



Rimantas BELEVIČIUS
Saulius VALENTINAVIČIUS

Programavimas

C++20 glaustai

Vadovėlis

Vilnius, 2023

Leidinį rekomendavo Vilniaus Gedimino technikos universiteto
Fundamentinių mokslų fakulteto studijų komitetas

Recenzavo: doc. dr. Mindaugas Rybokas,
Vilniaus Gedimino technikos universitetas,
Informacinių technologijų katedra
prof. dr. Vacius Jusas,
Kauno technologijos universitetas

Bibliografinė informacija pateikiama
Lietuvos integralios bibliotekų informacinės sistemos (LIBIS)
portale *ibiblioteka.lt*

Vilniaus Gedimino technikos universiteto
mokomosios metodinės literatūros knyga (2023-036-S)

ISBN 978-609-476-337-3
eISBN 978-609-476-338-0
doi: 10.20334/2023-036-S

Šį kūrinį, esantį bibliotekose, mokymo ir mokslo įstaigų bibliotekose,
muziejuose arba archyvuose, be leidėjo sutikimo draudžiama mokslinių tyrimų ar
asmeninių studijų tikslais atgaminti, viešai skelbti ar padaryti viešai prieinamą
kompiuterių tinklais tam skirtuose terminaluose tų įstaigų patalpose.
Dėl leidimo kreiptis el. paštu *eleidyba@vilniustech.lt*

TURINYS

PRATARMĖ	9
1. PROGRAMOS KŪRIMO IR VYKDYMO TECHNOLOGIJA	12
1.1. Programos kūrimo ir vykdymo etapai	13
1.2. Algoritmai	14
1.3. Paprasčiausios programos pavyzdys	17
1.4. Programavimo terpės parengimas	18
2. KALBOS ELEMENTAI IR PAPRASČIAUSI OPERATORIAI	23
2.1. Duomenys	23
2.2. Paprasčiausia įvestis ir išvestis	30
2.3. Reiškiniai	31
2.4. Matomumo sritis	34
2.5. Kontroliniai klausimai	35
2.6. Savarankiško darbo užduotys	35
3. PRIESKYROS, CIKLŲ IR SĄLYGŲ OPERATORIAI	36
3.1. Prieskyros operatoriai	36
3.2. Ciklai	38
3.3. Sąlygos	45
3.4. Operatoriai <i>break</i> ir <i>continue</i> cikliuose	51
3.6. Kontroliniai klausimai	52
3.7. Savarankiško darbo užduotys	52
4. MASYVAI	54
4.1. Vienmačiai masyvai	54
4.2. Daugiamatiai masyvai. Išvesties formatavimas	59
4.3. Atminties skyrimas masyvui vykdant programą	64
4.4. Konteineris <i><vector></i>	64
4.5. Kontroliniai klausimai	67
4.6. Savarankiško darbo užduotys	67
5. FUNKCIJOS	69
5.1. Funkcijos idėja	69
5.2. Funkcijos argumentai	71
5.3. Gražinama reikšmė	74
5.4. Argumentų perdavimo mechanizmai	77
5.5. Masyvai-funkcijų argumentai	79
5.6. Funkcijų perkrovimas	82

