

Programski jezici 2 - 2246

24.08.2017.

1. **(30)** Napisati sistem za automatizovanu obradu sadržaja izmišljene kompanije. Osnovni dokumenti sa kojima kompanija radi dobijaju se od nepoznatih spoljašnjih sistema u formatima koji se nalaze na Moodle stranici predmeta. Svi dokumenti se ručno smještaju u folder *PJ2_exam_data*. Aplikacija koju je potrebno razviti sastoji se iz nekoliko modula. *Importer* je modul koji čita sadržaj foldera sa dokumentima i uvozi podatke u sistem. Podaci se prate na 2 načina: ručno (pri čemu se šalje naziv fajla) i automatski. Automatski način praćenja podrazumijeva da aplikacija prati u realnom vremenu da li je u folder dodan neki fajl. Čim se detektuje dodavanje fajla bilo kojom od 2 moguće metode, počinje se sa uvozom podataka. Za uvoz podataka potrebno je kreirati odgovarajuće adapter klase, abstraktnu i konkretne za svaki format podataka i pozivati ih na optimalan način da se obezbijedi proširivost formata podataka. Parsiranje, odnosno obrada podataka obavlja se pozivom metode *import* iz odgovarajućeg konkretnog adaptera. Po potrebi koristiti generičke mehanizme za implementaciju. Adapteri su zaduženi da izvrše parsiranje sadržaja fajlova i kreiraju odgovarajuće objekte. Objekti koji se mogu napraviti su instance klasa: *Proizvod* (*sifra*, *naziv*, *količina*, *cijena*), *Proizvođač* (*naziv*) i *Vrsta* (*naziv*). Na početku svakog naziva fajla nalazi se broj proizvoda u fajlu. *Importer* pravi objekte na osnovu podataka u fajlu, a ako ukupan broj instanciranih objekata nije isti kao broj proizvoda koji se nalazi u nazivu fajla, tada se baca *InvalidPJ2FileException*. Ukoliko nedostaje bilo koji od navedenih atributa objekata baca se *InvalidPJ2DataException*, objekat se ne dodaje u sistem, ali se proces uvoza podataka ne prekida. Uvoz podataka se obavlja pojedinačnim nitima tako da se obezbijedi konkurentan uvoz podataka. Svi objekti se smještaju u odgovarajuće kolekcije nad kojim se zatim upotrebom *Stream* API-ja obavljaju sljedeće operacije:

- izdvajanje proizvoda po nazivu (parametar) i
- izdvajanje proizvoda čija je cijena veća od 100.

Nakon što se podaci uvezu u sistem, kolekcije je potrebno serijalizovati u foldere koji imaju naziv kao i originalni fajl (bez ekstenzije fajla). U sva tri zadatka koristiti parametre, imenovane konstante, enumeracije, generičke mehanizme i sl., gdje god je to moguće.

2. **(20)** *Processor* je drugi modul sistema koji ima zadatak da obradi uvezene podatke. Ovaj modul se pokreće periodično svakih 30 sekundi i obrađuje podatke samo jednom. Oznaku da li su neki podaci obrađeni (folderi) dodati u naziv foldera i na taj način izbjegavati ponovnu obradu. Foldere koje je potrebno obraditi izlistati upotrebom *FilenameFilter* iz *File* API-ja. Obrada podataka se obavlja na sljedeći način:

- obavi se deserijalizacija podataka,
- svaka cijena se uveća za 20%,
- ponovo se serijalizuju liste u iste fajlove iz kojih su deserijalizovane.

Obrada podataka se obavlja konkurentno, a nakon što se svi podaci obrade *Processor* završava rad i čeka novo pokretanje. Koristiti *join* metodu.

3. **(20)** Treći modul sistema je *Exporter*. Ovaj modul se pokreće ručno, nakon što *Processor* završi obradu podataka. Modulu se kao argumenti komandne linije proslijede putanja do serijalizovane liste koja se izvozi, i format za izvoz (*java Exporter -path putanja -format csv*). Dostupni formati su CSV i TXT. Izvezeni podaci se čuvaju u *Home* direktorijumu korisnika računara.

Napomena: Sve zadatke predavati odvojeno inače neće biti pregledani! Vrijeme rada: 180 minuta

Programski jezici 2 – A401

24.08.2017.

1. (30) Napisati klijent server aplikaciju za evidenciju turističkih mjesta, koja imaju naziv i turističke objekte. Svi turistički objekti imaju naziv. Turistički objekti mogu biti hoteli, moteli, restorani i kafići. Hoteli i moteli imaju broj soba a restorani broj mjesta. U hotelima i motelima turisti mogu da prenoće, a u kafićima i restoranima turisti mogu da konzumiraju proizvode. Hoteli, moteli i restorani primaju rezervacije. Klijentska aplikacija treba da prikaže sve turističke objekte u nekom mjestu, slanjem poruke INFO#naziv mjesta na server. Opcija OBJ#vrsta#mjesto služi da dobijanje svih turističkih objekata određene vrste u mjestu. Opcija SORT služi da dobijanje spiska turističkih objekata sortiranih po nazivu u opadajućem redoslijedu, a opcija MAX prikazuje hotel sa najvećim brojem soba u svim mjestima. Prije pokretanja servera potrebno je kreirati 3 mjesta sa proizvoljnim brojem (najmanje 2) turističkih objekata različitih vrsta. Kreirani objekti se serijalizuju na disk. Implementirati metodu na serveru koja prima naziv hotela a vraća deserijalizovani objekat.

2. (15) Napisati RMI aplikaciju koja omogućava pregled i obradu fajlova. Klijent unosi putanju do direktorijuma, nakon čega se prikazuje sadržaj tog direktorijuma i svih njegovih poddirektorijuma. Korisnik može da unese opciju za sortiranje fajlova, pri čemu se iz određenog direktorijuma izlistaju fajlovi i pošalju na serverski dio aplikacije gdje se sortiraju po veličini u opadajućem redoslijedu. Nakon toga se sortirani podaci prikazuju korisniku i upisuju u fajl na serveru sort.txt u formatu naziv_fajla - veličina u KB, ukoliko to korisnik želi. Upis podataka u fajl realizovati kao posebnu metodu.

3. (25) Napisati višenitnu aplikaciju za slanje pošiljki u gradu Java. Pošiljka ima broj pošiljaoca, broj primaoca i sadržaj generičkog tipa. Grad Java ima 10 polja, koja imaju redni broj i sadržaj proizvoljnog tipa. Tri poštara prolaze gradom različitim brzinama: 1 s, 2 s, 3 s po polju. Brzina je vrijeme koje se poštara zadrži na polju prije prelaska na sljedeće polje. Kada poštara dođe na neko polje, ukoliko ne prenosi pošiljku, kreiraće pošiljku sa sadržajem koji je u polju, a za odredište postavlja slučajno generisan broj veći od broja polja na kome se trenutno nalazi. Nakon toga, poštara odlazi do odredišnog polja gdje predaje pošiljku. Kretanje poštara se nastavlja na isti način do kraja grada. Simulacija se završava kada svi poštari dođu do kraja grada, i tada se prikazuje ukupan broj prenesenih pošiljki.

Vrijeme rada: 180 minuta