

**Vježba: 4 – Zadaća 1. Višedretveni sustav "Meteo informacije o aerodromima" putem mrežne utičnice/socket-a**

Naziv projekta: {LDAP\_korisnik}-zadaca\_1

Korijenski direktorij treba biti {LDAP\_korisnik}-zadaca\_1

Sve nove klase trebaju biti u paketu **org.foi.nwtis.{LDAP\_korisnik}.zadaca\_1**. Za rad s postavkama treba koristiti Java biblioteku iz {LDAP\_korisnik}-zadaca\_1\_lib (nastao na temelju {LDAP\_korisnik}-vjezba\_03\_1). **Projekt se isključivo treba predati u formatu Eclipse IDE projekta s maven upravljanjem.** Prije predavanja projekta potrebno je **napraviti Clean na projektu** (i svim pomoćnim projektima kao što je {LDAP\_korisnik}-zadaca\_1\_lib). Zatim cijeli projekt (i sve pomoćne projekte kao što je {LDAP\_korisnik}-zadaca\_1\_lib) sažeti u **.zip** (NE .rar) format s nazivom **{LDAP\_korisnik}-zadaca\_1.zip** i predati u Moodle. Uključiti izvorni kod, primjere datoteka konfiguracijskih podataka (.txt, .xml, .bin, .json) i popunjeni obrazac za zadaću pod nazivom **{LDAP\_korisnik}-zadaca\_1.pdf** (u korijenskom direktoriju projekta). Primjeri datoteka konfiguracijskih podataka (.txt, .xml, .bin, .json) trebaju sadržavati identične podatke. U radu programa datoteka konfiguracijskih podataka je smještena na direktoriju s kojeg se pokreće program (npr. NWTiS\_dkermek\_zadaca\_1.txt). Program se može pokretati s različitih direktorija kako bi se mogao izvršavati s različitim datotekama konfiguracijskih podataka.

Struktura .zip datoteke predane zadaće treba biti sljedeća:

```
{LDAP_korisnik}-zadaca_1
  {LDAP_korisnik}-zadaca_1.pdf
  {LDAP_korisnik}-zadaca_1_app
    pom.xml
    NWTiS_{LDAP_korisnik}_1.txt (.xml, .bin, .json) - konfig. za ServerMeteo
    NWTiS_{LDAP_korisnik}_2.txt (.xml, .bin, .json) - konfig. za ServerAerodroma
    NWTiS_{LDAP_korisnik}_3.txt (.xml, .bin, .json) - konfig. za ServerUdaljenosti
    NWTiS_{LDAP_korisnik}_4.txt (.xml, .bin, .json) - konfig. za ServerGlavni
    src
      main
      ...
      test
      ...
  {LDAP_korisnik}-zadaca_1_lib
    pom.xml
    src
      main
      ...
```

**Nazivi klasa, nazivi atributa, nazivi metode, nazivi varijabli, komentari i sl. pišu se na hrvatskom jeziku u skladu s preporukama za Java jezik.** Metode u klasama NE smiju imati više **od 35 linija programskog koda**, u što se ne broji definiranje metode, njenih argumenata i lokalnih varijabli, prazna linija, linija samo s { ili }. U jednoj liniji može biti jedna instrukcija. Linija ne može imati više od **120 znaka**. Uvlačenje je **4 znaka**.

Klase i metode trebaju biti dokumentirane u **Javadoc** formatu.

Treba pripremiti **JUnit 5 testove** za sve metode kod središnjeg serverskog dijela (ServerGlavni, DretvaZahtjeva) s različitih tvrdnjama. Metode serverskog dijela NE SMIJU biti privatne.