5.7. Rekursinės funkcijos	83
5.8. Kai kurios kitos funkcijų galimybės	86
5.9. Lambda išraiškos	90
5.10. Kontroliniai klausimai	93
5.11. Savarankiško darbo užduotys	94
6. PAPRASČIAUSI ALGORITMAI MASYVAMS	95
7. KINTAMŲJŲ MATOMUMAS IR GYVAVIMAS. MODULIAI IR VARDŲ SRITYS	103
7.1. Lokalieji kintamieji	103
7.2. Globalieji kintamieji	105
7.3. Statiniai lokalieji kintamieji	107
7.4. Moduliai	108
7.5. Vardų sritys	112
7.6. Kontroliniai klausimai	116
8. DUOMENŲ MAINAI SU FAILAIS	117
8.1. Įvadinės žinios	117
8.2. Srautų klasės ir pagrindiniai jų metodai	119
8.3. Srautų klasės <i>istream</i> ir <i>ostream</i>	123
8.4. Srautų formatavimas	126
8.5. Srautų klaidos. Srauto pabaiga. Srautas-funkcijos argumentas	128
8.6. Kontroliniai klausimai	132
8.7. Savarankiško darbo užduotys	133
9. EILUTĖS. STRUKTŪROS	134
9.1. C eilutės	134
9.2. Klasės <i>string</i> eilutės	140
9.3. Struktūros	143
9.4. Kontroliniai klausimai	148
9.5. Savarankiško darbo užduotys	149
10. RODYKLĖS	150
10.1. Įvadinės žinios apie rodykles. Rodyklės ir nuorodos	150
10.2. Netiesioginė kreiptis į kintamąjį	153
10.3. Rodyklės į masyvus	154
10.4. Argumentų perdavimas į funkciją rodyklėmis	156
10.5. Dinaminės atminties skyrimas	161
10.6. Rodyklės į eilutes	166
10.7. Išmaniosios rodyklės	170
10.8. Rodyklės į objektus. Rodyklių į objektus masyvai	171
10.9. Rodyklės į funkcijas. Funkcijos grąžinama reikšmė-rodyklė	183
10.10. Reikšminis žodis <i>this</i>	187
10.11. Kontroliniai klausimai	188
10.12. Savarankiško darbo užduotys	189

11. KLASĖS	190
11.1. Įvadinės žinios apie klases	190
11.2. Metodai-konstruktoriai ir destruktoriai	192
11.3. Išsamiau apie metodus ir konstruktorius su numatytosiomis argumentų reikšmėmis ..	198
11.4. Kopijos konstruktorius	205
11.5. Modifikatorius <i>const</i> klasėje. Statiniai laukai ir metodai	209
11.6. Draugiškos funkcijos ir klasės	214
11.7. Įdėtinės klasės. Vidinės klasės	216
11.8. Darbo su objektine programa technologija	218
11.9. Kontroliniai klausimai	221
11.10. Savarankiško darbo užduotys	221
12. OPERATORIŲ PERKROVIMAS. TIPŲ PERTVARKYMAS. DUOMENŲ PERVARDIJIMAS	223
12.1. Vianarių operacijų perkrovimas	224
12.2. Dvinarių operacijų perkrovimas	227
12.3. Duomenų tipų pertvarkymas	232
12.4. Duomenų pervardijimas	234
12.5. Kontroliniai klausimai	235
12.6. Savarankiško darbo užduotys	235
13. PAVELDIMUMAS	236
13.1. Bendrosios žinios	236
13.2. Išvestinės klasės konstruktoriai	239
13.3. Objektų tipų pertvarkymai	242
13.4. Išvestinės klasės kopijos konstruktorius	245
13.5. Paveldimumas ir destruktoriai	246
13.6. Paveldimumas ir metodų pakeitimas	247
13.7. Bendrasis ir dalinis paveldimumas	249
13.8. Daugybinis paveldimumas	252
13.9. Kontroliniai klausimai	256
13.10. Savarankiško darbo užduotys	257
14. POLIMORFIZMAS	258
14.1. Polimorfizmo idėja. Virtualios funkcijos ir abstrakčios klasės	258
14.2. Polimorfinės programos saugikliai	264
14.3. Polimorfizmo realizavimas per nuorodas į objektus	265
14.4. Polimorfinės programos destruktoriai	267
14.5. Dinaminė informacija apie objektų tipus. Dinaminis tipų pertvarkymas	270
14.6. Kontroliniai klausimai	276
14.7. Savarankiško darbo užduotys	277
15. IŠIMTYS	278
15.1. Pradinės žinios apie išimtis	278
15.2. Paveldimumas išimčių klasėse	285

