

# Procedūrinio programavimo pagrindai

## 7 tema

Procedūrinio programavimo esmė – kodo skaidymas į paprogrames (procedūras, funkcijas). Šių užduočių tikslas – išmokti apibrėžti ir iškviešti savo paties sukurtas funkcijas reikiamoms užduotims atlikti.

Kiekvienai iš užduočių skirtą kodą rašykite atskirame kodo faile. Funkcija *main* yra skirta parašytos funkcijos testavimui ir demonstracijai, kaip veikia jūsų parašytas kodas. Specialios vartotojo sąsajos daryti nereikia – pratybų rezultatas yra funkcija, o ne *main* esantis kodas.

Atliktas užduotis įkelkite į VU VMA, laikydamiesi pateiktų nurodymų.

### Užduotis 1.

Apibrėžkite funkciją, kuri patikrina, ar duotas skaičius papuola į nustatytą intervalą.

Funkcijos prototipas: `int isInRange(int number, int low, int high)`

Aprašymas: funkcija turi grąžinti *true*, jei skaičius *number* priklauso intervalui *[low; high]* (su sąlyga, kad *low* neviršija *high*), kitaip ji turi grąžinti *false*.

### Užduotis 2.

Apibrėžkite funkciją, kuri suskaičiuoja duoto skaičiaus faktorialą.

Funkcijos prototipas: `int getFactorial(int number)`

Aprašymas: funkcija skaičiavimams naudoja rekursiją ir grąžina neneigiamo skaičiaus *number* faktorialą, o jei jo suskaičiuoti negalima – grąžina 0.

Įdomu: apibrėžkite (kitą) funkciją, kuri duoda tą patį rezultatą, bet rekursija nesinaudoja.

### Užduotis 3.

Apibrėžkite funkciją, kuri leidžia gauti (įvesti) vieną skaičių iš standartinio įvedimo srauto.

Funkcijos prototipas: `int getPositiveNumber(char *msg)`

Aprašymas: funkcija atspausdina į ekraną tekstinę eilutę *msg* (kurioje tikisi gauti vartotojui skirtą pranešimą apie tai, ką reikia įvesti), ir ją (eilutę ekrane) kartoja (prašo vartotojo įvesti vėl ir vėl) tol, kol vartotojas neįveda vieno skaičiaus eilutėje, kurį ši funkcija ir grąžina.

### Užduotis 4.

Apibrėžkite funkciją, kuri leidžia užpildyti masyvą atsitiktinėmis reikšmėmis.

Funkcijos prototipas: `void generateArray(int data[], int size, int low, int high)`

Aprašymas: kiekvienam iš *size* elementų, esančių masyve *data*, funkcija turi priskirti atsitiktinę reikšmę iš intervalo *[low; high]*; nieko papildomai grąžinti nereikia, t.y. funkcijos rezultatas yra masyve esančios (naujos) reikšmės.

### Užduotis 5.

Apibrėžkite funkciją, kuri grąžina failo dydį baitais.

Funkcijos prototipas: `long getFileSize(char *fileName)`

Aprašymas: funkcija gauna failo vardą kaip parametą, bando jį atidaryti (skaitymui binariniu režimu), ir peršokus į failo pabaigą (`fseek`) sužinoti (`ftell`) kiek baitų jis užima; tą baitų skaičių funkcija ir grąžina (patikrinkite su OS rododomis reikšmėmis), o jei kažkas nepavyko (pvz. tokio failo nėra) – funkcija grąžina reikšmę -1.

### Užduotis 6.

Apibrėžkite funkciją, skirtą vartotojo sąsajoje pateikiamam meniu spausdinti ir vartotojo pasirinkimui gauti.

Funkcijos prototipas: `int showMenu(char *menuTitle, char *menuOptions[], int menuSize, char *inputMsg)`

Aprašymas: funkcija turi atspausdinti meniu antraštę (parametras *menuTitle*), tuomet atspausdinti meniu iš *menuSize* pasirinkimų, kur meniu elementų (eilučių) pavadinimai pateikiami (eilučių) masyve *menuOptions*. Tuomet funkcija atspausdina tekstinę eilutę *inputMsg*, kurioje perduodamas vartotojui skirtas tekstas, kuriame prašoma įvesti savo pasirinkimą, ir gavusi iš vartotojo skaičių (atitinkamo meniu punkto numerį) jį ir grąžina; jeigu vartotojo įvedimas nekorektiškas (įvedamas ne skaičius arba tokio meniu punkto nėra), visas spausdinimo procesas (antraštė, meniu, prašymas įvesti) kartojamas iš naujo, tol, kol įvedimas netaps korektiškas.

### Papildomos užduotys.

### Užduotis 7.

Apibrėžkite keturias funkcijas, skirtas binariniame faile saugomiems skaičiams apdoroti:

- a) funkcija *getValue* nuskaityto faile *file* įrašytą skaičių, esantį *index* pozicijoje, kurį ir grąžina
- b) funkcija *setValue* įrašo naują reikšmę *value* į failo *file* poziciją, nurodytą parametru *index*
- c) funkcija *addValue* įterpia naują reikšmę *value* į failo *file* poziciją, nurodytą parametru *index*
- d) funkcija *remValue* ištrina reikšmę iš failo *file* pozicijos *index*

Laikykite, kad failas *file* jau atidarytas skaitymui. Užtikrinkite, kad *index* neperžengia leistinų rėžių (tam galima pasinaudoti funkcija *getFileSize* iš 5 užduoties), priešingu atveju funkcija nieko nedaro ir grąžina reikšmę -1 (sėkmės atveju – 0).

### Užduotis 8.

Įgyvendinkite (skaičių) greitojo rikiavimo (quicksort) algoritmą (pseudokodą su paaiškinimais rasite internete), dviem būdais (t.y. apibrėžkite dvi funkcijas, kurių prototipai sutampa, bet realizacija skiriasi) – su rekursija (funkcija kreipiasi į save) ir be rekursijos (būsimoms iteracijoms įsiminti naudokite specialiai tam skirtą masyvą). Jei reikia, galite apibrėžti papildomų funkcijų užduočiai atlikti, jei jų reikia. Slenksčio parinkimui naudokite medianos-iš-trijų algoritmą.

### Užduotis 9.

Papildykite 6 užduotyje apibrėžtą funkciją taip, kad funkcija galėtų ne tik atspausdinti meniu ir fiksuoti vartotojo pasirinkimą, bet ir iškviesti (kažkokią) funkciją, sietiną su vieno ar kito meniu punkto įvykdymui. Tam reikės apibrėžti papildomą parametą, kuriame bus saugomas kviestinių funkcijų sąrašas, o gavus vartotojo pasirinkimą – ne jį grąžinti, bet iškart perduoti valdymą atitinkamai funkcijai.