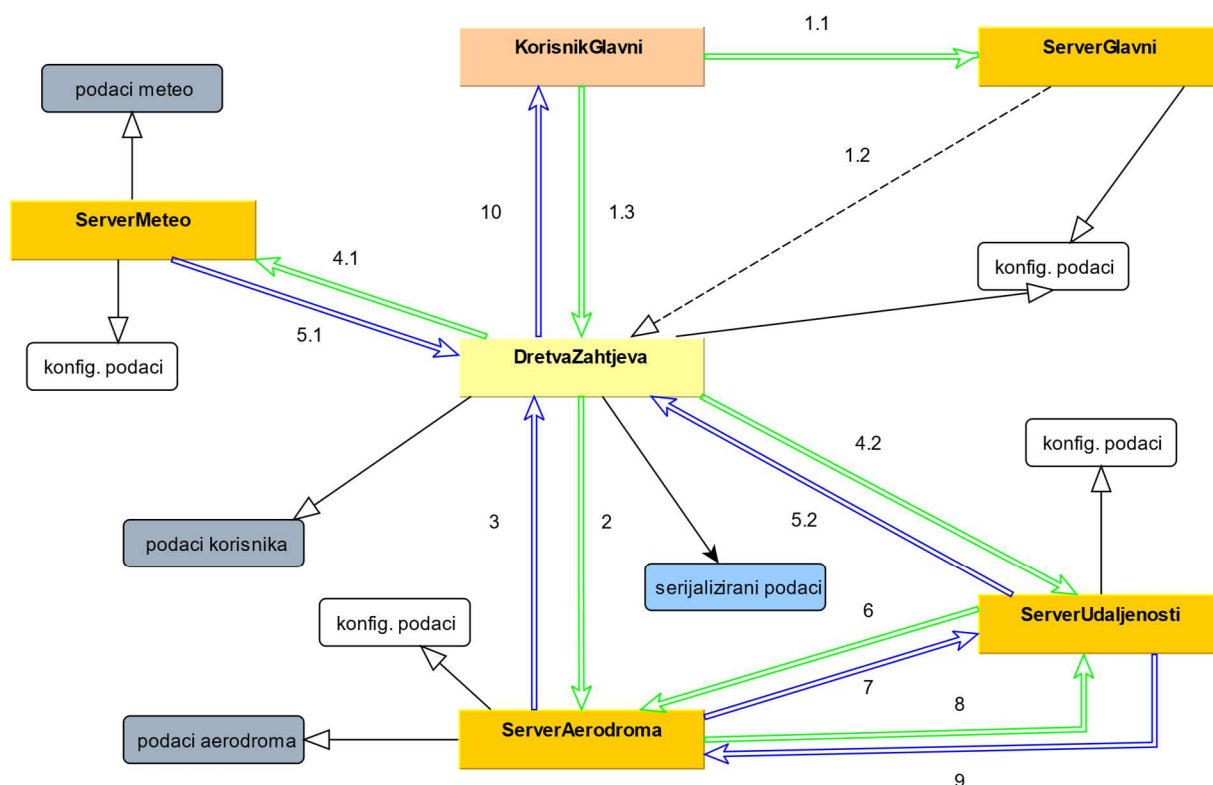
U projektu **ne smiju se koristiti klase i/ili metode koje su zastarjele** tj. anotirane su kao @Deprecated. Na Properties projekta u Java Compiler/Errors/Warnings kod Deprecated and restricted API **obavezno treba postaviti/označiti** *Signal use of deprecated API inside deprecated code* i *Signal overriding or implementing deprecated method*.

**Pripazite da nazivi postavki, nazivi glavnih klasa, komande zahtjeva i odgovori na zahtjeve budu u točnom obliku kako je definirano u opisu zadatke (velika, mala slova, razmak, točka, ...).** Ako je u opisu zadatke definirana klasa ServerMeteo onda točno tako mora biti vaša klasa, a ne Server\_Meteo ili Server\_Pristupa i sl. Ako u opisu zadatke piše da komanda sadrži "METEO" onda nije u redu "meteo" i sl. Kod testiranja će se koristiti nastavničke aplikacije koje će slati komande na vaše servere ili nastavnički server koji će primati vaše komande.

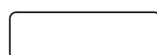
**Boduju se dijelovi koji su rađeni nakon vježbi!**

## Opis rada sustava:

Sustav se sastoji od tri poslužiteljska/serverska elementa i jednog korisničkog elementa. Shema sustava prikazana je na slici.



## Legenda:



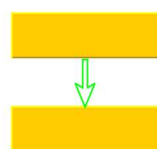
konfig. datoteka u txt, xml, bin ili json formatu



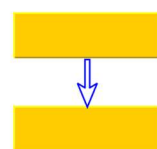
datoteka u csv formatu - znak odvajanja je ; (točka zarez)



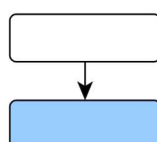
datoteka u binarnom formatu



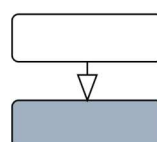
šalje zahtjev



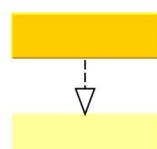
vraća odgovor



čita i upisuje podatke



čita podatke



kreira i pokreće dretvu

Prvo je potrebno pokrenuti poslužiteljske programe redoslijedom: ServerMeteo, ServerAerodroma, ServerUdaljenosti, ServerGlavni.

