

Zadaća 2 – Dostava paketa po područjima

Naziv projekta: {LDAP_korisničko_ime}_zadaca_3

Ishodišni direktorij projekta: {LDAP_korisničko_ime}_zadaca_3

Naziv rješenja: {LDAP_korisničko_ime}_zadaca_3.zip

Prije predavanja projekta potrebno je napraviti Clean na projektu (obrisati sve pomoćne i izvršne datoteke kao što su .class, .jar, .). Zatim cijeli projekt sažeti u .zip (NE .rar) format s nazivom {LDAP_korisničko_ime}_zadaca_3.zip i predati u Moodle. Uključiti izvorni kod, popunjenu datoteku dokumentacije (u ishodišnom direktoriju projekta). Potrebno je podesiti konfiguraciju projekta (maven) tako da se kreira Java izvršna .jar datoteka.

Uvod

Tvrtka obavlja dostavu različitih vrsta paketa uz određene usluge dostave. Paket može biti tipski koji ima predefinirane dimenzije (to su tipski paketi s oznakama A, B, C, D, E) i slobodnim dimenzijama (oznaka X, postoji maksimalna visina, širina i dužina). Za svaki tipski paket postoji maksimalna težina, a kod paketa X slobodnih dimenzija maksimalna težina određena je parametrom (mt) prilikom izvršavanja programa. Usluge dostave mogu biti standardna (S), hitna (H), plaćanje pouzećem (P), povratak paketa (R).

Tvrtka ima radno vrijeme koje je određeno parametrima (pr za početak rada i kr za kraj rad) prilikom izvršavanja programa. Tvrtka se sastoji od ureda za prijem paketa od strane pošiljatelja i ureda za dostavu paketa na adresu primatelja. Ured za prijem paketa evidentira za svaki paket vrijeme prijema paketa, pošiljatelja, primatelja, vrstu paketa (kod X i ostale podatke), vrstu usluge (iznos kod plaćanja pouzećem). Podaci za pakete na prijemu se kronološki evidentiraju. Za svaki primljeni paket potrebno je naplatiti iznos dostave (svaki tipski paket ima svoju cijenu dostave, a za paket X slobodnih dimenzija formula se nalazi u tablici 1), osim kod paketa s uslugom plaćanja pouzećem. Njega naplaćuje vozilo koje odrađuje isporuku. Kod prijema paketa sustav obavještava pošiljatelja i primatelja paketa da je paket zaprimljen. Obavješćavanje o promjeni statusa paketa mora se temeljiti na uzorku dizajna Observer. Svi pošiljatelji i primatelji paketa na početku imaju pridruženo obavješćavanje za svaki paket. Može se komandom promijeniti.

Ured za dostavu paketa raspolaže s određenim voznim parkom (od bicikla, skutera do automobila). Svako vozilo ima kapacitet prijevoza (težina u kg, prostor u m³). Prostor koji pokriva ured podijeljen je u više područja. Jedno područje može pokrivati jedno mjesto/grad ili više njih. Jedno mjesto ima jednu ili više ulica. Jedno područje može sadržavati jedno ili više mjesta pri čemu u područje ulaze sve ulice pojedinog mjesta/građa ili samo dio njih. Npr. može biti situacija da jedno područje pokriva jedno mjesto i samo neke njegove ulice. Ili jedno područje pokriva dva mjesta sa svim njihovim ulicama i dio ulica trećeg mjesta. Struktura područja mora se temeljiti na uzorku dizajna Composite.

