

Konspektas



C++ programavimas

5 pamoka

Turinys

Kintamieji ir jų reikšmės.....	1
Sveikieji ir realieji skaičiai.....	1
Realiųjų skaičių vaizdavimas	2
Aritmetinės operacijos.....	2
Kintamieji	3
Kintamųjų vardai ir reikšmės.	3
Reiškiniai	4
Žodinėlis.....	4
Pagrindiniai duomenų tipai.....	5

Kintamieji ir jų reikšmės.

Algoritmuose užrašomi veiksmai su duomenimis. Yra daug įvairiausių duomenų. Bendriausia prasme duomenys skirstomi į pastovius (konstantas) ir kintamus (kintamuosius). Kintamieji žymimi vardais, atliekant programą jie įgyja reikšmes. Konstantos ir kintamieji, sujungti operacijų (veiksmų) ženklais, sudaro reiškinius. Reiškinių reikšmių skaičiavimas - pagrindinė programos veiksmų dalis.

Duomenų sąvoka jums jau pažįstama. Ne kartą kalbėjome apie duomenis kompiuteryje bei algoritmuose. Žinome, kad algoritmui pateikiami pradiniai duomenys, jis atlieka su jais veiksmus ir gauna rezultatus - galutinius duomenis.

Duomenys gali būti sveikieji ir realieji skaičiai, simboliai, tekstas ir pan. Duomenų rūšys programavime vadinamos **duomenų tipais**.

Sveikieji ir realieji skaičiai

Mums įprasčiausios duomenų rūšys - sveikieji ir realieji skaičiai. Iš matematikos kurso prisimename natūraliuosius, sveikuosius, racionaliuosius ir realiuosius skaičius (15a pav.). Mažiausia yra natūraliųjų skaičių aibė, didžiausia - realiųjų. Taigi kiekvieną natūralųjį skaičių galima laikyti sveikuoju, kiekvieną sveikąjį - racionaliuoju, kiekvieną racionalųjį - realiuoju skaičiumi.

Matematikoje vartojami baigtiniai ir begaliniai skaičiai. Pavyzdžiui, dalybos $10 : 3$ rezultatas yra skaičius $3,333\dots$ su begaliniu trejetų skaičiumi po kablelio, bet skaičius $3,000\dots$ yra baigtinis.

Kompiuteryje visi skaičiai turi būti baigtiniai. Kiekvienas skaičius koduojamas tam tikru bitų skaičiumi, paprastai visiems vienodu. Kompiuteryje galima tiksliai koduoti tik diskrečiuosius dydžius. Tokie yra natūralieji ir sveikieji skaičiai. Racionalieji ir realieji skaičiai kompiuteryje išreiškiami apytiksliai, su tam

Parengė ITMM Artūras Šakalys

tikru skaitmenų po kablelio skaičiumi. Be to, kiekvienas skaičius priskiriamas kuriam nors duomenų tipui: sveikųjų arba realiųjų (15b pav.). Pavyzdžiui, programavime sveikasis skaičius 5 nepriklauso realiųjų skaičių tipui, o jam matematiškai lygus realusis skaičius 5,0 nepriklauso sveikųjų skaičių tipui. Taigi skaičiai 5 ir 5,0 programavime yra skirtingų tipų.

Realiųjų skaičių vaizdavimas

Sudarant programas, realiuosius skaičius galima užrašyti dviem būdais:

1) skaičiaus sveikąją dalį nuo trupmeninės atskiriant tašku, pavyzdžiui:

2.0 125.15 -235.0 -777.0 3.1415926536;

2) vartojant laipsnio rodiklį, t.y. užrašant standartine išraiška.

Antrasis būdas jums jau žinomas iš matematikos ir jį paprastai vartoja kompiuteris, spausdindamas rezultatus. Šiame užrašė vartojamas daugiklis 10^n . Paskalio kalboje vietoj laipsnio pagrindo 10 rašoma raidė E (arba e) ir po jos - sveikuoju skaičiumi išreikštas laipsnio rodiklis n. Pavyzdžiui:

Programavime	5E1	125E12	-2372E-1	-6.66E2
Matematikoje	$5 \cdot 10^1$	$125 \cdot 10^{12}$	$-2372 \cdot 10^{-1}$	$-6,66 \cdot 10^2$

Aritmetinės operacijos

C++ kalboje su skaičiais atliekami įprastiniai aritmetiniai veiksmai (operacijos): sudėtis, atimtis, daugyba ir dalyba. Sudėtis ir atimtis žymima kaip ir matematikoje: + ir -. Daugyba programose žymima žvaigždute (*).