KorisnikGlavni svoj rad započinje spajanjem na mrežnu utičnicu (1.1<sup>1</sup>) na ServerGlavni. Kod spajanja na mrežnu utičnicu potrebno je postaviti maksimalno vrijeme čekanja na odgovor servera (opcija -t). Isto treba obaviti kada se jedan server postaje korisnik i spaja se na neki drugi server (ServerMeteo, ServerUdaljenosti, ServerAerodroma) s tim da se koristi postavka maks.cekanje. ServerGlavni radi višedretvenu obradu zahtjeva. Postoji situacija kada privremeno prelazi u jednodretveni rad. Po uspješnom spajanju kreira novu DretvaZahtjeva (1.2), pokreće ju te prelazi na čekanje sljedećeg zahtjeva. KorisnikGlavni šalje mrežnom utičnicom zahtjev (1.3) na DretvaZahtjeva. Prvo se provjeravaju korisnički podaci (autentikacija korisnika). Ako se radi o zahtjevu koji se odnosi na aerodrom/e tada šalje svoj zahtjev (2) na ServerAerodroma kako bi provjerila valjanost aerodroma. ServerAerodroma provjerava primljene podatke za aerodrom (icao<sup>2</sup>) u odnosu na podatke koje je učitao iz datoteke aerodroma. Zatim vraća odgovor (3). DretvaZahtjeva kod uspješne provjere nastavlja s realizacijom zahtjeva. Tu se može raditi o vraćanju popisa svih učitanih aerodroma, vraćanju informacije o određenom aerodromu, vraćanju meteo podataka za određeni aerodrom, utvrđivanju udaljenosti između dva aerodroma na bazi njihovih icao oznaka, brisanju podataka i sl. Kod zahtjeva za meteo podacima šalje se zahtjev mrežnom utičnicom (4.1) na ServerMeteo, koji provjerava postoje li meteo podaci za određeni aerodrom u podacima koje je učitao iz datoteke meteo podataka i vraća u odgovoru (5.1). Kod zahtjeva za utvrđivanje udaljenosti šalje se zahtjev mrežnom utičnicom (4.2) na ServerUdaljenosti, koji izračunava udaljenosti i vraća ju u odgovoru (5.2). Kod zahtjeva za spremanjem podataka provodi se prijelaz u jednodretveni rad, čeka se dok svoj rad ne završe ostale radne dretve, serijaliziraju se podaci u datoteku, vraća se u višedretveni rad. Kod zahtjeva za brisanjem podataka provodi se prijelaz u jednodretveni rad, čeka se dok svoj rad ne završe ostale radne dretve, brišu se svi podaci, vraća se u višedretveni rad. Kod zahtjeva za vraćanjem podataka provodi se prijelaz u jednodretveni rad, čeka se dok svoj rad ne završe ostale radne dretve, deserijaliziraju se podaci iz datoteke, vraća se u višedretveni rad. DretvaZahtjeva kod uspješnih akcija kao odgovor (10) vraća tekst OK, a zatim potrebne podatke. Kod neuspješnih akcija vraća tekst ERROR, status pogreške te opis pogreške. Kasnije su detaljnije opisani formati zahtjeva i odgovora. KorisnikGlavni ispisuje na konzoli sadržaj primljenog odgovora od servera.

Za utvrđivanje ispravnosti primljenih argumenata kod pokretanja pojedinih programa (ServerGlavni, ServerMeteo, ServerUdaljenosti, ServerAerodroma, KorisnikGlavni) potrebno je koristiti dozvoljene izraze (en. Regular Expression). Isto vrijedi i za utvrđivanje ispravnosti primljenih zahtjeva kod servera (DretvaZahtjeva, ServerMeteo, ServerUdaljenosti, ServerAerodroma).

---

<sup>1</sup> To je broj na slici koji se nalazi uz pojedinu vezu između elemenata.

<sup>2</sup> icao – 4 slovna jednoznačna međunarodna oznaka aerodroma. Sva slova su velika npr. LDZA, LOWW, EDDF.  
Više informacija na: <https://airportcodes.io/en/icao-codes/>, <https://www.world-airport-codes.com/>

Pokretanje prvog **serverskog programa** sadrži naziv klase i jedan parametar:

**ServerMeteo** datoteka. (txt | xml | bin | json)

Npr. **ServerMeteo** NWTiS\_dkermek\_1.txt

Serverski program na početku provjerava postoji li datoteka konfiguracije te prekida rad ako ne postoji. Ako postoji, učitava postavke iz datoteke konfiguracije. Slijedi provjera da li su zauzeta zadana računalna vrata/port (postavka port). Ako su zauzeta, potrebno je ispisati poruku na konzoli i prekinuti rad programa. Ako nisu zauzeta, provjerava se da li postoji datoteka s podacima o korisnicima (postavka datoteka.meteo) te ju učitava u svoju internu evidenciju/kolekciju. Ako ne postoji, ispisuje informaciju na konzoli i prekida rad. **ServerMeteo** radi jednodretvenu obradu zahtjeva.

Slijedi kreiranje objekta klase **ServerSocket** na zadanim računalnim vratima/port-u (postavka port) s maksimalnim brojem zahtjeva na čekanju (postavka maks.cekaca). Zatim se u petlji čeka da se spoji korisnik korisnik tog poslužitelja tj. **DretvaZahtjeva** sa **ServerGlavni**. Nakon spajanja korisnika obavlja se potrebna radnja prema zahtjevu korisnika.

