

**1. (30)** *Virtuelna pošta* vrši raspodjelu i slanje pošiljki. Svaku pošiljku karakteriše adresa primaoca, adresa pošiljaoca i težina. Pošiljke se mogu podijeliti na razglednice, pisma i vrijednosne pošiljke. Razglednice imaju sadržaj i vizuelni dio, koji može da bude fajl sa ekstenzijom .jpg, .jpeg ili .png, a njihova težina je slučajan broj u opsegu od 1 do 10. Pismo ima sadržaj koji se nalazi u tekstualnom fajlu, a njegova težina je slučajan broj u opsegu od 1 do 20. Vrijednosna pošiljka ima vrijednost izraženu u jednoj od valuta: evro, konvertibilna marka ili američki dolar, a njihova težina je slučajan broj u opsegu od 10 do 100. Pošiljke je moguće spakovati u paket. Paket je generička klasa, koja u sebi sadrži listu pošiljki. Paket ima težinu, koja se računa kao zbir težina svih pošiljki u paketu. Pod slanjem pošiljki podrazumijeva se prikaz svih informacija (sadržaja metode *toString*) o pošiljki na konzoli. Ukoliko je težina pošiljke veća od 15, pošiljka se dijeli na tri dijela i šalje se dio po dio sa pauzom između slanja dijelova od 3 sekunde. Ista analogija primjenjuje se i na pakete.

Simulacija rada virtuelne pošte odvija se na sljedeći način:

- kreira se po 15 pošiljki svakog tipa i smještaju se u proizvoljno odabranu kolekciju,
- nad dobijenom kolekcijom odradi se *shuffle*,
- novodobijena kolekcija se proslijeđuje niti, koja vrši razvrstavanje dobijenih pošiljki u tri kolekcije, po tipu i serijalizuje ove kolekcije,
- nakon razvrstavanja, kolekcije se proslijeđuju drugoj niti, koja iz svake kolekcije uzima po pet slučajno odabranih pošiljki i pakuje ih u tri paketa (paketi se uklanjaju iz kolekcija),
- dobijeni paketi i kolekcije proslijeđuju se trećoj niti, koji obavlja slanje,
- simulacija završava kada se završi slanje,
- na kraju se ispisuje vrijeme trajanja simulacije, putanja do serijalizovanih kolekcija i veličine serijalizovanih fajlova.

**2. (20)** Napisati konzolnu aplikaciju u kojoj se putem argumenata komandne linije navodi putanja do foldera u kojem se vrši pretraga svih tekstualnih datoteka koje se u istom nalaze (važi i za foldere u sklopu tog foldera itd...). Tekst za pretragu se unosi nakon pokretanja aplikacije, pri čemu se od korisnika traži da se isti unese putem konzole. Pretraga se vrši na nivou kompletnog teksta, pri čemu se ne uzima u obzir razlika između malih i velikih slova (sve kombinacije se trebaju detektovati kao rezultat pretrage tj. pretraga nije *case sensitive*). Pored teksta u datoteci potrebone je pretražiti i naziv datoteke. Ukoliko je u nekoj tekstualnoj datoteci pronađena jedna ili više traženih sekvenci potrebno je ispisati putanju do datoteke, broj reda u tekstualnoj datoteci gdje se sekvenca nalazi i redni broj karaktera u redu od kojeg počinje sekvenca (drugo i treće nije potrebno izpisati ako je u pitanju naziv datoteke). Pored toga, koristeći paket *java.awt* i klasu *Desktop*, otvoriti prozor sa sadržajem foldera u kojem se nalazi detektovana tekstualna datoteka. Pretraga se nastavlja tek kada korisnik na konzolu unese komandu

za nastavak pretrage (komanda neka je po slobodnom odabiru tj. neki karakter, ali bitno je da se prije toga ispiše na konzolu koji se karakter očekuje).

**3. (20)** Napisati generičku metodu koja kao argumente prima listu *Predicate* objekata koji su uslovi koji će biti testirani nad podacima, dva cjelobrojna argumenta i neodređen broj listi podataka (više njih) koje implementiraju interfejs *Podatak*. Cjelobrojni argumenti predstavljaju početni i krajnji indeks u listi rezultata (krajnji indeks može biti jednak broju elemenata ili manji), koja se dobija tako što se pomoću *Predicate* uslova iz svake liste uzimaju samo oni elementi koji ispunjavaju sve proslijeđene uslove. Lista rezultata je sortirana od elementa koji ima najveću *hashCode* vrijednost ka objektima sa manjim vrijednostima. U *main* metodi potrebno je napraviti 3 liste objekata klase *Student* (ime, prezime, brojIndeksa) i testirati generičku metodu.

**Vrijeme za rad: 180 minuta**