Za svako vozilo određena su područja koja pokriva i to prema njihovom rangu. Vozila mogu biti aktivna, neispravna ili neaktivna. Samo aktivna vozila mogu sudjelovati u ukrcavanju paketa. Vozilo koje je ili je postalo neispravno ne može se koristiti do kraja rada programa ali sudjeluje u ispisima koji se odnose na vozila. Vozilo koje je postalo neaktivno ne može obavljati

svuju funkciju dok mu se ne promijeni status u aktivno. Rad s vozilima mora se temeljiti na uzorku dizajna State. Rad jednog vozila temelji se na ukrcavanju paketa, isporuci paketa i povratku u ured. To čini jednu vožnju. Vozilo može obaviti više vožnji tijekom radnog dana. Ukrcavanje paketa može se obaviti samo unutar radnog vremena. Isporuca paketa može se obaviti i nakon radnog vremena tako dugo dok se ne isporuče svi paketi.

Pakete koje treba dostaviti ukrcavaju se u slobodno vozilo tako da se ne prekorači njegova dozvoljena težina i prostor. Na kraju svakog punog sata odlučuje se koji paketi se mogu ukrcati u pojedino vozilo temeljem adrese paketa (a time i područja kojem pripada) i područja koje je pridruženo vozilu. Adresa paketa dobije se na temelju primatelja paketa, njegovog grada, ulice i kućnog broja. Vozilo u jednoj vožnji može prevoziti pakete samo iz jednog područja (određuje se na temelju prvog paketa koji se ukrca). Prvo se ukrcavaju paketi koji imaju hitnu dostavu. Ako postoje paketi s hitnom dostavom za svakog od njih traži se vozilo koje je već pridruženo za područje tog paketa s hitnom dostavom (prethodni paket s hitnom dostavom) ili ispravno slobodno vozilo koje je najviše rangirano za područje paketa s hitnom dostavom, uz uvjet da ima raspoloživu težinu i prostor za paket. Ako se pronađe raspoloživo vozilo tada će ono u toj vožnji obavljati dostavu za to područje. Ako za pojedini sat trenutno ne postoji raspoloživo vozilo za prijevoz paketa s hitnom dostavom taj će paket pričekati do sljedećeg punog sata. Nakon što su svi mogući paketi s hitnom dostavom pridruženi raspoloživim vozilima slijedi ukrcavanje ostalih paketa. Za ostale pakete vrijedi slična pravila kao i za pakete s hitnom dostavom. Prvo se traži vozilo koje je već pridruženo za područje tog paketa ili ispravno slobodno vozilo koje je najviše rangirano za područje paketa, uz uvjet da ima raspoloživu težinu i prostor za paket. Ako za pojedini sat trenutno ne postoji raspoloživo vozilo za prijevoz paketa taj će paket pričekati do sljedećeg punog sata. Kod ukrcavanja paketa u vozilo sustav obavještava pošiljatelja i primatelja paketa da je paket ukrcao u vozilo. Sada vozila kreću prema odredištima paketa.

Paketi se mogu isporučivati na dva načina (parametar isporuka):

1. Prema redoslijedu kako su ukrcao u vozilo. Za sve pakete u vozilu izračunava se udaljenosti između trenutne GPS koordinate (u prvom segmentu to je GPS ureda, parametar gps) i GPS koordinate za adresu paketa. GPS adresa paketa izračunava se interpolacijom¹ GPS koordinata početka ulice (gps_lat_1 i gps_lon_1) i GPS koordinata kraja ulice (gps_lat_2 i gps_lon_2) te kućnog broj adrese paketa i najvećeg kućnog broja u ulici. Nakon prvog paketa njegova GPS lokacija postaje trenutna lokacija te se izračunava udaljenosti za sljedeći paket. I tako se nastavlja dok se ne obavi za zadnji paket za dostavu.
2. Određivanjem paketa s najbližom dostavom. Za svako vozilo potrebno je odrediti redoslijed isporuka njegovih paketa. Prvo se određuje koji paket ima najbližu dostavu od ureda. Za sve pakete u vozilu izračunava se udaljenosti između

