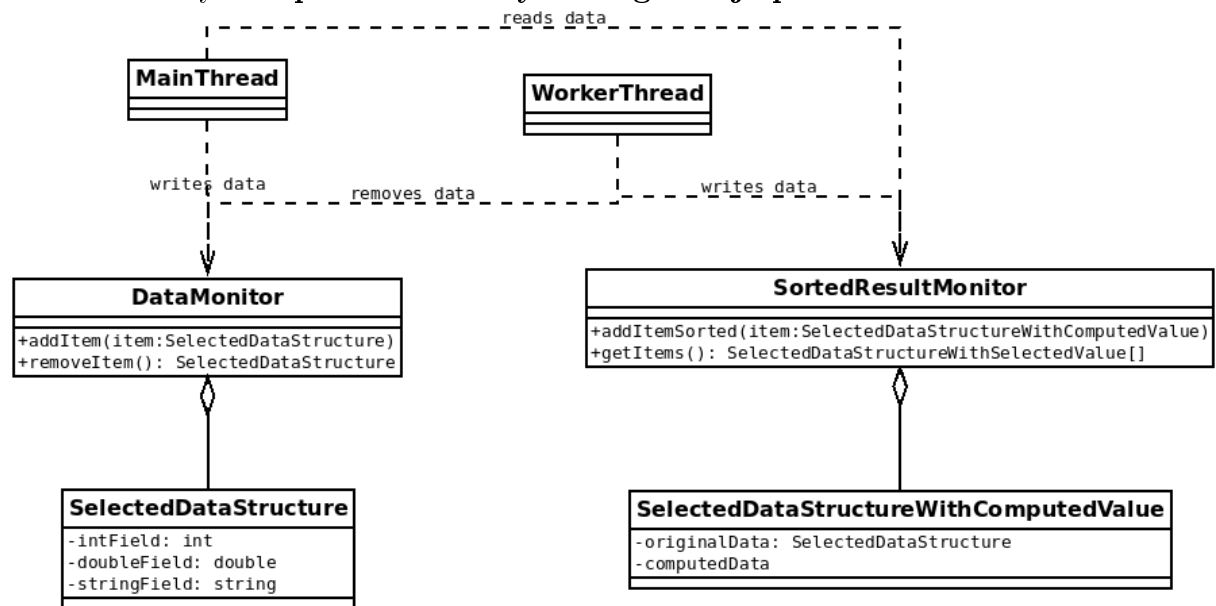


## L1. Bendra atmintis

### Užduotis

Paveiksle pateikiamas apibendrintas vaizdas, kaip turi veikti sukurta programa: sukuriame du monitoriai — duomenims ir rezultatams. **Vienu metu** pagrindinė gija ir pasirinktas kiekis darbininkų gijų dirba su duomenų monitoriumi. Pagrindinės gijos tikslas — visus duomenis, perskaitytus iš failo, sudėti į riboto dydžio duomenų monitorių, palaukti, kol darbininkės baigs darbą ir rezultatus iš rezultatų monitoriaus išvesti į rezultatų failą. Darbininkų tikslas — iš duomenų monitoriaus paimti (pašalinti) įrašą, suskaičiuoti rezultatą, nuspręsti, ar jis turi patekti į rezultatus, ir jei turi, jį įrašyti į rezultatų monitorių taip, kad monitoriaus saugomas masyvas liktų surikiuotas; veiksmus kartoti, kol apdorojami visi duomenys. **Neprivaloma laikytis diagramoje pateiktos struktūros.**



Pasirinkti savo duomenų struktūrą iš 3 laukų — vienas **string**, vienas **int** ir vienas **double** tipo, operaciją (funkciją), kuri iš šios struktūros duomenų suskaičiuotų naują reikšmę bei kriterijų, pagal kurį gautos reikšmės bus išfiltruojamos.

Paruošti tris duomenų failus, kuriuose būtų **bent po 25** pasirinktos struktūros įrašus. Duomenų failo formatas pasirenkamas laisvai. **Rekomenduojama** naudoti standartinį duomenų formatą (JSON, XML ar kt.) ir jam skaityti tinkamas priemones. Duomenų failo pavadinimas — **Grupe\_PavardeV\_L1\_dat\_x.json** (Grupe — Jūsų akademinė grupė, PavardeV — Jūsų pavardė, v., x — eilės nr., failo plėtinys — pagal pasirinktą formatą). Duomenų failai paruošiami tokie:

- **Grupe\_PavardeV\_L1\_dat\_1.json** — visi įrašai atitinka pasirinktą filtravimo kriterijų;
- **Grupe\_PavardeV\_L1\_dat\_2.json** — dalis įrašų atitinka pasirinktą filtravimo kriterijų;
- **Grupe\_PavardeV\_L1\_dat\_3.json** — nei vienas įrašas neatitinka pasirinkto filtravimo kriterijaus;

**Pagrindinė gija** atlieka tokius veiksmus duotu eiliškumu:

1. Nuskaito duomenų failą į lokalią masyvą, sąrašą ar kitą duomenų struktūrą;

2. Paleidžia pasirinktą kiekį gijų  $2 \leq x \leq \frac{n}{4}$  ( $n$  — duomenų kiekis faile), kurios ims įrašus, vykdys su jais pasirinktą operaciją ir, jei rezultatas atitinka pasirinktą kriterijų, įrašys į rezultatų struktūrą.
3. Į duomenų struktūrą, iš kurios gijos ims įrašus, įrašo visus nuskaitytus duomenis.
4. Palaukia, kol visos paleistos gijos baigs darbą.
5. Iš duomenų struktūros, kurioje saugomi gauti atfiltruoti rezultatai, duomenis išveda į tekstinį failą lentelė.

