

KONSPEKTAS C++ KALBOS

Parengė Kauno „Varpo“ gimnazijos 2 e kl. mokinys(nė)

VARDENIS PAVARDENIS

Turinys

Algoritmas, jo vykdymas, savybės.....	2
Pavyzdinis programos kūrimo planas.....	4
Kintamasis, kintamojo reikšmė	4
Aprašant kintamuosius, reikia prisiminti:.....	5
Priskyrimo sakinyss	5
Duomenų įvedimas klaviatūra.....	6
Rezultatų (duomenų) išvedimas į ekraną.....	6
Sąlyginis sakinyss if	6
Rekomenduojamų matematinių funkcijų sąrašas	7

Algoritmas, jo vykdymas, savybės

Algoritmū vadinami aiškūs vienareikšmiai nurodymai (sakiniai), kaip turint tam tikrus pradinius duomenis galima gauti reikiamus rezultatus.

Pradiniai duomenys – tai iš anksto žinomos reikšmės (paprasčiausiu atveju – skaičiai), būtinos veiksmams atlikti.

Rezultatai – tai reikšmės, gautos atlikus visus skaičiavimus.

Tarpiniai rezultatai – tai apskaičiuotos reikšmės, kurios naudojamos tolesniems veiksmams atlikti.

Diskretumas. Algoritmas suskaidomas į baigtinę žingsnių seką.

Aiškumas. Visus algoritmu aprašomus veiksmus bet kuris vykdytojas turi suprasti vienareikšmiškai.

Rezultatyvumas. Atlikus baigtinį skaičių algoritmo veiksmų, gaunamas rezultatas.

Baigtumas. Rezultatas gaunamas įvykdžius baigtinį skaičių algoritmo veiksmų.

Universalumas. Naudojant tą patį algoritmą, sprendžiami visi to tipo uždaviniai.

Algoritmai gali būti pateikiami skirtingais būdais:

Užrašomi *žodžiais*, vaizduojami *grąfiškai* – dažniausiai *simbòlinėmis (blòkinėmis) schémomis* arba *struktūrográmomis*.

Programávimas – tai procesas, kuris apima šiuos etapus:

- ✓ užduoties analizès;
- ✓ užduoties skaidymas į dalis;
- ✓ sprendimo metodų parinkimas ir sukūrimas;
- ✓ kintamųjų parinkimas;
- ✓ algoritmų sudarymas;
- ✓ programos teksto rašymas, derinimas ir testavimas.

Programáavimo aplinkà – tai aparatinès ir programinès įrangos priemonių visuma.

Kompiliátorius – tai programa, kuri verčia parašytą programos tekstą į kompiuteriui suprantamą kalbą.

Prográmos struktūrą (sandarą) lemia programavimo kalba.

```
#include <iostream> //biblioteka
using namespace std; //biblioteka
int main() //pagrindinė funkcija
{ //programos pradžia
    cout << "Vardenis Pavardenis" << endl; //pranešimas į ekraną
    return 0; //grąžiną reikšmę
} //programos pabaiga
```

Rašant programą, reikėtų laikytis tokių programáavimo kultūros taisyklių:

- ✓ algoritmas turi geriausiai tikti uždaviniui spręsti, būti aiškus, trumpas ir logiškai pagrįstas;
- ✓ kintamųjų, konstantų, tipų, funkcijų vardai turi atitikti aprašomų objektų prasmę, tačiau neturi būti ilgi;
- ✓ programa turi būti rašoma su komentarais.

Esminės programos teksto dėstymo taisyklės:

- ✓ programos dalims išskirti paliekamos tuščios eilutės arba rašomi komentarai, sudaryti tik iš minuso ar kitokių ženklų;
- ✓ pavaldumui išryškinti daromos įtraukos (pradedama rašyti toliau nuo krašto) ciklo ir sąlyginiuose sakiniuose;
- ✓ skaitomumui pagerinti sąlyginiuose ir sudėtiniuose sakiniuose vertikaliai lygiuojami šie žodžiai: **if**,
else ir **{, }**.

Programos tekstas rašomas iš abiejų pusių tarpais atskiriant ženklus.

Pavyzdinis programos kūrimo planas

Pradinių duomenų ir būsimų rezultatų analizė.

Uždavinio sprendimo idėja

Kintamųjų parinkimas pradiniam duomenims ir rezultatams laikyti.

Algoritmo sukūrimas.

Programos rašymas.

Testavimas.

Kintamasis, kintamojo reikšmė

Duomenys, kurie atliekant programą nekinta, vadinami *konstantomis*.

Kintamieji skirti duomenims, kurių reikšmės keičiasi atliekant programą, atmintyje laikyti.

Jų *vardai* sudaromi iš raidžių ir skaitmenų, tačiau pirmas ženklas turi būti raidė.

Kintamieji atmintyje gali laikyti tik nurodyto tipo duomenų *reikšmes*.

Kintamieji reikšmes (duomenis) gauna duomenų *įvedimo* (skaitymo) *sakiniais*.

Priskyrimo sakiniai, kai kintamiesiems reikšmės suteikiamos programos tekste.

Sudėtinis sakiny – tai tarp riestinių skliaustų { } parašyta sakinių seka.

Aprašant kintamuosius, reikia prisiminti:

- ✓ Kintamieji, aprašyti prieš pagrindinę funkciją `main()`, vadinami **globaliaisiais**. Jie galioja visoje programoje.
- ✓ Kintamieji, aprašyti funkcijoje, vadinami **lokaliaisiais**. Jie galioja tik toje funkcijoje, išėjus už funkcijos ribų, šių kintamųjų reikšmės neišsaugomos.
- ✓ Jei kintamasis tokiu pačiu vardu aprašytas prieš pagrindinę funkciją `main()` ir funkcijoje, pirmenybė suteikiama lokaliajam, t. y. funkcijoje globalusis kintamasis negalioja.

