmintyje visos eilutinės konstantos baigiamos simboliu *null*. Pabaigos simbolis nėra matomas, tačiau C++ kalbos kompiliatorius tikrina šį požymį ir pagal jį nustato konstantos tipą. Eilutės pabaigos simbolis *null* ir skaičius nulis nėra tas pats simbolis. Eilutės pabaigos simbolis *null* - *tai* pirmasis *ASCII* lentelės simbolis. Skaičiaus 0 numeris ASCII lentelėje yra 48.

Eilutinio tipo kintamųjų C++ kalboje nėra. Tam, kad galima būtų įvesti simbolių eilutę, naudojami simbolių masyvai.

Simbolinė konstanta - tai vienas bet koks simbolis, parašytas tarp apostrofų.

4 Pavyzdys

‘w’ ‘*’ ‘=’ ‘7’ ‘&’

“R” ir ‘R’ atmintyje saugomos skirtingai.

C++ kalboje naudojami specialūs ryšio kanalų ir duomenų atvaizdavimo įrenginių valdymo simboliai. Šių simbolių vardai pradedami simboliu \. Dažniausiai naudojami simboliai pateikiami 3.3 lentelėje.

Valdymo simboliai

3.3 lentelė

Simbolis	Pavadinimas
\a	skambutis
\b	trynimo simbolis
\f	perėjimas į naują puslapį
\n	nauja eilutė
\t	horizontalioji tabuliacija
\v	vertikalioji tabuliacija
\\	kairysis pasvirusis brūkšnys
\'	kabutė
\000	skaičius aštuntainėje sistemoje
\xhh	skaičius šešioliktainėje sistemoje
\0	nulis

2 Pratimas

```
#include <iostream.h>
main()
{
        char  bell = ’\a’;
        cout  << bell; // skambutis
        cout << “Petras”<<’\n’<<”Jonas”; // spausdinamos dvi eilutės
return 0;
}
```

Kontroliniai klausimai

1. Suklasifikuokite jums žinomus C++ kalbos bazinius duomenų tipus.
2. Kokiais simboliais pradedami programos komentarai?
3. Paaiškinkite skirtumą tarp globaliųjų ir lokaliųjų kintamųjų.
4. Kokias žinote kintamojo charakteristikas?
5. Kokias kintamojo charakteristikas būtina nurodyti aprašant kintamąjį?
6. Kurie kintamųjų vardai yra teisingi?
   mano_vardas1 89_pardavimas pardavimas_89
   d_u
7. Kokia instrukcija pirminiam procesoriui galima įterpti bibliotekų sąsajų failus į programos tekstą?
8. Kas vykdoma pirmiausiai - kompiliavimas ar instrukcija?
9. Kokio tipo yra šios konstantos?
   0 -12.0 “2.0” “X” ‘X’ 65.4 -708 ‘0’
10. Kiek kintamųjų ir kokio tipo aprašoma šiuose sakiniuose?

```
int   i, j, k;
char  c, d;
float x=65.34;
```

Užduotys

1. Parašyti programą, kurioje jūsų vardo ir pavardės pirmosios raidės bei telefono numeris būtų įsimenami trijuose kintamuosiuose.
2. Parašyti programą, kurioje būtų aprašomi visų jums žinomų tipų kintamieji. Jų reikšmes įvesti klaviatūra. Įvestus duomenis rodyti kompiuterio ekrane.
3. Parašyti programą, kuri aprašo sveikojo tipo kintamąjį. Po to pradinę reikšmę padidina 10000 kartų, iki programa bus baigta vykdyti. Sekite kompiuterio darbą.
4. Parašyti programą, kurioje aprašomas unsigned integer tipo kintamasis. Po to pradinė reikšmė padidinama 10000 kartų iki programa bus baigta vykdyti. Rezultatus palyginkite su 3 užduoties rezultatais.

Santrauka

Šiame skyriuje susipažinote su C++ kalbos baziniais duomenų tipais, jų modifikacijomis bei kintamųjų charakteristikomis. Be to, išmokote aprašyti įvairių tipų kintamuosius ir konstantas bei jais pasinaudoti.

Informacijos šaltiniai

4. SKAIČIAVIMŲ APRAŠYMAS

Šiame skyriuje skaitytojas susipažįsta su skaičiavimų aprašymu C++ programavimo kalboje, aritmetinėse išraiškose naudojamomis operacijomis ir jų atlikimo tvarka. Skyriuje pateikiamos priskyrimo sakinio, aprašančio skaičiavimus, modifikacijos bei jų panaudojimo taisyklės. Be to, analizuojamos kintamojo reikšmės didinimo ir mažinimo vienetu operacijos bei duomenų tipų redukavimo skaičiavimuose tvarka.