Korisnik servera spaja se na server **ServerMeteo** putem mrežne utičnice/socket-a i šalje komandu serveru na temelju upisanih parametara i traži izvršavanje određene akcije:

- **METEO icao**
  - Npr. METEO LDZA
  - Provjerava postoje li za aerodrom meteo podaci u internoj kolekciji meteo podataka i vraća najsvježije meteo podatke. Korisniku se vraća odgovor OK temp vlaga tlak vrijeme;
  - Npr. OK 2,1 82,0 1010,0 2021-01-07 19:16:01.733;
- **METEO icao datum**
  - Npr. METEO LDZA 2021-01-07
  - Provjerava postoje li za aerodrom meteo podaci na određeni datum u internoj kolekciji meteo podataka i vraća sve meteo podatke tog datuma. Korisniku se vraća odgovor OK temp1 vlaga1 tlak1 vrijeme1; temp2 vlaga2 tlak2 vrijeme2;...
  - Npr. OK 4,4 87,0 1015,0 2021-01-07 16:01:01.718; 2,1 82,0 1010,0 2021-01-07 19:16:01.733;
- **TEMP temp1 temp2**
  - Npr. TEMP 4,0 5,0
  - Provjerava postoje li meteo podaci s temperaturom u rasponu od temp1 do temp2 u internoj kolekciji meteo podataka i vraća sve icao i meteo podatke koji zadovoljavaju uvjet. Korisniku se vraća odgovor OK icao1 temp1 vlaga1 tlak1 vrijeme1; icao2 temp2 vlaga2 tlak2 vrijeme2;...
  - Npr. OK LDZA 4,4 87,0 1015,0 2021-01-07 16:01:01.718; EHAM 4,0 87,0 1011 2021-01-07 15:44:40.117;
- **TEMP temp1 temp2 datum**
  - Npr. TEMP 4,0 5,0 2021-01-07
  - Provjerava postoje li meteo podaci na određeni datum s temperaturom u rasponu od temp1 do temp2 u internoj kolekciji meteo podataka i vraća sve icao i meteo podatke koji zadovoljavaju uvjet. Korisniku se vraća odgovor OK icao1 temp1 vlaga1 tlak1 vrijeme1; icao2 temp2 vlaga2 tlak2 vrijeme2;...
  - Npr. OK LDZA 4,4 87,0 1015,0 2021-01-07 16:01:01.718; EHAM 4,0 87,0 1011 2021-01-07 15:44:40.117;

Kodovi pogrešaka za **ServerMeteo** su:

- Kada format komande nije ispravan, vraća odgovor ERROR 10 tekst (tekst objašnjava razlog pogreške)
- Npr. ERROR 10 Format komande nije ispravan
- Kada nema aerodroma ili meteo podataka za aerodrom, vraća odgovor ERROR 11 tekst (tekst objašnjava razlog pogreške).
- Kada nešto drugo nije u redu vraća se odgovor ERROR 19 tekst (tekst objašnjava razlog pogreške).

Pokretanje drugog **serverskog programa** sadrži naziv klase i jedan parametar:

**ServerAerodroma datoteka.(txt | xml | bin | json)**

Npr. ServerAerodroma NWTiS\_dkermek\_2.txt

Serverski program na početku provjerava postoji li datoteka konfiguracije te prekida rad ako ne postoji. Ako postoji, učitava postavke iz datoteke konfiguracije. Slijedi provjera da li su zauzeta zadana računalna vrata/port (postavka port). Ako su zauzeta, potrebno je ispisati poruku na konzoli i prekinuti rad programa. Ako nisu zauzeta, provjerava se da li postoji datoteka s podacima o aerodromima (postavka datoteka.aerodroma) te ju učitava u svoju internu evidenciju/kolekciju. Ako ne postoji, ispisuje informaciju na konzolu i prekida rad. ServerAerodroma radi višedretvenu obradu zahtjeva na bazi interne klase za dretvu.

Slijedi kreiranje objekta klase ServerSocket na zadanim računalnim vratima/port-u (postavka port) s maksimalnim brojem zahtjeva na čekanju (postavka maks.cekaca). Zatim se u petlji čeka da se spoji korisnik tog poslužitelja tj. DretvaZahtjeva sa ServerGlavni. Nakon spajanja korisnika obavlja se potrebna radnja prema zahtjevu korisnika.

Korisnik servera spaja se na server **ServerAerodroma** putem mrežne utičnice/socket-a i šalje komandu serveru na temelju upisanih parametara i traži izvršavanja određene akcije:

- **AIRPORT**

- Npr. AIRPORT
- Vraća icao svih aerodroma iz interne kolekcije aerodroma. Korisniku se vraća odgovor OK icao1; icao2; icao3;...
- Npr. OK LDZA; LDVA; LOWW;...
- Uspješan odgovor može se spremi u međuspremnik

- **AIRPORT icao**

- Npr. AIRPORT LDZA
- Provjerava postoji li aerodrom u internoj kolekciji aerodroma. Korisniku se vraća odgovor OK icao "naziv aerodroma" GŠ GD
- Npr. OK LDZA "Zagreb Airport" 45.74070545276801 16.067404412140704
- Uspješan odgovor može se spremi u međuspremnik

- **AIRPORT icao brojKm**

- Npr. AIRPORT LDZA 500
- Provjerava postoji li aerodrom u internoj kolekciji aerodroma. Vraća sve aerodrome koji su udaljeni do brojKm od zadanog aerodroma. Za utvrđivanje udaljenosti između dva aerodroma šalje zahtjev na ServerUdaljenosti. Ako je u odgovor u redu vraća odgovor OK icao1 km1; icao2 km2;...
- Npr. OK LDVA 70; LDSP 312;
- Ako nema ni jednog aerodroma koji zadovoljava uvjet vraća OK
- Npr. OK
- Uspješan odgovor može se spremi u međuspremnik