15.3. Pakartotinis klaidos sukėlimas. Operatorius <i>catch(...)</i>	288
15.4. Funkcijos <i>noexcept</i>	289
15.5. Standartinės C++ išimtys	290
15.6. Kontroliniai klausimai	292
15.7. Savarankiško darbo užduotis	292
16. ŠABLONAI	293
16.1. Funkcijų šablonai	293
16.2. Kitos šabloninių funkcijų galimybės.	297
16.3. Klasių šablonai	300
16.4. Kitos šabloninių klasių galimybės.	302
16.5. Apribotieji šablonai. <i>concepts</i> ir <i>requires</i>	305
16.6. Kontroliniai klausimai	308
16.7. Savarankiško darbo užduotys.	309
17. SEKLUSIS KOPIJAVIMAS	310
17.1. Kairieji ir dešinieji reiškiniai	310
17.2. Giliojo ir sekiojo kopijavimo palyginimas	311
17.3. Prieskyros-kreipties operatorius	319
17.4. Numatytieji kreipties operatoriai.	321
17.5. Kontroliniai klausimai	322
17.6. Savarankiško darbo užduotis	322
18. KELIOS PAGALBINĖS KLASĖS	323
18.1. <i>std::optional<></i>	323
18.2. <i>std::span<></i>	326
18.3. <i>std::string_view</i>	328
18.4. Kontroliniai klausimai	329
19. TRUMPAS ĮVADAS Į SL KLASES	330
19.1. Konteineriai	330
19.2. Algoritmai	338
19.3. Kitos algoritmų galimybės.	341
19.4. Kontroliniai klausimai	342
20. PROGRAMŲ DERINIMAS	343
20.1. Pagrindinės VS derintuvo funkcijos	344
PABAIGA	348
LITERATŪRA	351
RODYKLĖ	352
SAVARANKIŠKO DARBO UŽDUOČIŲ ATSAKYMAI	355
PRIEDAS	435

PRATARMĖ

Šis vadovėlis parašytas mūsų ankstesnio 2015 metų vadovėlio *Programavimas C++* pagrindu, atsisakius kai kurių senstelėjusių temų, išsamiai aprašant naująsias C++ kalbos galimybes ir daugiau dėmesio skiriant gerajam programavimo C++ kalba stiliui. Nuo senojo vadovėlio laikų C++ kalba stipriai pakito, patvirtinti keli nauji kalbos standartai – C++14 (šis ir visi vėlesni standartai dar vadinami *moderniąja C++*), C++17, C++20 ir jau kalbama apie būsimąjį standartą C++23. Kalbos standartas – tai tarptautinės ISO (*International Organization for Standardization*) organizacijos patvirtintas kalbos konstrukčių rinkinys, kurį privalo realizuoti visi tą standartą savo pavadinime deklaruojantys kompiliatoriai.

Šis vadovėlis skirtas studentams, klausantiems programavimo dalykus. Jo tikslas – suteikti žinių ir įgūdžių, padėsiančių sėkmingai programuoti tiek procedūrinio, tiek objektinio stiliumi. Vadovėlio pabaigoje taip pat trumpai užsimenama apie funkcinį programavimo stilių ir pateikiami keli tokiu stiliumi parašyti programų pavyzdžiai. C++ neverčia programuoti griežtai kuriuo nors iš išvardintų stilių, skirtingai, pavyzdžiui, nuo tik procedūrinės C kalbos, griežtai objektinės – *Java* kalbos, griežtai funkcinės – *Haskell*. Pirmuosiuose 10 vadovėlio skyrių programos rašytos procedūrinio stiliumi, o tolesniuose – dažniausiai objektinio. Tačiau vadovėlyje nėra nubrėžta aiški takoskyra tarp šių dviejų programavimo stilių, nes C++ kalba, griežtai kalbant, grynai objektinės programos parašyti ir negalima. Knygos apimtis pakankama vienu metų dviejų studijų dalykų studijoms. Vadovėlio medžiaga bus gera įžanga į standartinį duomenų struktūrų ir algoritmų dalyką, skaitomą visų informatikos ir daugelio matematikos studijų programų studentams.

Vadovėlis skirtas skaitytojui, turinčiam tik pačių elementariausių informatikos ir matematikos žinių. 1 knygos skyriuje trumpai pakartotos įvadinės informatikos žinios ir sąvokos. Šiame skyriuje taip pat glaustai paaiškinti naudojimosi programavimo terpėmis, kurios gerokai palengvina darbą su programomis, pagrindai. Kad būtų galima vykdyti vadovėlyje pateiktas programas, būtina kompiuteryje turėti bent jau kažkurį C++20 standarto kompiliatorių ir kurį nors teksto redaktorių. Patogu naudotis programavimo terpėmis (IDE, *Integrated Development Environment*), kuriose iškart yra kompiliatorius, specializuotas teksto redaktorius, programos derinimo ir kiti įrankiai. Aptariamą kalbos standartą šiuo metu palaiko nemokami atvirosios prieigos kompiliatoriai *CLang* ir *GCC*, o šie kompiliatoriai įdiegti į laisvai prieinamas programavimo terpes *Code::Blocks* ir *QT Creator*. Vadovėlyje parodyta, kaip naudotis profesionalia *Microsoft Visual Studio 2022* terpe (sutrumpintai ją vadinsime *VS2022*), kurios kompiliatorius realizuoja visas C++20 galimybes (terpėje įdiegti ir kitų programavimo kalbų, pavyzdžiui, *C#* ar *Javascript*, kompiliatoriai). Nors *VS2022* yra profesionalus įrankis, asmeniniam (ir net profesionaliam nedidelėse kompanijose) naudojimui skirta *Community* licencija yra nemokama.