¹ Zbog blizine adresa i jednostavnosti izračuna, karta se promatra kao ravna ploha/ravnina (tako da za izračun nije potrebno uključiti zaobljenost Zemlje.), ulica kao linija između dvije točke, a svi kućni brojevi imaju istu širinu i ravnomjerno su raspoređeni na obje strane ulice. Kućni broj 1 nalazi se na GPS koordinati početka ulice, a najveći kućni broj nalazi se na GPS koordinata kraja ulice. Duljina ulice izračunava se kao udaljenost između dviju točaka u ravnini. Djeljenjem kućnog broja adrese paketa i najvećeg kućnog broja u ulici dobije se % s kojim se množi duljina ulice kako bi se dobila udaljenost adrese paketa od početka ulice. U slučaju da je kućni broj paketa veći od najvećeg kućnog broja u ulici tada se za izračun uzima najveći kućni broj u ulici. Sada je još potrebno tu udaljenost pretvoriti u GPS koordinate. Za to se primjenjuje trigonometrija pravokutnog trokuta.

trenutne GPS koordinate (u prvom segmentu to je GPS ureda, parametar gps) i GPS koordinate za adrese paketa. Kada se odredi paket s najbližom dostavom onda njegova GPS lokacija postaje trenutna lokacija te se traži sljedeći paket s najbližom dostavom. I tako se nastavlja dok se ne utvrdi zadnji paket za dostavu.

Za svaki paket ukupno vrijeme za isporuku sastoji se od vremena vožnje do adrese i vremena postupka isporuke. Vrijeme vožnje izračunava se na temelju potrebnog vremena da se prođe udaljenost od trenutne pozicije do pozicije odredišta paketa uz prosječnu brzinu vozila. Vrijeme postupka isporuke određeno je parametrom (vi u minutama) prilikom izvršavanja programa. Kod isporuke paketa primatelju potrebno je evidentirati vrijeme preuzimanja paketa. Ako je paket s uslugom plaćanja pouzecom potrebno je kod vozila ažurirati prikupljeni novac s iznosom pouzeća. Nakon preuzimanja paketa sustav obavještava pošiljatelja i primatelja paketa da je paket preuzet. Kada se isporuči zadnji paket iz jednog vozila potrebno je vozilo vratiti u ured za što se izračunava vrijeme koje je potrebno da vozilo prođe udaljenost od trenutne pozicije do pozicije ureda uz prosječnu brzinu vozila.

Objekt ureda sadrži objekte svojih vozila. Za svaku vožnju potrebno je kreirati objekt klase koja će sadržavati potrebne podatke o vožnji. Objekt pojedinog vozila sadrži objekte svojih vožnji. Za svaki paket koji se primi u uredu potrebno je kreirati objekt klase koja će sadržavati potrebne podatke. Ured sadrži objekte svojih paketa. Za svaki segment vožnje potrebno je kreirati objekt klase koja će sadržavati potrebne podatke (od GPS, do GPS, udaljenost, vrijeme početka, vrijeme kraja, trajanje vožnje, trajanje isporuke (ako nije povratak u ured), ukupno trajanje segmenta, referenca objekta paketa (ako nije povratak u ured jer tada je null)). Objekt pojedine vožnje sadrži objekte svojih segmenata. **Objekti vozila, vožnji, segmenata i paketa NE mogu se pridruživati (duplirati) drugim kolekcijama osim gore navedenim.**

Sve datoteke su u UTF-8 kodnom zapisu i koriste csv (en. Comma Separated Values) format zapisa u kojem se koristi znak ; za odvajanje vrijednosti pojedinih atributa/stupaca u jednom retku. U svakoj datoteci prvi redak sadrži popis atributa koji se mogu nalaziti u ostalim redcima, on je informativan i preskače se kod učitavanja podataka.

Unutar datoteka svi atributi su obavezni! U cijeloj zadaći potrebno je koristiti hrvatski format datuma (dd.mm.yyyy. hh:mm:ss). Više o datotekama i njihovim atributima možete pronaći u tablici 1.