Kodovi pogrešaka za **ServerAerodroma** su:

- Kada format komande nije ispravan, vraća odgovor ERROR 20 tekst (tekst objašnjava razlog pogreške)
- Npr. ERROR 20 Format komande nie ispravan
- Kada nema aerodroma, vraća odgovor ERROR 21 tekst (tekst objašnjava razlog pogreške).
- Kada ServerUdaljenosti ne radi, vraća odgovor ERROR 22 tekst (tekst objašnjava razlog pogreške).
- Kada nešto drugo nije u redu vraća se odgovor ERROR 29 tekst (tekst objašnjava razlog pogreške).



Pokretanje trećeg **serverskog programa** sadrži naziv klase i jedan parametar:

**ServerUdaljnosti datoteka.(txt | xml | bin | json)**

Npr. ServerUdaljnosti NWTiS\_dkermek\_3.txt

Serverski program na početku provjerava postoji li datoteka konfiguracije te prekida rad ako ne postoji. Ako postoji, učitava postavke iz datoteke konfiguracije. Slijedi provjera da li su zauzeta zadana računalna vrata/port (postavka port). Ako su zauzeta, potrebno je ispisati poruku na konzoli i prekinuti rad programa.

Slijedi kreiranje objekta klase ServerSocket na zadanim računalnim vratima/port-u (postavka port) s maksimalnim brojem zahtjeva na čekanju (postavka maks.cekaca). Zatim se u petlji čeka da se spoji korisnik tog poslužitelja tj. DretvaZahtjeva sa ServerAerodromaGlavni. Nakon spajanja obavlja se potrebna radnja prema zahtjevu korisnika. ServerUdaljnosti radi višedretvenu obradu zahtjeva na bazi interne klase za dretvu.

Korisnik servera spaja se na server **ServerUdaljenosti** putem mrežne utičnice/socket-a i šalje komandu serveru na temelju upisanih parametara i traži izvršavanja određene akcije:

- **DISTANCE icao1 icao2**
  - npr. DISTANCE LDZA LOWW
  - Provjera da li je postoje icao1 i icao2 u lokalnom spremniku aerodroma na servera. Ako bilo koji od njih ne postoji šalje se zahtjev na server ServerAerodroma kako bi se dobili podaci za aerodrom i spremili u lokalni spremnik. Ako postoje aerodromi ServerAerodroma vraća udaljenosti između dva aerodroma na temelju njihovih GPS pozicija. Algoritam treba uzeti u obzir zakrivljenost Zemlje. Vraća broj km. Korisniku se vraća odgovor OK broj
  - Npr. OK 271
  - Uspješan odgovor može se spremiti u međuspremnik
- **DISTANCE CLEAR**
  - npr. DISTANCE CLEAR
  - Briše sve aerodrome iz lokalnog spremnika. Korisniku se vraća odgovor OK
  - Npr. OK

Kodovi pogrešaka za **ServerUdaljenosti** su:

- Kada format komande nije ispravan, vraća odgovor ERROR 30 tekst (tekst objašnjava razlog pogreške)
- Kada ne postoji aerodrom(i) za traženi icao1 i/ili icao2, vraća odgovor ERROR 31 tekst (tekst objašnjava razlog pogreške)
- Kada ServerAerodroma ne radi, vraća odgovor ERROR 32 tekst (tekst objašnjava razlog pogreške).
- Kada nešto drugo nije u redu vraća se odgovor ERROR 39 tekst (tekst objašnjava razlog pogreške).

Pokretanje četvrtog **serverskog programa** sadrži naziv klase i jedan parametar:

**ServerGlavni datoteka.(txt | xml | bin | json)**

Npr. ServerGlavni NWTiS\_dkermek\_4.txt

Serverski program na početku provjerava postoji li datoteka konfiguracije te prekida rad ako ne postoji. Ako postoji, učitava postavke iz datoteke konfiguracije. Slijedi provjera da li su zauzeta zadana računalna vrata/port (postavka port). Ako su zauzeta, potrebno je ispisati poruku na konzoli i prekinuti rad programa. Ako nisu zauzeta, provjerava se da li postoji datoteka s podacima o korisnicima (postavka datoteka.korisnici) te ju učitava u svoju internu evidenciju/kolekciju. Ako ne postoji, ispisuje informaciju na konzoli i prekida rad. ServerGlavni u većini vremena radi višedretvenu obradu zahtjeva. Određene komanda prebacuju ServerGlavni u jednodretveni rad, obavljaju svoj dio posla i na kraju vraćaju ServerGlavni u višedretveni rad. Detaljnije piše kod pojedinih komandi.