Priskyrimo sakiny

Priskyrimo sakiny naudojamas, kai kintamajam reikia suteikti reikšmę programos tekste.

Skiriami trys reiškinų tipai:

Aritmètinis. Tai reiškinys, kuriame kintamieji ir konstantos jungiamos aritmetinių operacijų ženklais.

Sàntykio. Tai du aritmetiniai reiškiniai, tarp kurių rašomas santykio operacijos ženklas.

Lòginis. Tai reiškinys, kuriame naudojami loginių operacijų ženklai.

Įvedimo ir išvedimo srauto samprata

Įvedimo / išvedimo srautas – tai loginis įrenginys, kuris išveda ir priima naudotojo informaciją.

Klasė – tai darinys, kuriame saugomi kintamieji ir **metodai** – funkcijos, taikomos darbui su klasės kintamųjų reikšmėmis.

Duomenų įvedimas klaviatūra

Duomenų įvedimo procesas vykdomas dviem etapais: pirmiausia duomenys įvedami klaviatūra, po to jie priskiriami kintamiesiems.

Duomenims įvesti iš standartinio įvedimo srauto `cin`, paprastai susieto su klaviatūra, ir / arba iš srauto, susieto su failu, dažniausiai naudojamas operatorius `>>`. Asmeniniuose kompiuteriuose standartinis įvedimo įtaisas yra klaviatūra.

Rezultatų (duomenų) išvedimas į ekraną

Duomenims išvesti į standartinį išvedimo srautą `cout`, paprastai susietą su ekranu, ir / arba į srautą, susietą su failu, dažniausiai naudojamas operatorius `<<`.

Išvedamų duomenų sąrašuose galima nurodyti kintamuosius, konstantas, reiškinius ir tekstinius duomenis.

Sąlyginis sakinyis if

Sąlyginių sakiniių programoje keičiama nuosekli sakinių atlikimo tvarka: jei *Sąlyga* tenkinama, atliekamas *PirmasSakinys*, jei ne – po `else` esantis *AntrasSakinys*. Jeigu reikia atlikti kelis sakinius, kai *Sąlyga* tenkinama arba netenkinama, tai jie rašomi tarp `{` ir `}`. *Sąlyga* – bet koks santykio arba loginis reiškinys.

Sąlyginio sakinio šakose galima užrašyti bet kokius C++ kalbos sakinius. Bet kurioje sąlyginio sakinio šakoje galima užrašyti dar vieną sąlyginį sakinį, pastarojo šakose – dar po vieną ir t. t. Toks sakinytis vadinamas **sudėtingu sąlyginių sakiniių**.

Rekomenduojamų matematinų funkcijų sąrašas

Funkcija	Paskirtis
<pre>int abs(int x);</pre>	<p>Grąžina sveiką skaičių x absoliutųjį didumą. Esant teigiamam ar neigiamam funkcijos parametrai, rezultatas yra teigiamas. Pavyzdys:</p> <pre>int x = -10; cout << "Sveiką skaičių " x << " modulis yra " << abs(x) << endl;</pre> <p>Ekrane matysite: Sveiką skaičių -10 modulis yra 10</p>
<pre>double fabs(double x);</pre>	<p>Grąžina realiojo skaičiaus x absoliutųjį didumą.</p> <p>Pavyzdys:</p> <pre>double x = -15.5; cout << "Realiojo skaičiaus " x << " modulis yra " << fabs(x) << endl;</pre> <p>Ekrane matysite: Realiojo skaičiaus -15.5 modulis yra 15.5</p>
<pre>double pow(double x, double y);</pre>	<p>Skaičių x kelia laipsniu y. (Jei y yra 0.5, tai funkcija skaičiuoja neneigiamo argumento x kvadratinę šaknį.) Pavyzdys:</p> <pre>int x = 15, y = 3; cout << "Skaičius " << x << " pakeltas laipsniu " << y << ": " << pow(x, y) << endl;</pre> <p>Ekrane matysite: Skaičius 15 pakeltas laipsniu 3: 3325</p>
<pre>double sqrt(double x);</pre>	<p>Skaičiuoja neneigiamo argumento x kvadratinę šaknį.</p> <p>Pavyzdys:</p> <pre>double x = 15.5; cout << "Kvadratinė šaknis iš realiojo skaičiaus " << x << " yra " << sqrt(x) << endl;</pre> <p>Ekrane matysite: Kvadratinė šaknis iš realiojo skaičiaus 15.5 yra 3.937</p>
<pre>int random(int Riba);</pre>	<p>Grąžina atsitiktinį sveiką skaičių iš intervalo $[0; Riba)$.</p> <p>Pavyzdys:</p> <pre>cout << "Atsitiktinis skaičius intervale 0-99: "</pre> <p>Ekrane matysite: << random(100) << endl;</p>

	<p>Ekrane matysite: Atsitiktinis skaičius intervale 0-99: 84</p>
<pre>int rand(void);</pre>	<p>Grąžina atsitiktinį sveikąjį skaičių iš intervalo [0; RAND_MAX]. Konstanta RAND_MAX dažniausiai lygi 32767. Norint pasinaudoti konstanta RAND_MAX, reikia prie programos prijungti failą <code>stdlib.h</code> sakiniu <code>#include <stdlib.h></code> Pavyzdys:</p> <pre>cout << "Atsitiktinis skaičius: " << << rand() << endl;</pre> <p>Ekrane matysite: Atsitiktinis skaičius: 130</p>