Tablica 1. Nazivi datoteka, dani primjeri i dodatne informacije o njihovim atributima

Datoteka	Primjer	Dodatne informacije
Podaci vrsta paketa	DZ_3_vrste.csv	Oznaka, opis, visina, širina, dužina, maksimalna težina, cijena, cijena hitno, cijenaP, cijenaT. CijenaP i cijenaT ne koriste se kod tipskih paketa (imaju 0,0). Visina, širina, dužina su u m. Maksimalna težina je u kg. Kod vrste X visina, širina i dužina predstavljaju maksimalne vrijednosti, a maksimalna težina je 0,0 jer se drugačije određuje. Kod vrste X cijena se izračunava na bazi osnovne cijene te joj se pribrajaju: 1. umnožak cijenaP i volumena prostora u m ³ 2. umnožak cijenaT i težine u kg.
Popis vozila	DZ_3_vozila.csv	Registracija, opis, kapacitet težine u kg, kapacitet prostora u m ³ , redoslijed, prosječna brzina, područja po rang, status
Popis prijema paketa	DZ_3_paketi.csv	Oznaka, vrijeme prijema, pošiljatelj, primatelj, vrsta paketa (ako je tipski onda se za sljedeća 3 podatka ne unose vrijednosti tj. imaju 0,0), visina, širina, dužina, težina, usluga dostave, iznos pouzeća (ako je ta usluga, inače 0). Visina, širina, dužina su u m. Težina je u kg. Paketi se upisuju prema vremenu prijema (kronološki).
Popis ulica	DZ_3_ulice.csv	Id, naziv, gps_lat_1, gps_lon_1 (GPS koordinate početka ulice), gps_lat_2, gps_lon_2 (GPS koordinate kraja ulice), najveći kućni broj
Popis mjesta	DZ_3_mjesta.csv	Id, naziv, ulica, ulica, ulica, ...
Popis osoba	DZ_3_osobe.csv	Osoba, grad, ulica, kbr Rade se o primateljima paketa, a oni mogu biti osobe i tvrtke. Mogu biti uključeni i pošiljatelji paketa, odnosno neki mogu biti i pošiljatelji i primatelji paketa.
Popis područja	DZ_3_podrucja.csv	Id, grad:ulica, grad:ulica, grad:* (sve ulice grada), ...

Priložene datoteke su samo jedan primjer sa svojim sadržajem. Nastavnik će svoje testiranje provoditi i na drugim datotekama i sadržajima. Neke od njih će se koristiti na prezentacijama zadaća. A druge će se koristiti za testiranje i bodovanje zadaća.

Predlaže se da studenti/ce pripreme svoje dodatne datoteke s kojima će testirati svoje programe. To se može uraditi tako da se priloženim datotekama dodaju novi zapisi, promijene podaci postojećim zapisima i/ili obrišu pojedini zapise. Posebno se predlaže da se pripreme datoteke u kojima će biti osim ispravnih zapisa i zapisi koji nisu ispravni u pojedinom segmentu (npr. nema prvi informativni redak, ima ili premalo ili previše podataka/atributa u retku, neispravna vrijednost u atributu koji je brojčanog tipa i sl).

Opis problema

Na početku je potrebno inicijalizirati sustav tvrtke za dostavu paketa tako da se učitaju datoteke u pravilnom redoslijedu te se kreiraju potrebni objekti za izvršavanje sustava.

Prilikom svakog učitavanja podataka iz datoteke potrebno je provjeriti ispravnost pojedinog zapisa/retka i ako nije ispravan potrebno ga je preskočiti i ispisati redni broj pogreške, sadržaj zapisa/retka i opis zašto je neispravan. Redni broj pogreške se odnosi na broj ukupnih pogrešaka tijekom rada sustava tj. **ne** odnosi se na redni broj greške unutar datoteke. Nakon što se učitaju sve potrebne datoteke potrebno je inicijalizirati virtualni sat prema podatku iz parametra (vs) prilikom izvršavanja programa, odrediti množitelj sekunde u virtualnom vremenu prema podatku iz parametra (ms) prilikom izvršavanja programa i zatim pripremiti za izvršavanje komandi. Tijekom rada programa može se izvršiti više komandi sve dok se ne upiše komanda Q. **Izvršavanje komande mora se temeljiti na uzorku dizajna Chain-of-Responsibility.**