2–19 skyriuose jau kalbama apie programavimą C++ kalba ir išdėstoma praktiškai visa standartinė C++ sintaksė. Kiekvieno skyriaus pabaigoje yra kontroliniai klausimai, skirti pasitikrinti, ar suvokėte skyriaus medžiagą, ir užduotys programoms rašyti. Vadovėlio pabaigoje yra ir mūsų parašytų programų toms užduotims tekstai.

Standartinėse C++ programose galima taikyti tik lotynišką abėcėlę. Norėdami lietuviškas raides rašyti programos spausdiniuose ar ir pačioje programoje – identifikatoriuose ir literaluose – turite į programą įtraukti papildomą biblioteką ir iškviešti lietuvinimo funkciją. Kaip tą padaryti, paaiškina priede pateikta programa ir jos komentarai.

Negalvokite, kad perskaitę ir perpratę šią knygą tapsite programavimo guru. Stengėmės parašyti kiek įmanoma glaustą vadovėlį. Nors jame išaiškintos visos esminės C++20 konstrukcijos ir galimybės, vis tik paėmę profesionalią C++ kalba parašytą programą, greičiausiai beveik nieko joje nesuprasite. Taip yra dėl to, kad profesionalūs programuotojai kur tik įmanoma stengiasi rašyti programas ne nuo nulio, o naudotis kitų žmonių įdirbiu – taikyti klasių ir funkcijų bibliotekas. Vadovėlio skyriuose rasite trumpą įvadą į SL (*Standard Library*) biblioteką, skirtą duomenų struktūroms bei su jomis veikiančioms algoritmams. Norint išsamiai apžvelgti šios bibliotekos galimybes, tektų rašyti atskirą ir didelės apimties vadovėlį. Tačiau išstudijavę vadovėlio medžiagą, žinosite kalbos sintaksę ir todėl galėsite suprasti IDE pagalbos failų informaciją ir galėsite skaityti programų bibliotekoms skirtas knygas. Ir dar viena pastaba: net jei ir atidžiai perskaitysite ir suvoksite viską, kas parašyta vadovėlyje – patys nbandydami rašyti programų vis tik programuoti neišmoksitė. Todėl po kiekvieno skyriaus pateikėme bent po kelias programavimo užduotis ir labai rekomenduojame tas užduotis įvykdyti. Savikontrolei knygos gale rasite ir mūsų parašytas programas su komentarais tiems uždaviniams (tačiau tos programos, be abejo, nėra vienintelis galimas uždavinių sprendimo būdas; jūs galite sugalvoti ir kitokias visiškai teisingas programas). Jei programos užduotis buvo gana sudėtinga, kad būtų galima rinktis rašyti efektyvią programą (naudojančią kiek įmanoma mažiau operatyviosios atminties ir reikalaujančią mažiausiai aritmetinių operacijų) ar ne tiek efektyvią, tačiau paprastą, aiškesnę struktūrą – rinkomės paskutinį variantą. Programose, taikančiose klasės sintaksę, rasite ir perteklinio klasių funkcionalumo: daugelyje programų yra visi konstruktoriai ir destruktoriai, tarnybiniai metodai, nors startinė funkcija dalies to funkcionalumo visai nenaudoja. Taip programas rašėme specialiai, kad įprastumėte parašyti tvarkingas klases. Beje, kai kurie uždaviniai pradedančiam programuotojui atrodys sudėtingi. Visos mūsų programos derintos VS2022 terpės kompiliatoriumi. Jei norėsite eksperimentuoti su mūsų programomis, kad nereikėtų iš naujo rinkti teksto, visi programų tekstai yra <https://github.com/Programavimas-CPP-Vadovelis/Vadovelyje-naudojamas-programinis-kodas>. Savarankiško darbo užduotims pasitikrinti mūsų parašytos programos (minėjome, kad jūsų programos gali skirtis) yra <https://github.com/Programavimas-CPP-Vadovelis/Savarankisko-darbo-uzduociu-atsakymai>.

