

PROGRAMAVIMAS IR ALGORITMAVIMAS

1-oji tema

Algoritmai ir programos. Pagrindiniai programavimo kalbų elementai.

“Taisyklių rinkinys, nustatantis kaip iš pradinių duomenų gauti galutinius duomenis (t. y. rezultatus), vadinamas **algoritmu**.”

Kadangi mus domina ryšis tarp žmogaus ir kompiuterio, mes kalbame, kad algoritmas nusako ką turi padaryti kompiuteris. Tam, kad lengviau suvoktumėte šias ir su jomis susijusias sąvokas, siūlyčiau tokia analogija, kaip valgio ruošimas:

algoritmas - kulinarinis receptas;
procesas - valgio ruošimas;
pradiniai duomenys - žaliava;
rezultatai - paruoštas valgis (patiekalas).

Kompiuteris ir jo įrengimai – virtuvė, virėjas, puodai, įrankiai. Pats žodis “algoritmas” (žodžio atsiradimo istorija vadinasi etimologija), kilęs iš IX a. arabu matematiko ir astronomo al-Chorezmi vardo, kuris sudarė keturių aritmetinių veiksmų su dešimtainiais skaičiais taisyklės. Vėliau Europoje imta vadinti jas “*algorizmais*”, ir dar vėliau – *algoritmais*. Taigi kompiuterį galima pavadinti “algoritmu atlikėju”. Reiškia žmogus turi užrašyti algoritmą abiem (žmogui ir kompiuteriui) suprantama kalba. Informatikoje tokią formalią kalbą, kuria užrašomi algoritmai, įprasta vadinti “programavimo kalba”. Algoritmas, užrašytas konkrečia programavimo kalba, vadinamas “programa”. Aš jau minėjau, kad didžiausias Paskal’io kalbos privalumas, kad jis vienodai gerai tinka (t. y. suprantamas) ir žmogui ir kompiuteriui. Todėl Paskalį galima vadinti ir algoritminė, ir programavimo kalba.

Bet kokia **programavimo kalba** (Pascal/Delphi, C/C++/C#, Basic/Visual Basic, Java, Logo ir kt.) turi suteikti šias galimybes jos naudojančiam žmogui (vartotojui, programuotojui), kad jis galėtų nusakyti kompiuteriui kaip atlikti algoritmą :

- komandos, leidžiančios įvesti pradinius duomenis - **įvedimas** (angl. *input*). Mūsų sąlygomis tai dažniausiai reiškia informacijos iš klaviatūros, diskelio, disko ir pan. įvedimą). Pvz.:
`C: scanf(&x);` `C++: cin >> x; cout << y;`
- išskirti vieta atmintyje (angl. **RAM**) informacijai saugoti ir nurodyti šios informacijos tipą (**duomenų tipą**): konstantos, kintamieji, sveikojo ir realaus tipo skaičiai, tekstai (simboliai ir simbolių eilutės), atsakymai į klausimus (loginės reikšmės) ir kt. Pvz.: `int a,b,c;` `bool b1,b2;`
- turėti efektyvias komandas duomenų apdorojimui, taip vadinamas “**operacijas**”. Jos priskiria vienų kintamųjų reikšmės kitiems, lygina reikšmes ir t. t.
- suteikti vartotojui galimybę gauti rezultatus, t. y. duomenų **išvedimas** (angl. *output*). Mūsų sąlygoms tai dažniausiai reiškia informacijos spausdinimą, atvaizdavimą displejaus ekrane, įrašymą į diskelį arba diską, išsiuntimą į vietinį ar globalų tinklą (elektroninį paštą) ir pan. Pvz.: `C: printf(a);` `C++: cout << y;`
- organizuojant programos instrukcijas vartotojas turi turėti galimybę **nuosekliai nurodyti instrukcijų eilę**.

Programavimo kalba taip pat turi turėti tokias valdymo konstrukcijas (angl. *statement*):

- atlikti kai kurias instrukcijas tik tuo atveju, jei tenkinama tam tikra sąlyga, arba pasirinkti vieną iš kelių galimų veiksmų: **sąlyginis vykdymas**, **sąlyginiai sakiniai** (angl. *conditional execution*). Pvz.:

`if (sąlyga) sak1;`
`else sak2;`

- galimybę pakartoti kokias nors komandas daug kartų. Pakartotiniams veiksmams naudojami **ciklai** (angl. *loops*).
- galimybę padalyti (suskaityti) visą programą į atskiras dalis, kurias galima būtų ruošti, nagrinėti ir vykdyti atskirai. Taigi, reikalingos **paprogramėlės** (angl. *subroutines*). Paskalyje yra dviejų rūšių paprogramėlės, procedūros ir funkcijos, C/C++ tik funkcijos.

Taigi mes išvardinome septynis (tiksliau, 7–8) pagrindinius bet kurios universalios programavimo kalbos elementus.

C / C++ kalbų sandara

Abėcėlė.

Anglų kalbos didžiosios ir mažosios raidės: **ABCD..XYZ abcd..xyz**

Skaičiai (skaitmenys): **0123456789ABCDEF**

Specialūs simboliai: **. , ; : - _ + - * / = > < ^ @ () { } [] .. ' " ! ? | % & # \ ~**

ir dar vienas specialusis simbolis - tarpas

Tokios specialių simbolių poros taip pat traktuojamos kaip specialūs simboliai:

<= >= := .. (* *) (. .)

