Statistička analiza planinskih vrhova (1.2)

1. Opis problema:

Potrebno je analizirati podatke o planinskim vrhovima koji se mogu da se nalaze zapisani u fajlovima različitih formata (.properties, .csv) i snimiti rezultat obrade u izlazni .html fajl.

2. Arhitektura implementacije I

Rešenje treba da bude implementirano u vidu posebne biblioteke, čiji će API da ima sledeći prototip:

class MountainAnalyzer

public void analyze(List<File> inputFiles, OutputStream report)

Napomena: klasu implementirati kao *Singleton*

2.1.Ulazni fajlovi

Ulazni fajlovi sadrze podatke o imenima planina kao i spisak svih vrhova odgovarajuće planine zajedno sa osnovnih podacima o vrhovima (ime vrha i nadmorska visina vrha u metrima).

2.1.1. CSV (*Comma Separated Values*) fajl

Svaki red je identicnog formata:

Ime planine DELIMITER ime vrha 1 DELIMITER visina vrha 1 DELIMITER (... DELIMITER) ime vrha N DELIMITER visina vrha N KRAJ_REDA

pri čemu je DELIMITER znak **,** a KRAJ_REDA je CR ili CRLF

Smatrati da se jedna planina pojavljuje samo u jedom redu.

Napraviti jedan primer fajla koji ispunjava navedene kriterijume sa najmanje 5 redova.

2.1.2. *Properties* fajl

Ime ključa se sastoji od imena planine i imena vrha, koji su razdvojeni tackom pri čemu nisu dozvoljeni razmaci. Vrednost je visina vrha u metrima. Primer

mountain1.hill1=height_in_meters

mountain1.hill2= height_in_meters

mountain1.hill3= height_in_meters

mountain2.hill1=height_in_meters

mountain3.hill2= height_in_meters

mountain3.hill3= height_in_meters

Napraviti jedan primer fajla koji ispunjava navedene kriterijume sa najmanje 5 redova.

2.2.Parseri

Svaki podržani fajl format ima odgovarajuću implementaciju *FileParser* interfejsa:

interface FileParser

ParseResult parse (File f)

Dizajnirati i implementirati klasu *ParseResult* tako da sadrži **isključivo** sve potrebne informacije iz parsiranog fajla. Sprečiti **bilo kakvu izmenu** jednom postavljenih podataka u klasi.

Instanciranje odgovarajućeh parsera se vrći pomoću klase ParseFactory (*Factory Design Pattern)* koja vraća odgovarajući parser na osnovu ekstenzije fajla. Smatrati da *csv* i *properties* fajlovi imaju ekstenzije .csv i .properties respektivno. U slučaju da format nije podržan vratiti *null* vrednost.

2.3. Obrada rezultata

Napraviti posebnu klasu za obradu rezultata parsiranja tako da se dobiju sledeći rezultati:

a) Za svaku planinu pojedinačno izračunati: ukupan broj vrhova, prosečnu visinu vrhova, medijanu vrhova, najniži vrh i njegovu visinu, najviši vrh i njegovu visinu.
b) Naći planinu sa najvišim vrhom
c) Naći planinu sa najviše vrhova
d) Naći planinu sa najvećom prosečnom visinom vrhova

2.4. Čuvanje rezultata

Rezultat upisati u strim koji je parametar *MountainAnalyzer.analyze()* metode i to u html formatu, tako da se:

a) Svi podaci dobijeni u Obrada rezultata a) smeste u jednu zajedničku tabelu u kojoj svaka vrsta sadrži podatke o jedoj planini pri čemu su planine poređane u **ISTOM** redosledu kao i u ulaznim fajlovima.
b) Podaci dobije u Obrada rezultata b),c),d) se smeštaju u posebne paragrafe sa odgovarajućim tekstom
c) Ukoliko je došlo do bilo kakve greške (loš fajl, loš sadržaj i sl...), i one se ispisuju u posebnom paragrafu

Html fajl treba da bude najjednostavniji mogući – nema potrebe za CSS-om i sličnim ulepšavanjima.

2.5. Ostale napomene

- Pojava greške pri obradi jednog fajla, ne sme uzrokovati prekid obrade ostalih fajlova. Grešku ispisati pri čuvanju rezultata.
- Rezultate obraditi zbirno, a ne pojedinačno po fajlovima
- Sve klase mogu da se nalaze u jednom java projektu
- Napraviti klasu za testiranje klase **MountainAnalyzer**, koja će koristite napravljene test fajlove iz tačaka CSV (Comma Separated Values) fajl i Properties fajl
- Postoji mogućnost da će se u nekoj narednoj izmeni zahtevati podrška za dodatne formate čuvanja rezultata
- Polja klasa postaviti da budu privatni i pristupati im **isključivo** preko getera i setera.