Slijedi kreiranje objekta klase ServerSocket na zadanim računalnim vratima/port-u (postavka port) s maksimalnim brojem zahtjeva na čekanju (postavka maks.cekaca). Zatim se u petlji čeka da se spoji KorisnikServera. Nakon spajanja povećava se brojač aktivnih dretvi, kreira se dretva DretvaZahtjeva. DretvaZahtjeva ima naziv koji započinje {LDAP\_korisnik}\_ a slijedi redni broj dretve (od 0 pa dalje). Zatim se starta DretvaZahtjeva kako bi ona dalje obavila potrebne radnja prema zahtjevu korisnika. ServerGlavni može nastaviti u petlji za čekanje na spajanje sljedećeg korisnika.

Zahtjev KorisnikServera sadrži komandu. Komanda ima dva dijela: autentikacijski i predmetni. Autentikacijski dio sastoji se od:

**USER korisnik PASSWORD lozinka**

U predmetni dio ulazi sve što slijede autentikacijski dio (u EBNF notaciji):

**AIRPORT icao (brojKm)\***

**METEO icao (datum)\***

**TEMP datum temp1 temp2**

**CACHE (BACKUP | RESTORE | CLEAR | STAT)+**

**DISTANCE (icao1 icao2 | CLEAR)+**

Svaki zahtjev prvo prolazi provjeru/authenticaciju korisnika na bazi autentikacijskog dijela komande. Ako korisnik ne postoji vraća odgovarajuću poruku za pogrešku. Ako je autentikacija uspješna provjerava radi li se o predmetu komande koja čiji se odgovor može spremati u međuspremnik. Ako spada u tu skupinu onda se provjerava postoji li u međuspremniku odgovor na postavljeni zahtjev (predmet komande). Ako postoji vraća se odgovor te mu se povećava broj korištenja u međuspremniku. Ako ne postoji ili nije u skupini za međuspremanje slijedi obrada predmeta komande. Nakon obrade ako spada u tu skupinu za međuspremanje predmeta komande i odgovor se spremaju u međuspremnik, a broj korištenja je 1.

ServerGlavni i njegove dretve rade u višedretvenom načinu rada tako da je potrebno voditi brigu o međusobnom isključivanju dretvi kod operacija koje se bave promjenom vrijednosti pojedine djeljive varijable i sl.

Korisnik servera spaja se na server **ServerGlavni** putem mrežne utičnice/socket-a i šalje komandu serveru na temelju upisanih parametara i traži izvršavanja određene akcije:

- **USER korisnik PASSWORD lozinka AIRPORT icao**
  - npr. USER pero PASSWORD 123456 AIRPORT LDZA
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako su u redu šalje zahtjev na server ServerAerodroma za provjeru postoji li aerodrom s traženim icao. Korisniku se vraća odgovor OK icao "naziv aerodroma" GŠ GD
  - Npr. OK LDZA "Zagreb Airport" 45.74070545276801 16.067404412140704
- **USER korisnik PASSWORD lozinka AIRPORT icao brojKm**
  - npr. USER pero PASSWORD 123456 AIRPORT LDZA brojKm
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako su u redu šalje zahtjev na server ServerAerodroma da vrati sve aerodrome koji su udaljeni do brojKm od zadanog aerodroma. Korisniku se vraća odgovor OK icao1 km1; icao2 km2;...
  - Npr. OK LDVA 70; LDSP 312;
- **USER korisnik PASSWORD lozinka METEO icao**
  - npr. USER pero PASSWORD 123456 METEO LDZA
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako su u redu šalje zahtjev na server ServerAerodroma za provjeru postoji li aerodrom s traženim icao. Ako postoji šalje zahtjev na server ServerMeteo za važeće meteo podatke. Korisniku se vraća odgovor OK icao temp vlaga tlak vrijeme
  - Npr. OK LDZA 4,4 87,0 1015,0 2021-01-07 16:01:01.718
- **USER korisnik PASSWORD lozinka METEO icao datum**
  - npr. USER pero PASSWORD 123456 METEO LDZA 2021-01-07
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako su u redu šalje zahtjev na server ServerAerodroma za provjeru postoji li aerodrom s traženim icao. Ako postoji šalje zahtjev na server ServerMeteo za meteo podatke na datum. Korisniku se vraća odgovor OK icao; temp1 vlagal tlak1 vrijeme1; temp2 vlaga2 tlak2 vrijeme2;...
  - Npr. OK LDZA; 4,4 87,0 1015,0 2021-01-07 16:01:01.718; 2,1 82,0 1010,0 2021-01-07 19:16:01.733;
- **USER korisnik PASSWORD lozinka TEMP temp1 temp2**
  - npr. USER pero PASSWORD 123456 TEMP 4,0 5,0
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako postoji šalje zahtjev na server ServerMeteo za meteo podatke na datum. Korisniku se vraća odgovor OK icao1; temp1 vlagal tlak1 vrijeme1; icao2 temp2 vlaga2 tlak2 vrijeme2;...
  - Npr. OK LDZA 4,4 87,0 1015,0 2021-01-07 16:01:01.718; EHAM 4,0 87,0 1011 2021-01-07 15:44:40.117;
- **USER korisnik PASSWORD lozinka TEMP temp1 temp2 datum**
  - npr. USER pero PASSWORD 123456 TEMP 4,0 5,0 2021-01-07
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako postoji šalje zahtjev na server ServerMeteo za meteo podatke na datum. Korisniku se vraća odgovor OK icao1; temp1 vlagal tlak1 vrijeme1; icao2 temp2 vlaga2 tlak2 vrijeme2;...
  - Npr. OK LDZA 4,4 87,0 1015,0 2021-01-07 16:01:01.718; EHAM 4,0 87,0 1011 2021-01-07 15:44:40.117;