1. Baziniai žodžiai.

C kalbos rezervuoti (arba baziniai) žodžiai (angl. *reserved words* arba *keywords*). Jų reikšmės yra fiksuotos ir negali pasikeisti. Šie 32 žodžiai ir sudaro C kalbos pagrindą:

auto	const	double	float	int	short	struct	unsigned
break	continue	else	for	long	signed	switch	void
case	default	enum	goto	register	sizeof	typedef	volatile
char	do	extern	if	return	static	union	while

C++ kalboje naudojami ir šie C kalbos rezervuoti žodžiai ir dar 30 naujų:

asm	dynamic_cast	namespace	reinterpret_cast	try
bool	explicit	new	static_cast	typeid
catch	false	operator	template	typename
class	friend	private	this	using
const_cast	inline	public	throw	virtual
delete	mutable	protected	true	wchar_t

Dar 11 C++ kalbos žodžių Jie gali būti naudojami kai taikoma ne ASCII, o kita nors kita koduotė:

and	bitand	compl	not_eq	or_eq	xor_eq
and_eq	bitor	not	or	xor	

Šie žodžiai gali būti naudojami žymint tam tikras C++ operacijas vietoj paprastai naudojamų specialiųjų simbolių, pvz., **and** vietoj **&**, t. y. įprasto loginės operacijos **ir** žymėjimo.

2. Standartiniai žodžiai arba vardai (angl. *predefined words* arba *predefined identifiers*).

Jie taip pat iš anksto apibrėžti, bet jų nereikėtų painioti su baziniais žodžiais, nes jų reikšmės programuotojas gali keisti (tai nepatartina, bet galima). Šie žodžiai žymi standartines funkcijas, manipulatorius (pvz., **endl**) ir kai kuriuos kitus elementus iš C++ bibliotekų.

Čia sąrašas dažniausiai naudojamų (ne visų!) standartinių C++ vardų:

cin	endl	INT_MIN	iomanip	main	npos	std
cout	include	INT_MAX	iostream	MAX_RANDOM	NULL	string

3. Programuotojo (vartotojo) žodžiai arba vardai (angl. *user identifiers*).

Programuotojai tokiais savo sugalvotais vardais gali pavadinti: konstantas, kintamuosius, funkcijas, programas, modulius, duomenų tipus, klases, objektus ir kitus programos elementus. Tačiau,

programuotojas gali kurti savo vardus tik pagal tam tikras taisykles (panašios taisyklės galioja daugumoje universalių programavimo kalbų):

- vardas turi prasidėti raide arba pabraukimo simboliu (angl. *underscores*),
- vardo viduje gali būti tik raidės, skaičiai ir pabraukimo simboliai,
- tarp dviejų vardų būtinas tarpas arba kitas atskyrėjas (: , ; . []), $x + y$
- vardo ilgis nėra ribotas, bet Paskalis atpažins tik pirmuosius 63 simbolius,
- C/C++ skiria didžiąsias ir mažąsias raides.

Pavyzdys.

Taisyklingi vardai: *ManoNamas, Metai_2002, a1, a2, Tu154b, Mazda323, _2021*.

Netaisyklingų vardų pavyzdžiai: *Klaipėda, aa"aa, Mazda-323, Tu 154 b, 2016_Metai, 1a*.

PROGRAMAVIMO KALBŲ SANDARA

Pascal kalbos sandara

Paskalio abėcėlė.

Anglų kalbos didžiosios ir mažosios raidės: **ABCD..XYZ abcd..xyz**

Skaičiai (skaitmenys): **0123456789ABCDEF**

Specialūs simboliai: + - * / = < > [] , () : ; ^ . @ { } \$ #

ir dar vienas specialusis simbolis - tarpas

Tokios specialių simbolių poros taip pat traktuojamos kaip specialūs simboliai:

<= >= := .. (* *) (. .)

Paskalio žodžiai.

- 1) **Rezervuoti žodžiai arba baziniai žodžiai** (angl. *reserved words*). Jų reikšmės yra fiksuotos ir negali pasikeisti. Šie žodžiai ir sudaro Pascal kalbos pagrindą:

*and asm array begin case const constructor destructor div do downto else end exports
file for function goto if implementation in inherited inline interface label library mod
nil not object of or packed procedure program record repeat set shl shr string then to type
unit until uses var while with xor*

- 2) **Standartiniai žodžiai arba vardai**. Jie taip pat iš anksto apibrėžti, bet jų reikšmės programuotojas gali keisti (tai nepatartina, bet galima). Šie žodžiai žymi standartinius duomenų tipus, standartinės procedūras ir funkcijas ir kai kuriuos kitus kalbos elementus

Pavyzdžiui: *sin, cos, exp, integer, char* ir kt.

- 3) **Programuotojo (vartotojo) žodžiai arba vardai** (angl. *identifiers*).

Programuotojai tokiais savo sugalvotais vardais gali pavadinti: konstantas, kintamuosius, funkcijas ir procedūras, programas, modulius, duomenų tipus ir žymes. Tačiau, programuotojas gali kurti savo vardus tik pagal tam tikras taisykles (panašios taisyklės galioja daugumoje universalių programavimo kalbų):

- vardas turi prasidėti raide arba pabraukimo simboliu (angl. *underscores*),
- vardo viduje gali būti tik raidės, skaičiai ir pabraukimo simboliai,
- tarp dviejų vardų būtinas tarpas arba kitas atskyrėjas (: , ; . []),
- vardo ilgis nėra ribotas, bet Paskalis atpažins tik pirmuosius 63 simbolius,
- Paskalis neskiria didžiųjų ir mažųjų raidžių.

Pavyzdys.

Taisyklingi vardai: *ManoNamas, Metai_2002, a1, a2, Tu154b, Mazda323*.

Netaisyklingų vardų pavyzdžiai: *Klaipėda, aa"aa, Mazda-323, Tu 154 b, 2002_Metai, 1a*.