

Individuali užduotis sesijos metu - aktorių modelis

Užduotis

Programos duomenų failai tokie pat, kaip L1 ir L2 lab. darbų, gautas rezultatų failas — taip pat toks pat, kaip L1 ir L2 metu. Programa realizuojama naudojant **JavaScript** programavimo kalbą su **node** ir **Nact** biblioteką (išsiaiškinti naudojimo principus savarankiškai). Programos bibliotekų valdymui naudoti JavaScript paketų tvarkytuvą **npm**.

Sukuriami tokie aktoriai (sukurti aktorių hierarchiją savarankiškai):

- skirstytuvai — skirsto žinutes tarp kitų aktorių;
- darbininkai (kiekis toks pat, kaip L1 ir L2 programose) — skaičiuoja pasirinktos funkcijos rezultatą ir filtruoja duomenis;
- rezultatų kaupiklis — saugo atfiltruotus rezultatus surikiuotame masyve;
- spausdintojas — išvedinėja rezultatus į failą.

Programa iš failo nuskaityto duomenis, sukuria reikalingus aktorius ir skirstytuvui po vieną persiunčia nuskaitytus duomenis. Skirstytuvai privalo priimti tokias žinutes:

- iš pagrindinio scenarijaus priima duomenis, perskaitytus iš failo — gavęs žinutę, ją persiunčia vienam iš darbininkų apdorojimui; reikia sugalvoti ir realizuoti darbų paskirstymo algoritmą.
- iš darbininko priima apskaičiuotą rezultatą ir persiunčia jį rezultatų kaupikliui.

Taip pat skirstytuvai darbo pabaigoje rezultatus, saugomus kaupiklyje, persiunčia spausdintojui.

Darbininkai priima žinutes iš skirstytuvo, apskaičiuoja funkcijos rezultatą ir jei jis tenkina pasirinktą kriterijų, persiunčia atgal skirstytuvui.

Rezultatų kaupiklis priima rezultatus ir juos saugo savo masyve.

Spausdintojas priima rezultatų masyvą ir jį spausdina lentele į failą.

Visi procesai gali siuntinėti ir priimti daugiau žinučių, nei aprašyta — čia pateikta dalis komunikacijos tarp procesų, likusią dalį sugalvoti ir realizuoti savarankiškai.

Jei įrašas atitinka pasirinktą kriterijų, jo „kelionė“ turėtų būti tokia: failas -> pagrindinis scenarijus -> skirstytuvai -> darbininkas -> skirstytuvai -> rezultatų kaupiklis -> skirstytuvai -> spausdintojas. Jei įrašas neatitinka kriterijaus, jo „kelionė“ užsibaigia darbininke, jis toliau nebesiunčiamas.

Papildomas reikalavimas programai: programoje negali būti kintamųjų, tik funkcijos ir konstantos. Rekomenduojama vietoj ciklų naudoti **map** ir **forEach** masyvų metodus; darbininkų sukūrimui gali būti naudojamas masyvas su skaičiais nuo 0 iki n: `[...Array(n).keys()]` (grąžins masyvą `[0, 1, ..., n-1]`).

Individulios užduoties vertinimas

- Programos veikimas ir realizacija — 20 taškų.
- Atsakymai į dėstytojo klausimus apie programą — 5 taškai.

- Programos apiforminimas (komentarai, kodo tvarkingumas) — 5 taškai.
- Atsakymas žodžiu į dėstytojo pasirinktą klausimą iš egzamino medžiagos — 5 taškai.

Darbą įkelti į Moodle iki 01-04 12:00.