- **USER korisnik PASSWORD lozinka CACHE BACKUP**
  - npr. USER pero PASSWORD 123456 CACHE BACKUP
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako postoji prebacuje servera u jednodretveni rad, čeka se dok svoj rad ne završe ostale radne dretve, serijaliziraju se podaci međusprenika u datoteku (postavka datoteka.meduspremnika) vraća se u višedretveni rad. Korisniku se vraća odgovor OK
  - Npr. OK
- **USER korisnik PASSWORD lozinka CACHE RESTORE**
  - npr. USER pero PASSWORD 123456 CACHE RESTORE
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako postoji prebacuje servera u jednodretveni rad, čeka se dok svoj rad ne završe ostale radne dretve, deserijaliziraju se podaci međusprenika iz datoteke (postavka datoteka.meduspremnika), vraća se u višedretveni rad. Korisniku se vraća odgovor OK
  - Npr. OK
- **USER korisnik PASSWORD lozinka CACHE CLEAR**
  - npr. USER pero PASSWORD 123456 CLEAR
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako postoji prebacuje servera u jednodretveni rad, čeka se dok svoj rad ne završe ostale radne dretve, brišu se svi podaci međusprenika, vraća se u višedretveni rad. Korisniku se vraća odgovor OK
  - Npr. OK
- **USER korisnik PASSWORD lozinka CACHE STAT**
  - npr. USER pero PASSWORD 123456 CACHE STAT
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako postoji kreira privremenu datoteku i u nju upiše tablicu sa stupcima predmet komande, broj korištenja i vrijeme zadnjeg korištenja. Podaci se padajuće sortiraju prema broju korištenja i ispišu (poravnato: tekst lijevo, broj desno) u tablicu. Uzme se dužina datoteke. Korisniku se vraća odgovor OK dužinaDatoteke;sadržajDatoteke
  - Npr. OK 30127;Predmet komande            Broj korištenja            Zadnje vrijeme\nAIRPORT LDZA                            27                            19.02.2022. 11:05:14.826\nMETEO LDZA                                25                            19.02.2022. 11:07:02.691\nDISTANCE LDZA LOWW                      9                            19.02.2022. 10:23:57.028\n
- **USER korisnik PASSWORD lozinka DISTANCE icao1 icao2**
  - npr. USER pero PASSWORD 123456 DISTANCE LDZA LOWW
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako postoji šalje zahtjev na server ServerUdaljenosti za udaljenost između dva aerodroma. Korisniku se vraća odgovor OK broj
  - Npr. OK 271
- **USER korisnik PASSWORD lozinka DISTANCE CLEAR**
  - npr. USER pero PASSWORD 123456 DISTANCE CLEAR
  - Provjerava podatke korisnika u učitanim podacima iz datoteke korisnika. Ako postoji šalje zahtjev na server ServerUdaljenosti za brisanjem svih aerodroma iz lokalnog spremnika. Korisniku se vraća odgovor OK
  - Npr. OK

Kodovi pogrešaka za **ServerGlavni** su:

- Kada format komande nije ispravan, vraća odgovor ERROR 40 tekst (tekst objašnjava razlog pogreške)
- Kada ne postoji korisnik ili ne odgovara lozinka vraća odgovor ERROR 41 tekst (tekst objašnjava razlog pogreške)
- Kada server ServerMeteo ne radi vraća se odgovor ERROR 42 tekst (tekst objašnjava razlog pogreške).
- Kada server ServerAerdroma ne radi vraća se odgovor ERROR 43 tekst (tekst objašnjava razlog pogreške).
- Kada server ServerUdaljenosti ne radi vraća se odgovor ERROR 44 tekst (tekst objašnjava razlog pogreške).
- Kada nešto drugo nije u redu vraća se odgovor ERROR 49 tekst (tekst objašnjava razlog pogreške).

