

JAVA programavimo kalba

Java URL ir serialize

Objektu serializavimas (serialize)

- Tai toks veiksmas, kai java objektas paduodamas į binarinį srautą, kuris tą objektą paverčia tiesiog baitų seka.
- Galimas ir atvirkščias reiškinys - objekto atkūrimas pagal duomenis gaunamus iš srauto.
- Objektas deserializuojamas į tą pačią būseną kokia buvo objekto serializavimo metu, t.y. atstatomos visų serializuotų jo laukų reikšmės.
- Norint padaryti objektą serializuojamu užtenka parodyti, kad jis realizuoja **Serializable** sąsają

Objektu serializavimas (serialize)

- Kai kurių klasių objektai negali būti serializuoti, pvz klasių: `OutputStream`, `Thread`, `Socket` objektai
- Jei serializuojamas objektas tiesiogiai arba netiesiogiai (per kitus objektus) turi neserializuojamo tipo laukus, tai juos reikia pažymėti **`transient`** modifikatoriumi - tai reišk, kad duoto lauko nereikia serializuoti arba deserializuoti.
- Jei bandysime serializuoti objektą kurio negalima, tai programa išmes **`NotSerializableException`**

Objektu serializavimas (serialize)

- Objekto deserializavimo metu java suskaičiuoja duotos klasės hešą (serialVersionUID) pagal visus klasės laukų, metodų vardus, tipus, priėjimo modifikatorius ir t.t. ir tą reikšmę palygina su deserializuojamos klasės. Jei tos reikšmės nesutampa - išmetama klaida: **InvalidClassException - local class incompatible ...**
- Reikia klasėje aprašyti serialVersionUID lauką su atitinkama reikšme:
- `private static final long serialVersionUID = 123...;`
- **SerialVersionUID** reikšmė serializuojamoje klasėje paprastai yra sugeneruojama kaip atsitiktinis skaičius.

Objektu serializavimas (serialize)

- Jei ruošiamės perduoti serializuotus objektus per nesaugią (internetas) aplinką, tai žinotina, kad standartinės objektų serializavimo procedūros sugeneruotas objekto binarinis vaizdas nėra užkoduotas ir todėl bet kas gali pilnai atkurti objektą (jo laukų reikšmės)
- Galima sukurti savo metodus toje klasėje, kurie būtų atsakingi už **serializavimą/deserializavimą**:
 - private void **writeObject**(java.io.ObjectOutputStream out) throws IOException
 - private void **readObject**(java.io.ObjectInputStream in) throws IOException, ClassNotFoundException

URL

- Norint pasiekti kažkokius resursus esančius internete, reikia visų pirma žinoti jų adresą. Tam skirta URL klasė.
- URL - (Uniform Resource Locator) gali rodyti ne tik į paprastą failą gulintį kažkur internetiniame serveryje, bet gali reikšti ir įvairias užklausas (Duomenų bazės, API ir pan.)

`https://en.wikipedia.org:443/wiki/Regular_expression`

- `https` - protokolas
- `en.wikipedia.org` - mazgo / serverio adresas
- `443` - portas
- `/wiki/Regular_expression` - failas

URL/URI

- URL klasė nedaro URL simbolių kodavimo/dekodavimo, t.y. ji pvz nepaverčia užklausoje ' ' (tarpo) į %20 ir todėl du URL:

`http://example.com/hello world/`

`http://example.com/hello%20world`

laikomi nelygiais.

- Tokius pavertimus automatiškai daro klasės URI objektas.
- Iš URI objekto lengva gauti URL objektą ir atv.
- URI paskirtis - palengvinti darbą su internetiniais adresais

URI

Bendra URI struktūra:

[scheme:][//authority][path][?query][#fragment]

kur:

scheme: http, https, ftp, ...

authority: [*user-info*@]*host*[:*port*]

Internetinės užklausos – skaitymas / rasymas

- **URL** objektas turi metodą **openStream()**, kuris grąžina **InputStream** objektą.
- Jei norime kažką nusiųsti, tai reikia naudoti **URLConnection** arba **HttpURLConnection** klases, kurių objektą galima gauti iš **URL** su **openConnection()** metodu.

GSON

- Biblioteka skirta darbui su JSON tipu. Pavercia java objekta i JSON formata ir atvirksciai, is JSON (String) reiksmes pavercia i JAVA objekta.
- Metodai:
 - toJson()
 - fromJson()

Uždaviniai

1. Parašykite funkciją, kuri pagal duotą adresą Lietuvoje, grąžintų jo indeksą. Patikrinkite funkcijos veikimą su keliais adresais. JSON atsakymo parsinimui pabandykit panaudoti kokias nors bibliotekas, kaip kad **Gson**.
2. Papildykite kavinės aptarnavimo sistemą. Padarykite taip, kad galėtume išsaugoti duomenis faile ir po to perskaityti iš failo duomenis JSON formatu (galima naudoti GSON biblioteka). Failas kas kartą papildomas ir tik pirmą kartą automatiškai sukuriamas. Faile turėtų būti išsaugoma:
 1. Data, kada buvo aptarnautas staliukas ir kas jį aptarnavo. Pvz (kaip bus parasyta nesvarbu, svarbu, kad butu JSON formatu ir butu staliuko nr, data, suma):
 - Staliukas Nr.1 buvo aptarnautas 2018-05-24 08:00 už sumą 20eur. Eurorija Eurora.
 - Staliukas Nr.1 buvo aptarnautas 2018-05-24 12:25 už sumą 25eur. Eurorija Eurora.
 - Staliukas Nr.1 buvo aptarnautas 2018-05-24 18:45 už sumą 35eur. Eurorija Eurora.
 - Staliukas Nr.2 buvo aptarnautas 2018-05-28 08:00 už sumą 20eur. Eurorija Eurora.
 - Staliukas Nr.2 buvo aptarnautas 2018-05-24 08:00 už sumą 20eur. Eurorija Eurora. Ir t.t
 2. Padarykite taip, kad datos skirtusi (galima is console perskaityti ir irasyti i faila kiekvieną kartą vis iš naujo paliedus programą padidinti dieną ar metus ar menesį).
 3. Paskaičiuokite ketvirčiais kiekvieno staliuko sumą. Pvz (1 ketvirtis – 1 staliukas – suma 100)
 4. Sukurkite galimybę buhalterui įvesti staliuko nr. ir datą (be laiko) , pagal pateiktus duomenis suraskite tokį įrašą/įrašus faile ir grąžinkite tik tą vieną įrašą. Pvz: paieška ir grąžinam surastą įrašą/įrašus pagal staliuko nr. Ir pagal datą. Pvz: įvedam Nr.1 2018-05-24 , tai grąžins 3 įrašus (Nr1 suma 20, Nr1 suma 25 , Nr1 suma 35), visus įrašus atitinkamai padarykite kaip objektus. Tam panaudojam GSON biblioteką.
3. Patobulinkite slaptažodžio atpažinimo sistemą. Kai registruojasi vartotojas, įrašykite j failą duomenis. Kai vartotojas atnaujinamas (keičiamas userName arba password) , suraskite faile tą vartotoją ir atnaujinkite duomenis. Kai adminas ištrina vartotoją, suraskite faile įrašą ir ištrinkite. Visais atvejais failo turinys keičiasi, bet kiekvieną kartą įrašius pridedami įrašai prie esančių, o ne iš naujo sukuriamas failas ir perrašomas.