➢ Mokymosi tikslai

Mokydamasis ir baigęs šį skyrių skaitytojas turi:

- žinoti C++ kalboje naudojamus aritmetinius veiksmus ir jų atlikimo tvarką;
- mokėti aprašyti skaičiavimus priskyrimo sakiniais;
- mokėti naudotis priskyrimo sakiniais ir jų modifikacijomis;
- žinoti priešdėlinius ir priesaginius aritmetinius veiksmus;
- mokėti naudotis skirtingų tipų duomenimis skaičiavimuose.

1. Aritmetiniai veiksmai ir jų atlikimo tvarka

Skaičiavimai aprašomi išraiškomis, kurias sudaro operandai ir su jais atliekami veiksmai, nurodomi operatoriais. Aritmetinėse išraiškose veiksmai nurodomi simboliais, kurie pateikiami 4.1 lentelėje.

4.1 lentelė

Prioritetas	Ženklas	Reikšmė
1	-	minuso ženklas
2	*	daugyba
2	/	sveikųjų skaičių dalyba (liekana atmetama)
2	%	sveikųjų skaičių dalyba (liekanos radimas)
3	+	sudėtis
3	-	atimtis

Kintamajam galima priskirti teigiamąją arba neigiamąją reikšmę.

1 Pavyzdys

```
b=25+a;
c=-a;
naujas = senas - -kitas;
cout << 10 / 3  ;          // rezultatas 3
cout << 300 / 165 ;        // rezultatas 1
cout << 10  %  3 ;         // rezultatas 1
cout << 300 % 165;         // rezultatas 135
```

Skliausteliais galima pakeisti veiksmų atlikimo tvarką.

2 Pavyzdys

```
2 + 3 * 2          rezultatas   8
(2+3) * 2          rezultatas   10
```

4.2 Priskyrimo sakinys ir jo modifikacijos

C++ kalboje naudojamos lanksčios skaičiavimų aprašymo taisyklės. Skaičiavimai aprašomi priskyrimo sakiniais ir įvairiomis jų modifikacijomis.

Priskyrimo sakinio sintaksė:

kintamojo vardas = išraiška;

Kairėje lygybės ženklo pusėje rašomas kintamojo vardas, kurio reikšmė skaičiuojama. Dešinėje pusėje rašoma išraiška, aprašanti skaičiavimo tvarką. Aritmetinės išraiškos rodo, kokias operacijas, kokia tvarka ir su kokiais operandais reikia atlikti.
Kintamajam, esančiam kairėje pusėje, priskiriama išraiškos, esančios dešinėje, reikšmė.

3 Pavyzdys

```
a = 100;
a = b = c = 0;          (a,b,c bus priskirtas 0)
y = sin(x);
val = 5 + ( r = 9 - c );
```

C++ automatiškai nepriskiria kintamiesiems reikšmės 0. Tai reikia atlikti atskiru priskyrimo sakiniu. Priskyrimo sakinį galima įterpti bet kurioje programos vietoje, net išraiškos viduje. Pavyzdyje skaičiuojamas reiškinys *9 - c* ir gauta reikšmė priskiriama kintamajam *r*. Vėliau skaičiuojama kintamojo *val* reikšmė. Deja, tokie kombinuoti priskyrimo sakiniai yra labai sunkiai skaitomi.

Dažnai tenka keisti kintamųjų reikšmes. Pavyzdžiui, su jau esančia rezultato kintamojo reikšme reikia atlikti aritmetinį veiksmą ir gautą naują reikšmę priskirti tam pačiam kintamajam.

C++ kalboje galima naudotis tokiomis priskyrimo sakinio modifikacijomis (žr. 4.2 lentelė).

Priskyrimo sakinio modifikacijos

4.2 lentelė

Veiksmas	Pavyzdys	Ekvivalentas
+=	b+=500;	b=b+500;
-=	b-=50;	b=b-50;
*=	s*=1.2;	s=s*1.2;
/=	f/=.50;	f=f/.50;
%=	d%=7;	d=d%7;

Sudėtiniai veiksmai aritmetinėse išraiškose atliekami paskutiniai.

1 Pratimas

```
// priskyrimo sakinio modifikacijos
#include <iostream.h>
main()
{
        int ats = 2;
        ats *= 5 + 3;
        cout << ats <<endl;     // rezultatas 16
        ats +=4 - 2;
        cout << ats <<endl;     // rezultatas 18
return 0;
}
```

4.3 Priešdėliniai ir priesaginiai aritmetiniai veiksmai

Priešdėliniai ir priesaginiai aritmetiniai veiksmai kintamųjų reikšmes didina ir mažina vienetu.