Kaip matysite iš programų pavyzdžių, tuos pačius dalykus C++ kalboje galima užprogramuoti keliais skirtingais būdais, kurių kai kurie paveldėti iš senesnių kalbos standartų, kai kurie įvesti visai neseniai, tik 20-ajame standarte. Vadovėlio programose parodome visas tas alternatyvas, kad, nagrinėdami kitų autorių programas, nenustebtumėte pamatę

nesuprantamas konstrukcijas. Dažniausiai savo programose iš pradžių taikome paprasčiausias programavimo alternatyvas, o paaiškinę naujesnes konstrukcijas, tolesnėse programose jau dažniausiai naudojame būtent naujasias. Dalis sudėtingesnių programų parašytos keliais variantais: taikant arčiausiai „kompiuterio geležies“ esančias konstrukcijas (jos iš esmės apsiriboja C++11 standartu ar atėję dar iš C kalbos), taikant C++20 galimybes ir dažnai dar trečias variantas – naudojant šabloninius SL konteinerius ir algoritmus.

Jei galvosite savo būsimą darbą sieti su programavimu, turėsite išmokti ir kitų programavimo kalbų. Tam C++20 yra tikrai geras pagrindas, kadangi daugelis vėlesnių, labai populiarių šiandien kalbų sukurtos C++ sintaksės pagrindu (*Java*, *Python*, *Scala* ir kt.) ir dažnai dar teikia programuotojui ne visas C++ galimybes. Pavyzdžiui, *Java* kalboje iš viso nerasite rodyklių, daugybinio paveldėjimo ir t. t.

Knygos pabaigoje rasite papildomam skaitymui rekomenduojamų knygų sąrašą. Čia išvardytos, autorių nuomone, geriausios pasaulinės C++ kalbai skirtos knygos (jos – anglų kalba). Taip pat nurodome kelias, mūsų nuomone, vykusias knygas apie *Java*, *Python*’ą ir *Scala*.

Vadovėlyje išsamiai arba lakoniškai kalbama beveik apie visas C++20 sintaksės galimybes. Ko čia nerasite – tai bitinių operacijų, paveldėtų iš C kalbos ir skirtų operuoti duomenimis žemu lygiu – ne baitais, o atskirais bitais. Tik trumpas įvadas pateiktas į SL konteinerius, algoritmus, diapazonus ir filtrus. Dabartinis C++ kalbos standartas turi sintaksės konstrukcijas lygiagretiesiems skaičiavimams keletu branduolių kompiuteriuose programuoti. Vadovėlyje tik parodėme, kad tokios galimybės yra, bet jų išsamiai nesiaiškinome, kadangi manome, jog pradedančiojo programuotojo parašyta programa lygiagretiesiems skaičiavimams bus neišvengiamai lėtesnė už viengiję, kompiliatoriaus optimizuotą programą. Visų C++20 kalbos galimybių aprašymas lemtų dar kelis šimtus papildomų teksto puslapių. Visas šias žinias galite rasti išsamesnėje, pavyzdžiui, I. Hortono ir P. Van Weerto knygoje.

Kad vadovėlio tekstas būtų lengviau perskaitomas, programų fragmentai ir visi pilni kalbos operatoriai bei programų rezultatai pateikiami *skirtingu šriftu, dydžiu ir kursyvu*, o visi komentarai – dar ir kita *spalva*. Kai į tekstą įterpiamos visos iš programavimo terpės nukopijuotos programos – jų standartiniai operatoriai bus tokios *spalvos*, o programos spausdinimai – *tokios* (spalvas matysite tik skaitmeninėje knygos formoje). Kai tekste tenka parašyti kalbos operatorių fragmentus – jie išskiriami *kursyvu*. Dažnai šalia lietuviškų kalbos konstruktų pavadinimų pateikiame ir originalius angliškus pavadinimus – jie taip pat išskiriami *kursyvu*.