Kako bi se mogle ispitati razlike u radu s paketima i vozilima uvedena je komanda za spremanje trenutnog stanja virtualnog sata, paketa i vozila te komanda za vraćanje spremljenog stanja virtualnog sata, paketa i vozila. Tako se može realizirati zamišljeni scenarij. Npr. na početku se spremi inicijalno stanje virtualnog sata, paketa i vozila pod nazivom 'početak', a zatim se izvrši komanda VR 2. Sada se može komandom IP dobiti uvid u status paketa. Slijedi spremanje stanja virtualnog sata, paketa i vozila pod nazivom 'verzija 1'. Nastavlja se vraćanjem spremljenog stanja virtualnog sata, paketa i vozila pod nazivom 'početak'. Sada se postavlja da je vozilo neispravno putem komande PS VŽ100PK NI. Ponovno se izvrši komanda VR 2. Slijedi spremanje stanja virtualnog sata, paketa i vozila pod nazivom 'verzija 2'. Ponovno se izvrši komanda IP. Sada se može utvrditi imali li razlika između paketa (pridruživanje vozilu i sl) u prvom i drugom ispisu. Ako je potrebno sada se može vratiti na spremljeno stanje 'verzija 1' i nastaviti s izvršavanje ostalih komandi.

Korisniku se daje mogućnost da izvrši sljedeće komande za aktivnosti:

- Pregled statusa paketa trenutku virtualnog vremena
 - Sintaksa:
 - IP
 - Primjer:
 - IP
 - Opis primjera:
 - Ispis tablice s primljenim i dostavljenim paketima (vrijeme prijema, vrsta paketa, vrsta usluge, status isporuke, vrijeme preuzimanja, iznos dostave, iznos pouzeća) u trenutno vrijeme virtualnog sata. Ispis primljenog paketa provjerava da li je vrijeme prijema manje od virtualnog vremena. Paket je dostavljen ako je vrijeme preuzimanja manje od virtualnog vremena.
- Izvršavanje programa određeni broj sati virtualnog vremena
 - Sintaksa:
 - VR hh
 - Primjer:
 - VR 4
 - Opis primjera:
 - Virtualni sat programa radi prema korigiranom broju sekundi. Npr. ako je -ms 600 znači da se u jednoj sekundi stvarnog vremena izvrši 600 sekundi virtualnog sata. To znači da se odradi jedna sekunda spavanja u stvarnom vremenu, korigirati virtualni sat, ispisati vrijeme virtualnog sata na ekran, provjeriti da li je stigao novi paket ili više njih u uredu za prijem. Primljeni paketi se ukrcavaju u vozilo (na kraju svakog punog sata prema gornjem opisu ukrcavanja paketa). Na ekran se ispisuje svaki paket koji je ukrcan u vozilo (i virtualno vrijeme). Kada dođe vrijeme za određeno vozilo ono može krenuti s dostavom kod primatelja. Na ekran se ispisuje kada vozilo krene na dostavu paketa. Na ekran se ispisuje svaki paket koji je isporučen primatelju (i virtualno vrijeme). Tako se izvršava virtualno vrijeme dok ne istekne zadano vrijeme izvršavanja ili dođe do kraja radnog vremena. Na ekran se ispisuje zašto je došlo do kraja rada. Može se više puta izvršiti komanda.
- Pregled podataka za vozila u trenutku virtualnog vremena
 - Sintaksa:
 - SV
 - Primjer:
 - SV
 - Opis primjera:
 - Ispis tablice s podacima o svim vozilima (status, ukupno odvoženi km, broj paketa u vozilu (hitnih, običnih, isporučenih), trenutni % zauzeća (prostora i težine), broj vožnji. Ispis podataka mora se temeljiti na uzorku dizajna Visitor.
- Pregled podataka za vožnje vozila tijekom dana
 - Sintaksa:

- VV vozilo
 - Primjer:
 - VV VŽ100PK
 - Opis primjera:
 - Ispis tablice s podacima o pojedinim vožnjama odabranog vozila (vrijeme početka, vrijeme povratka, trajanje, ukupno odvoženo km, broj paketa u vozilu (hitnih, običnih, isporučenih), % zauzeća (prostora i težine) na početku vožnje). Ispis podataka mora se temeljiti na uzorku dizajna Visitor.
- Pregled podataka za segmente vožnje vozila tijekom dana
 - Sintaksa:
 - VS vozilo n
 - Primjer:
 - VS VŽ100PK 1
 - Opis primjera:
 - Ispis tablice s podacima o segmentima 1. vožnje odabranog vozila (vrijeme početka, vrijeme kraja, trajanje, odvoženo km, paket (ako nije povratak u ured)). Ispis podataka mora se temeljiti na uzorku dizajna Visitor.
- Pregled područja s hijerarhijskim prikazom mjesta koja uključuje, a mjesta koje ulice. Svaka nova razina ima uvlačku od 4 praznine.
 - Sintaksa:
 - PP
 - Primjer:
 - PP
 - Opis primjera:
 - Ispis tablice s podacima o područjima, njihovim mjestima i ulicama mjesta koje ulaze u područje.
- Promjena statusa vozila u trenutku virtualnog vremena
 - Sintaksa:
 - PS vozilo [A | NI | NA]
 - A – aktivno
 - NI – nije ispravno
 - NA nije aktivno
 - Primjer:
 - PS VŽ100PK NI
 - Opis primjera:
 - Vozilu VŽ100PK se postavlja status da nije ispravno.
- Promjena statusa slanja obavijest za pošiljatelja ili primatelja paketa
 - Sintaksa:
 - PO 'primatelja/pošiljatelja' paket [D | N]
 - D – šalju se obavijesti
 - N – ne šalju se obavijesti
 - Primjer:
 - PO ' Pero Kos' CROVŽ0001 N

- Opis primjera:
 - Pošiljatelj Pero Kos ne želi primati obavijesti za paket CROVŽ0001.
- Spremanje trenutnog stanja virtualnog sata, paketa i vozila (NE koristiti serijalizaciju podataka)
 - Sintaksa:
 - SPV 'naziv'
 - Primjer:
 - SPV 'korak 1'
 - Opis primjera:
 - Spremanje trenutnog stanja virtualnog sata, paketa i vozila pod nazivom 'korak 1'.
- Povratak spremljenog stanja virtualnog sata, paketa i vozila (NE koristiti deserijalizaciju podataka)
 - Sintaksa:
 - PPV 'naziv'
 - Primjer:
 - PPV 'korak 1'
 - Opis primjera:
 - Povratak spremljenog stanja virtualnog sata, paketa i vozila pod nazivom 'korak 1'.
- Prekid rada programa
 - Sintaksa:
 - Q

Potrebno je dodati vlastitu funkcionalnost (aktivnost/komandu) projektu tako da se koristi GOF uzorak Proxy. To znači da je u dokumentaciju projekta **potrebno dodati opis funkcionalnosti** koja nije zadana u opisu zadaće i koja će se realizirati zadanim GOF uzorkom.

Potrebno je dodati vlastitu funkcionalnost projektu tako da se koristi GOF uzorak Decorator i to kod komande IP. Komanda se može proširiti dodatnim podacima. To znači da je u dokumentaciju projekta **potrebno dodati opis funkcionalnosti** koja nije zadana u opisu zadaće i koja će se realizirati zadanim GOF uzorkom.