++ *kintamojo reikšmės didinimo veiksmas.*

- - *kintamojo reikšmės mažinimo veiksmas.*

++ ir - - veiksmai (žr. 4.3 lentelė) gali būti rašomi *prieš* kintamojo vardą ir *už* jo. Pirmuoju atveju veiksmas vadinamas *priešdėliu* , antruoju - *priesaga*.

Kintamojo reikšmės didinimo ir mažinimo vienetu veiksmai bus atliekami skirtingai, priklausomai nuo užrašo - kaip priešdėlis, ar kaip priesaga.

Priešdėlis rodo, kad kintamojo reikšmė didinama arba mažinama vienetu prieš kintamojo panaudojimą.

Priesaga rodo, kad kintamojo reikšmė didinama arba mažinama vienetu po kintamojo panaudojimo.

++ ir - - veiksmai

4.3 lentelė

Veiksmas	Aprašymas
i++	Priskyrimas ir didinimas vienetu
++i	Didinimas vienetu ir priskyrimas
i- -	Priskyrimas ir mažinimas vienetu
- - i	Mažinimas vienetu ir priskyrimas

Labai dažnai veiksmai ++ ir - - naudojami kuriant ciklus, didinant arba mažinant ciklo kintamojo reikšmes.

++ ir - - veiksmai atliekami tik su sveikojo tipo kintamaisiais.

4 Pavyzdys		
	a=6; b=++a-1;	a=6; b=a++-1;
Rezultatas	b=6, a=7	b=5, a=7

Patariama prisiminti ++ ir - - veiksmų naudojimo taisykles:

1. *Skliausteliais negalima pakeisti veiksmų vykdymo sekos.*
2. *Šiais veiksmais negalima naudotis išraiškose.*
3. *++ ir - - veiksmais galima naudotis su loginiais kintamaisiais.*

4.4 Skirtingi duomenų tipai skaičiavimuose

Jeigu išraiškose naudojami skirtingi duomenų tipai, tai mažesnės reikšmių atkarpos tipas pervedamas į didesnės reikšmių atkarpos tipą. Pavyzdžiui, sudedant slankiojo kablelio skaičių su sveikuoju skaičiumi, sveikasis skaičius pervedamas į slankiojo kablelio formatą ir po to atliekama sudėtis.

Skaičiavimų aprašymuose leidžiama laikinai keisti duomenų tipus (redukuoti). Redukuojant kintamojo tipas laikinai keičiamas nauju, nurodytuoju lenktiniuose skliausteliuose.

Duomenų tipo redukavimo sintaksė:

(tipas) išraiška

tipas - bet kuris leidžiamas duomenų tipas.

išraiška - išraiška, kintamasis arba konstanta.

Atlikus skaičiavimus, tipas atstatomas.

5 Pavyzdys

```
int     metai;
double f;
metai_f = (double)metai* f;
```

Pavyzdyje *int* tipo kintamasis *metai* laikinai redukuojamas į dvigubo tikslumo su slankiuoju kableliu formatą.

🕮 Klausimai

1. Perrašykite šias formules *C++* programavimo kalba.
   a) x=(a-b) · (a-c)/2
   b) $f = \frac{a1/2}{b1/3}$
   c) $d = \frac{(8 - x2)}{(x - 9)} - \frac{(4 * 2 - 1)}{x3}$
2. Koks yra šių reiškinių rezultatas?
   a) 9%2+1
   b) 1+(10-(3%2+5))

✎ Užduotys

1.	Parašykite programą reiškinio *103/4* rezultato liekanai rodyti ekrane.
2.	Parašykite programą, kuri parodytų ekrane apskritimo, kurio spindulio r reikšmė įvedama klaviatūra, plotą. Apskritimo plotas skaičiuojamas pagal formulę *pi*r*2* (pi=3,14159) .
3.	Apskaičiuokite z kintamojo reikšmę pagal formulę $\frac{x \cdot y - 2 \cdot \sqrt{\lvert x \cdot y \rvert}}{\sqrt{\lvert x - y \rvert}}$. Pradiniai duomenys: x = -5, y = -2.

⌛ Santrauka

Šiame skyriuje susipažinote su skaičiavimų aprašymu C++ programavimo kalboje, aritmetinėmis išraiškomis, naudojamomis operacijomis ir jų atlikimo tvarka. Taip pat išmokote naudotis priskyrimo sakiniais, aprašančiais skaičiavimus, bei jų modifikacijomis ir panaudojimo taisyklėmis. Be to, susipažinote su kintamojo reikšmės didinimo ir mažinimo vienetu veiksmais bei duomenų tipų redukavimo skaičiavimuose tvarka.

ⓘ Informacijos šaltiniai