C++ kalboje dalyba yra trijų rūšių: viena atliekama su realiaisiais skaičiais ir dvi - su sveikaisiais skaičiais. Realiųjų skaičių dalyba atliekama taip kaip ir matematikoje, ir žymima įstrižu brūkšniu (/), pavyzdžiui, 5/2.

Dalijant sveikuosius skaičius, galima gauti du rezultatus: dalmenį ir liekaną (kai pageidaujama, kad rezultatai būtų sveikąjo tipo). Pavyzdžiui, 14 padalijus iš 5, gaunamas dalmuo 2 ir liekana 4. Sprendžiant uždavinius, kai kada reikia tik dalmens, o kai kada - tik liekanos. Šios dalybos operacijos žymimos ženklų / (divide - "dalyti") ir % (module - "modulis, liekana").

Sveikųjų skaičių dalybos pavyzdžiai:

$$\begin{array}{lll} 13 / 5 = 2 & 3 / 8 = 0 & 0 / 7 = 0 \\ 13 \% 5 = 3 & 3 \% 8 = 3 & 0 \% 7 = 0 \end{array}$$

Primename, kad, kaip ir matematikoje, iš nulio dalyti negalima.

Dalijant neigiamus skaičius, operacija / atliekama taip: skaičiai dalijami be ženklų (kaip teigiami), gauto rezultato - dalmens - ženklas nustatomas kaip matematikoje (kai dalinys ir daliklis vienodų ženklų - pliusas, kai skirtingų - minusas). Neigiamų skaičių liekaną įsivaizduoti sunkiau, todėl operacijos % su neigiamais skaičiais ir nevertosime.

Kintamieji

Žinome, kad pastovūs duomenys vadinami konstantomis. Paprasčiausiu atveju konstanta - tai skaičius. Vadinasi, konstantos - tai tokie duomenys, kurie išlieka pastovūs visą laiką, kol atliekama programa.

Duomenys, kurių reikšmės gali keistis atliekant programą, vadinami kintamais duomenimis, arba tiesiog **kintamaisiais**.

Kintamieji žymimi vardais. Vardai C++ kalboje sudaromi iš raidžių ir skaitmenų, bet būtinai turi prasidėti raide. Pavyzdžiui:

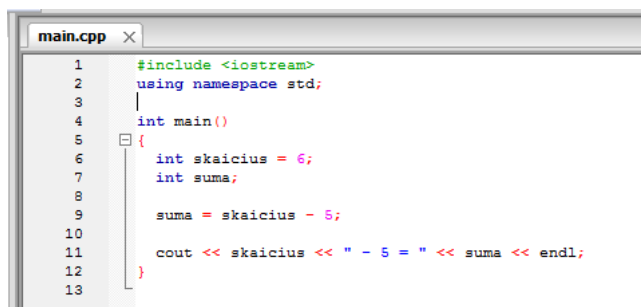
a, suma, vidurkis, plotas, kaina, sprend1, sprend2.

Matematikoje įprasta kintamuosius dydžius žymėti viena raide. Čia vartojamos įvairių abėcėlių raidės (pvz., graikiškos). Dirbant su kompiuteriu tenka prisitaikyti prie jo ypatumų ir vartoti kompiuterio klaviatūroje esančias raides (lotyniškas, lietuviškas).

Parinkant kintamųjų vardus reikia stengtis, kad jie nusakytų jais pažymėtų duomenų paskirtį. Pavyzdžiui, skaičiaus kvadratą patogų žymėti santrumpomis **kv**, **kvadr** ar net visu žodžiu **kvadratas**, trikampio kraštinės - matematikoje vartojamais įprastais vardais **a**, **b**, **c**, skaičių sumą - **sum**, **suma**, **s** ar panašiai.

Kompiuteriui visiškai nesvarbu, kokiais vardais žymimi kintamieji. Svarbu tik tai, kad visi viename algoritme vartojami kintamieji turėtų skirtingus vardus - kad būtų galima atskirti juos vieną nuo kito. Tuo tarpu žmogui, skaitančiam algoritmą ir norinčiam jį suvokti, kur kas geriau, kai kintamųjų vardai parinkti taip, kad nusako jais pažymėtų duomenų paskirtį.