Gijų darbui keliami tokie reikalavimai:

- Kai apdoroti visi duomenys, gijos pačios baigia darbą; kol bent viena gija dirba, pagrindinė gija turi laukti jų darbo pabaigos.
- Duomenys gijoms perduodami naudojant sinchronizuotą duomenų struktūrą — monitorių. Jį rekomenduojama realizuoti kaip klasę ar struktūrą, turinčią bent dvi operacijas: elementui įdėti bei pašalinti. Keliami reikalavimai:
  - Duomenys viduje saugomi **masyve** (sąrašas ar kita struktūra **netinka**), kurio dydis neviršija pusės pasirinkto duomenų kiekio.
  - Jei iš monitoriaus bandoma paimti įrašą, kai vidinis masyvas tuščias, arba įdėti įrašą, kai vidinis masyvas pilnas:
    - \* programai a: gija **blokuojama**. Gijos blokavimui naudojamos pasirinktos kalbos priemonės sąlyginei sinchronizacijai.
    - \* programai b: grąžinamas rezultatas, kad nepavyko atlikti operacijos ir elementą reikia bandyti įterpti/pašalinti iš naujo.
- Rezultatų duomenų struktūra turi tenkinti tuos pačius reikalavimus, kaip pradinių duomenų struktūra, išskyrus masyvo dydžio reikalavimą — masyvas parenkamas tokio dydžio, kad tilptų visi rezultatai. Taip pat rezultatų duomenų struktūra po kiekvieno įrašymo į ją turi likti **surikiuota**, t.y., duomenys į masyvą įterpiami reikiamoje vietoje.

Rezultatų failas pavadinamas **Grupe\_PavardeV\_L1\_rez.txt**. Failas formatuojamas lentelė su antrašte, jame išvedami apskaičiuoti atfiltruoti rezultatai kartu su pradiniais duomenimis.

---

### Laboratorinio darbo programos:

- a) C++, Java, C#, Go — monitoriui kurti panaudojamos pasirinktos kalbos priemonės;
  - b) C++ & OpenMP — monitoriui kurti panaudojamos OpenMP priemonės;
- Abi programos veikia vienodai ir naudoja tuos pačius duomenų failus.

### L1 programų vertinimas

- L1a — 3 taškai
- L1b — 3 taškai

- Kontrolinis — 4 taškai

---

Laboratorinio darbo atsiskaitymo savaitės: a) 5, b) 6, kontr.) 8.

LD programų veikimą demonstruoti užsiėmimų laiku pagal tvarkaraštį, programų (.java, .cpp, .cs, .go), duomenų ir rezultatų failus (.txt) pateikti Moodle prieš atsiskaitymą.

---

### Rekomenduojamos priemonės JSON formato skaitymui

- C++ — nlohmann
- Java — org.json
- C# — DataContractJsonSerializer
- Go — Unmarshal

### Duomenų ir rezultatų failų pavyzdžiai

Duomenų faile pateikiami studentų duomenys. **Pastaba:** faile turėtų būti daugiau duomenų; studentai turėtų sugalvoti savo duomenų struktūrą.

```
{
  "students": [
    {"name": "Antanas", "year": 1, "grade": 6.95},
    {"name": "Kazys", "year": 2, "grade": 8.65},
    {"name": "Petras", "year": 2, "grade": 7.01},
    {"name": "Sonata", "year": 3, "grade": 9.13},
    {"name": "Jonas", "year": 1, "grade": 6.95},
    {"name": "Martynas", "year": 3, "grade": 9.13},
    {"name": "Artūras", "year": 2, "grade": 7.01},
    {"name": "Vacys", "year": 2, "grade": 8.65},
    {"name": "Robertas", "year": 3, "grade": 6.43},
    {"name": "Mykolas", "year": 1, "grade": 6.95},
    {"name": "Aldona", "year": 3, "grade": 9.13},
    {"name": "Asta", "year": 2, "grade": 7.01},
    {"name": "Viktoras", "year": 2, "grade": 8.65},
    {"name": "Artūras", "year": 5, "grade": 8.32},
    {"name": "Vytas", "year": 3, "grade": 7.85},
    {"name": "Jonas", "year": 1, "grade": 6.95},
    {"name": "Zigmas", "year": 3, "grade": 9.13},
    {"name": "Artūras", "year": 2, "grade": 7.01},
    {"name": "Simas", "year": 3, "grade": 6.43}
  ]
}
```

Duotiesiems duomenims buvo pritaikyta operacija — vardas, metai ir pažymys apjungti į vieną tekstinę eilutę ir gautam rezultatui apskaičiuota SHA1 maišos funkcija. Faile išvedami tik tie rezultatai, kurie prasideda raide, ne skaitmeniu. Rezultati surikiuoti pagal abėcėlę.

Name	Year	Grade	Hash
Jonas	1	6,95	A18EAC8F30AC0FC630AE175A851CA5DA24FA8C85
Jonas	1	6,95	A18EAC8F30AC0FC630AE175A851CA5DA24FA8C85
Antanas	1	6,95	C4DE32954C067D224E22D55EB51F3774C3514B53
Asta	2	7,01	DC9556314AD1156B6E0CDBCB0927213B3316298D
Kazys	2	8,65	E763185F4B7303A787ACC513B7AA56706C7A42AC
Aldona	3	9,13	F1276E77671506983D8AE579D5BCC497F2F61346
Mykolas	1	6,95	FCBAB847E2FA87F8D54B77ED416B274D8677C61A