

Laboratorijska vježba 7 – IO

1. Kreirati klasu AnalizaProdaje koja u main metodi treba da obradi podatke o prodaji u JavaTechnoShop-u. Podaci o prodaji se čuvaju u fajlu *prodaja.txt*. Podaci koji se čuvaju imaju format kao na slici.

```
prodaja.txt - Notepad
File Edit Format View Help
monitor#lg234#3#400
telefon#samsung galaxy s4#1#899
telefon#htc one#2#500
racunar#lenovo#4#1205
racunar#hp#2#900
monitor#samsung#1#400
racunar#lenovo#1#1200
```

Prvi parametar je kategorija proizvoda (monitor, telefon, racunar), drugi naziv proizvoda, treći je broj prodanih proizvoda a četvrti cijena jednog proizvoda. Isti proizvodi se mogu ponavljati u listi. Potrebno je napraviti tri fajla (*prodajaRacunari.txt*, *prodajaMonitori.txt* i *prodajaTelefoni.txt*) u kojima će se nalaziti proizvodi iz fajla *prodaja.txt*, razvrstani prema kategoriji proizvoda. U fajl *prodajaSumarno.txt* upisati izvještaj o ukupnoj vrijednosti prodane robe za svaku kategoriju (suma proizvoda cijene i broja prodanih proizvoda).

```
prodajaSumarno.txt - Notepad
File Edit Format View Help
Monitori 1600.0
Racunari 7820.0
Telefoni 1899.0
```

2. Modifikovati zadatak JavaTechnoShop (laboratorijska vježba 6) tako da se obezbijedi trajno čuvanje izdanih računa. Račune je potrebno serijalizovati u folderu *racuni*, a svaki serijalizovani objekat treba da bude u fajlu sa imenom u formatu *dan.mjesec.godina_sat_minute_sekunde.etf*. U meni dodati opciju za prikaz svih serijalizovanih računa i prikaz ukupnog prihoda sa svih računa.
3. Napisati pojednostavljen program za rad sa fajl sistemom. Metode koje je potrebno implementirati su date u tabeli:

Metoda	Opis
void changeDir(String name)	Prelazi u novi direktorijum sa nazivom koji se dobija kao argument. Ako je argument name jednak ".." prelazi se u direktorijum iznad trenutnog. Trenutnu poziciju čuvati u atributu <i>File currentPath</i> .
void listFiles()	Prikazuje sadržaj trenutnog direktorijuma. Ukoliko direktorijum sadrži druge direktorijume oni se prikazuju u formatu <DIR> naziv
void createDirectory(String name)	Kreira direktorijum u tekućem direktorijumu
void delete(String name)	Briše fajl ili direktorijum u tekućem direktorijumu
void copyImageFile(String sourceName, String destinationName)	Kopira jpg slike u istom direktorijumu (Koristiti <i>FileInputStream</i> i <i>FileOutputStream</i> klase)
void listWordDoc()	Prikazuje sve Word fajlove u tekućem direktorijumu

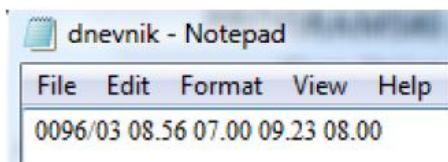
U main metodi omogućiti upotrebu svih implementiranih metoda. Opcije koje program treba da podrži su:

md name == kreiranje foldera sa proslijedjenim imenom
dir == prikaz sadrzaja direktorijuma
cd name == otvaranje drugog direktorijuma
del name == brisanje fajla ili direktorijuma
copy source destination == kopiranje jpg slika u istom folderu
word == prikaz svih word datoteka
quit == izlaz

Početni direktorijum treba da se dobije opcijom *System.getProperty("user.home")*.

Zadaci za samostalan rad:

4. Pod pretpostavkom da se u fajlu *dnevnik.txt* nalaze prosječne ocjene svakog studenta za svaku godinu studija, u formatu prikazanom na slici, omogućiti upis prosječne ocjene na kraju završetka fakulteta u njegov fajl, dodavajući u sam fajl liniju: Uspjeh u toku studija – X.Y (za pronalazak odgovarajućeg fajla koristiti broj indeksa, koji je dat u prvoj koloni u fajlu na slici, dok preostale 4 kolone predstavljaju prosječnu ocjenu za godine studija od prve do četvrte; prosječnu ocjenu na kraju studija potrebno je izračunati na osnovu prosječnih ocjena po godini studija; kolone su razdvojene jednim space-om, a prosječne ocjene su u formatu XX.ZZ).



5. Za kompaniju **JavaWorld** potrebno je razviti aplikaciju koja će omogućiti praćenje vremena koje jedan korisnik provede u radu za računarom. Zaposlenog treba predstaviti kao objekat, koji ima ime, prezime, korisničko ime i broj sati koje je proveo za računarom. Aplikacija treba da bude konzolna, i da vrši sljedeće funkcije:

- Pokreće se zaposleni kad sjeda za računar prvi put, i kreira se objekat tipa **Zaposleni**, pomoću interaktivnog unosa. Nakon toga počinje da mu se mjeri vrijeme koliko je za računom (prilikom svakog pokretanja potrebno je ispisati koji su korisnici trenutno aktivni, pa unosom korisničkog imena, ukoliko se korisnik vratio, učitati njegov objekat i dalje raditi sa njim).
- Ukoliko korisnik unese u nekom trenutku u konzolu riječ PAUZA, aplikacija se gasi, i čeka da se korisnik ponovo pojavi (objekat **Zaposleni** potrebno je serijalizovati).
- Kada se korisnik ponovo pojavi, vrijeme mu se nastavlja obračunavati.
- Unosom ključne riječi KRAJ u nekom trenutku, u datoteku *korisniklme.txt* upisati podatke o korisniku i provedeno vrijeme za računom.