Jau žinome, kad kiekvienam kintamajam kompiuteris skiria vietą savo atmintinėje. Ten saugoma to **kintamojo reikšmė** - konkretus duomuo.



```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int skaicius = 6;
7     int suma;
8
9     suma = skaicius - 5;
10
11     cout << skaicius << " - 5 = " << suma << endl;
12 }
13
```

Programos fragmentas

Kintamųjų vardai ir reikšmės.

Taigi kintamasis apibūdinamas vardu. Kintamojo vardas žymi atmintinės laukelį, o pačiame laukelyje įrašoma kintamojo reikšmė. Į tą patį atmintinės laukelį gali būti įrašomos vis kitos to paties kintamojo reikšmės.

Susitarta, kad užrašant kintamojo vardą turi būti nurodoma, kokias reikšmes gali įgyti šis kintamasis. Tai apibūdinama užrašant duomenų tipą.

Sveikųjų skaičių tipas nurodomas žodžiu **int** (sveikasis), realiųjų - žodžiu **float** (realusis). Kintamųjų vardai ir tipai nurodomi aprašuose, tuoj pat po programos antraštės. Kiekvienas kintamasis C++ kalboje turi būti aprašytas, t.y. būtina nurodyti jo tipą.

Kintamųjų aprašų pavyzdys:

```
int main()
```

```
{  
  
    int skaičius = 6;  
  
    int suma;
```

Aritmetinės operacijos su sveikaisiais skaičiais atliekamos kaip matematikoje. Jeigu argumentai yra sveikieji, tai ir rezultatas bus sveikojo tipo duomuo, o jei nors vienas bet kurios operacijos argumentas yra realusis skaičius, tai ir rezultatas bus realiojo tipo.

Atliekant veiksmus su sveikaisiais skaičiais galima gauti didelį skaičių, netelpantį jam skirtoje atmintinės vietoje. Tokia situacija vadinama **perpildymu**. Perpildymas gali įvykti greitai - leistinas sveikųjų skaičių intervalas nėra didelis. Sveikojo tipo (**int**) reikšmės C++ kalboje gali būti tik iš intervalo $[-2^{31} \quad 2^{31}]$.

Reiškiniai

Programavime ir matematikoje reiškinių sąvoka suprantama vienodai. **Reiškinys** - tai konstantos ir kintamieji, sujungti operacijų (veiksmų) ženklais bei lenktiniais skliaustais.

Reiškinių C++ kalba pavyzdžiai:

```
(p1 + p2 + p3) / 3  
sk1 + sk2 - sk3 % 10  
(a + b) * (a - b)
```

Veiksmų atlikimo tvarka reiškiniuose įprasta: pirmiausia atliekami veiksmai skliaustuose, po to - dauginama ir dalijama, o galiausiai - sudedama ir atimama. Vienodo prioriteto veiksmai (sudėtis ir atimtis arba daugyba ir dalyba) atliekami paėiliui iš kairės į dešinę. Pavyzdžiui, reiškiny

$$10 / 6 \% 3 + 4 * 3 \% 5$$

skaičiuojamas tarsi būtų šitaip suskliaustas:

$$((10 / 6) \% 3) + ((4 * 3) \% 5)$$

Šio reiškinių reikšmė lygi 3.

Reiškinių rezultato tipas priklauso nuo jų sudarančių kintamųjų ir operacijų tipų. Jei visi argumentai yra sveikojo tipo, tai sudėtis, atimtis, daugybos ir dalybų / bei % rezultatas yra sveikojo tipo. Dalybos operacijos, kai ji žymima /, rezultatas visada yra realiojo tipo.

Žodinėlis

duomenų tipas	int
sveikasis skaičius	float
realusis skaičius	* /
kintamasis	+ -
kintamojo vardas	()
kintamojo reikšmė	%
reiškiny	
perpildymas	

Pagrindiniai duomenų tipai

Delphi	Reikšmė/dydis	C++
ShortInt	8-bit integer	signed char
SmallInt	16-bit integer	short
LongInt	32-bit integer	int
Byte	8-bit unsigned integer	unsigned char
Word	16-bit unsigned integer	unsigned short
Integer	32-bit integer	int
Cardinal	32-bit unsigned integer	unsigned int
Boolean	true/false	bool
ByteBool	true/false or 8-bit unsigned integer	unsigned char
WordBool	true/false or 16-bit unsigned integer	unsigned short
AnsiChar	8-bit unsigned character	char
Char	8-bit unsigned character	char
Single	32-bit floating point number	float
Double	64-bit floating point number	double
Extended	80-bit floating point number	long double
Real	32-bit floating point number	double