

# Egzaminų vertinimo sistema

**Sistemos tikslas** – ištaisyti gautus studentų egzaminus ir įrašyti gautus rezultatus į egzaminų rezultatų failą.

## Veikimo aprašymas

Studentai laiko egzaminus internetiniame puslapyje. Atsakius į klausimus, studentas pateikia savo atsakymus į sistemą. Sistema sugeneruoja failus, su kiekvieno studento atsakymais, *json* formatu. Failai atrodo taip:

```
{
  "studentas":{
    "id":"1256",
    "vardas":"Vardenis",
    "pavarde":"Pavardenis"
  },
  "egzaminas":{
    "id":"124",
    "pavadinimas":"OOP pagrindai",
    "tipas":"testas"
  },
  "atsakymai":[
    {
      "klausimas":1,
      "atsakymas":"c"
    },
    {
      "klausimas":2,
      "atsakymas":"a"
    },
    {
      "klausimas":3,
      "atsakymas":"d"
    },
    {
      "klausimas":4,
      "atsakymas":"c"
    }
  ]
}
```

Po kiekvieno egzamino sistema sudeda visus atsakymus į egzamino katalogą.

Mums reikalinga programėlė, kuri ištaisyti studentų egzaminus pagal nurodytą teisingų atsakymų failą. Programėle įvertina kiekvieno studento atsakymus, naudodamasi teisingų atsakymų failu. Gauti studentų įvertinimai surašomi į egzaminų rezultatų failą.

Teisingų atsakymų failas galėtų atrodyti taip:

```
{
  "egzaminas":{
    "id":"124",
```

```

    "pavadinimas":"OOP pagrindai",
    "tipas":"testas"
  },
  "atsakymai":[
    {
      "klausimas":1,
      "atsakymas":"a"
    },
    {
      "klausimas":2,
      "atsakymas":"a"
    },
    {
      "klausimas":3,
      "atsakymas":"d"
    },
    {
      "klausimas":4,
      "atsakymas":"c"
    }
  ]
}

```

Rezultatų failas galėtų atrodyti taip:

```

{
  "egzaminai":[
    {
      "id":"1",
      "pavadinimas":"OOP pagrindai",
      "studentųRezultatai":[
        {
          "id":"11",
          "vardas":"Vardenis",
          "pavarde":"Pavardenis",
          "ivertinimas":7
        }
      ]
    }
  ]
}

```

Įsivaizduokite, kad gali būti skirtingi egzaminų tipai, su skirtingomis atsakymų struktūromis, tačiau pirmą įgyvendinkite sistemą veikiančią tik su vienu tipu – testu. Jei turėsite laiko pridėkite papildomus tipus kaip pavyzdžiui – kelių atsakymų testas, matematinių uždavinių testas ir t.t.

Pavyzdžiuose duomenys laikomi *json* formato failuose, tačiau jei šis formatas jums netinka, galite naudoti kokį kitą formatą - egzistuojantį (pvz. xml), arba sugalvotą jūsų.

Failų struktūros yra tik pavyzdžiai ir gali būti pakeistos pagal jūsų poreikius.

## Funkciniai reikalavimai

- Programa vykdoma iš komandinės eilutės
- Programa priima du argumentus – kelią iki egzamino katalogo ir kelią iki atsakymų failo.  
Pavyzdžiui: `java exams C:\exams\oop-basics`

C:\exams\answers\oop\_basics\_answer.json

- Rezultatų failas yra sukuriamas jei jis neegzistuoja
- Teisingai įvykdžius programą rezultatų failas papildomas naujais studentų rezultatais
- Jau įrašyto egzamino rezultatai nėra rašomi į failą antrą kartą
- Įvykus klaidai, matomos aiškos klaidos žinutės

## Projektų techniniai reikalavimai

---

- Panaudoti *Generics*
- Panaudoti *Collections* (sąrašai, mapai ir pan.)
- Programos kodas turi būti padengtas bent 10 *Unit* testų
- Programos supakavimui panaudoti *Maven*, o programą turi būti galima paleisti naudojant komandinę eilutę
- Panaudoti sąsajas ir paveldėjimus, bei metodų perrašymus (*override*), perkrovimus (*overload*)
- Panaudoti *Java Date Time API*
- Panaudoti išimčių (*Exceptions*) mechanizmą
- Panaudoti *Java IO API* (pvz. failai)
- Panaudoti daugiagijįškumą (*Multi threading*)
- Panaudoti *lambda* ekspresijas ir srautus (*streams*)
- Panaudoti *Enum*
- Programos kodas turi būti tvarkingas, su aiškiais kintamaisiais ir lengvai skaitomas, naudojamos objektiškai orientuoto programavimo praktikos