Potrebno je napraviti program tj. aplikaciju za komandni/linijski mod/terminal u operacijskom sustavu putem kojeg će se izvršiti opisane akcije. Program se NEĆE izvršavati putem razvojnog alata (IDE). Aplikacija NE smije biti s grafičkim korisničkim sučeljem. **Kod izvršavanja programa NE smiju se ispisivati nepotrebni podaci (ostaci od testiranja i sl).**

Kod izvršavanja programa upisuje se naziv datoteke parametara (u jednoj liniji). Npr:
> java -jar {putanja}dkermek_zadaca_3.jar DZ_3_parametri.txt

Datoteka parametara sadrži parametre u obliku para ključ=vrijednost:

```
vp=DZ_3_vrste.csv  
pv=DZ_3_vozila.csv  
pp=DZ_3_paketi.csv  
po=DZ_3_osobe.csv  
pm=DZ_3_mjesta.csv  
pu=DZ_3_ulice.csv  
pmu=DZ_3_podrucja.csv  
mt=100  
vi=5  
vs=21.11.2023. 06:00:00  
ms=600  
pr=07:00  
kr=19:00  
gps= 46.305433, 16.336646  
isporuka=2
```

Provjeru ispravnosti upisanih argumenata/opcija koji su definirani u 1. zadaći treba refaktorirati tako da se obavlja u jednoj klasi na bazi metode kojoj se šalje argument tipa String u koji se prenose svi primljeni argumenti/opcije. Zatim se dodaje nova klasa za provjeru upisanih parametara koja sadrži metodu kojoj se šalje argument tipa Properties u koji se prenose svi parametri iz datoteka parametara. Metoda koristi objekt gornje klase za provjeru parametara koji su bili definirani u 1. zadaći tako da poziva njegovu metodu za provjeru. A zatim sama provjerava novo dodane parametre u 2. zadaći.

Redoslijed podataka je proizvoljan tako da treba voditi brigu kod obrade primljenih podataka. Vrijednosti za parametara su proizvoljne tako da će tijekom prezentacije zadaća biti neke druge, a treće kod testiranja i bodovanja zadaća. Nazivi datoteka mogu biti drugačiji nego što su ovdje prikazani.

Za izvršavanje programa prvo je potrebno postaviti da je važeći direktorij/mapa onaj na kojem se nalaze podaci (npr.: cd /home/UzDiz/DZ_3/podaci). To će biti mjesto s kojeg se izvršava program tako da nazivi datoteka s podacima ne smiju sadržavati putanju. Zbog toga se kod izvršavanja programa mora nalaziti putanja do njegove izvršne verzije (npr.: /home/UzDiz/DZ_3/dkermek_zadaca_3/target/dkermek_zadaca_3.jar).

```
> java -jar  
/home/UzDiz/DZ_3/dkermek_zadaca_3/target/dkermek_zadaca_3.jar  
DZ_3_parametri.txt
```

U ishodišnom direktoriju projekta treba priložiti datoteku dokumentacije {LDAP_korisničko_ime}_zadaca_3.pdf kako je opisano u dokumentima „Preporuke u vezi zadaća“ i „Opći model ocjenjivanja zadaća“.

Napomena: Preporučuje se refaktoriranje postojećeg rješenja 2. zadaće kako bi se mogli iskoristiti uzorci dizajna koji su obrađeni nakon 2. zadaće. Ne smiju se koristiti ugrađene osobine odabranog programskog jezika za realizaciju funkcionalnosti pojedinih uzoraka dizajna. Ne smiju se koristiti dodatne biblioteke/knjižnice klasa.

Metode u klasama NE smiju imati više od **30 linija programskog koda**, u što se ne broji definiranje metode, njenih argumenata i lokalnih varijabli. U jednoj liniji može biti jedna instrukcija. Linija ne može imati **više od 120 znaka**. Ne broji se linija u kojoj je samo vitičasta zagrada ili je prazna linija. Ne broje se linije u kojima se nalazi komentar osim u slučaju da se u komentaru nalazi programski kod.