Pokretanje **korisničkog programa** sadrži naziv klase i parametre:

```
KorisnikGlavni -k korisnik -l lozinka -s (ipadresa | adresa) -p port -t cekanje  
(--aerodrom | --aerodrom icao | --aerodrom icao --km brojKm |  
--meteo icao | --meteo icao --datum datum |  
--tempOd temp1 --tempDo temp2 | --tempOd temp1 --tempDo temp2 --datum datum |  
--spremi | --vrati | --isprazni | --statistika |  
--udaljenost --aerodromOd icao1 --aerodromDo icao2 |  
--udaljenost --isprazni)
```

Korisnički program može se izvršavati više puta kako bi se koristili različiti načini rada na bazi primljenih argumenata. Opcije -s, -p i -t odnose se na podatke za spajanje na ServerGlavni. Komunikacija korisničkog programa servera temelji se na jednostavnom protokolu koji ima određenu sintaksu, a sadrži skup komandi za server. Komande i njihova sintaksa objašnjeni su uz opis servera. Izvršavanje korisničkog programa započinje utvrđivanjem vrste rada na bazi opcija u argumentima. Na temelju opcija kreira se komanda u skladu sa sintaksom servera s kojim komunicira. Kod određenih opcija potrebno je obaviti pretvaranje podataka iz primljenog oblika u oblik koji traži sintaksa komande na serveru. Komunikaciju otvara korisnik tako da šalje zahtjev serveru u obliku pripremljene komande. Server provodi analizu zahtjeva i ako je zahtjev u redu provodi određenu operaciju te vraća odgovor koji sadrži oznaku za uspješnu operaciju (OK) i ostale potrebne podatke. Kada zahtjev nije u redu, vraća odgovor koji sadrži oznaku za neuspješnu operaciju (ERROR), status i opis problema. Kontrola argumenata treba se obaviti u posebnim funkcijama unutar klase koja je zadužena za program. Korisnički program na kraju ispisuje primljeni odgovor od servera.

Dozvoljene vrijednosti za opcije:

- korisnik (min 3, maks 10 znakova) je korisničko ime i može sadržavati mala i velika slova, brojeve i znakove: \_ -
- lozinka (min 3, maks 10 znakova) je korisnička lozinka i može sadržavati mala i velika slova, brojeve i znakove: \_ -, #, !
- ipadresa je IPv4 adrese poslužitelja (npr. 127.0.0.1, 192.168.15.1)
- adresa je naziv poslužitelja (opisna adresa) (npr. localhost, dkermek.nwtis.foi.hr)
- port je broj računalnih vrata/port, a može biti u intervalu između 8000 i 9999
- cekanje je broj milisekundi za čekanje na odgovor servera
- icao, icao1 i icao2 su ICAO kodovi aerodroma (npr. Zagreb ima LDZA, Vienna ima LOWW, Varaždin ima LDVA)
- brojKm je broj km (npr. 500)
- datum je datum u hr formatu dd.mm.gggg. (npr. 19.02.2022.)
- temp1 i temp2 su temperatura u C. Može biti i negativna vrijednost. (npr. 0,0 -10,0 5,5)

Primjeri komandi (sljedom su povezane na prikazane komande kod ServerGlavni):

```
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --aerodrom
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --aerodrom LDZA
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --aerodrom LDZA --km 350
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --meteo LDZA
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --meteo LDZA --datum 19.02.2022.
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --tempOd -30,0 --tempDo 0,0
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --tempOd -30,0 --tempDo 0,0      --
datum 19.02.2022.
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --spremi
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --vrati
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --isprazni
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --statistika
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --udaljenost --aerodromOd LDZA
--aerodromDo LOWW
KorisnikGlavni -k pero -l 123456 -s1 localhost -p1 8000 --udaljenost --isprazni
```



*Tablica 1. Postavke koje trebaju sadržavati datoteke konfiguracije*

Ključ	Vrijednost	Server			
		Meteo	Aero droma	Udalje nosti	Glavni
port	port na kojem radi server, između 8000 i 9999	DA	DA	DA	DA
maks.cekaca	maksimalni broj korisničkih zahtjeva koji čekaju na ostvarivanje veze kod servera	DA	DA	DA	DA
maks.cekanje	maksimalno čekanje kod spajanja na server u ms		DA	DA	DA
datoteka. meteo	datoteka u kojoj se nalaze meteo podaci za aerodromime u csv formatu. Podaci su odvojeni znakom ; (točkazarez). Podaci su: icao; temp; vlaga; tlak; vrijeme	DA			
datoteka.aerodroma	datoteka u kojoj se nalaze podaci o aerodromima u csv formatu. Podaci su odvojeni znakom ; (točkazarez). Podaci su: icao; naziv; GŠ; GD		DA		
datoteka.korisnika	datoteka u kojoj se nalaze podaci o korisicima u csv formatu. Podaci su odvojeni znakom ; (točkazarez). Podaci su: prezime; ime; korisničko ime; lozinka				DA
datoteka.meduspremnika	datoteka u koju se provodi serijalizacija podataka međuspremnika i iz koje se provodi deserijalizacija podataka u međuspremnik				DA
server.meteo.adresa	adresa (IP ili opisna) poslužitelja ServerMeteo				DA
server.meteo.port	računalna vrata/port poslužitelja ServerMeteo				DA
server.aerodroma.adresa	adresa (IP ili opisna) poslužitelja ServerAerodroma			DA	DA
server.aerodroma.port	računalna vrata/port poslužitelja ServerAerodroma			DA	DA
server.udaljenosti.adresa	adresa (IP ili opisna) poslužitelja ServerUdaljenosti		DA		DA
server.udaljenosti.port	računalna vrata/port poslužitelja ServerUdaljenosti		DA		DA