

Specifikacija prvog projekta iz predmeta Softverske komponente - školska
2021/2022. godina

Implementacija komponente za upravljanje skladištem fajlova

Opis projekta:

Osmisliti i implementirati biblioteku (komponentu) koja će se koristiti za skladištenje različitih vrsta fajlova i podržavati skup operacija za upravljanje skladištem. Skladište treba da ima korisnike sa različitim privilegijama i da može da se konfiguriše. Komponentu treba realizovati tako da bude odvojena specifikacija (API) u posebnu biblioteku.

Pored specifikacije, potrebno je napraviti i dve implementacije ove specifikacije kao dve odvojene komponente (biblioteke). Prva implementacija skladišti fajlove na udaljeno skladište na Google Drive sa implementacijom autentifikacije preko gmail-a, a druga treba da skladišti fajlove u neko lokalno skladište (lokalni fajl sistem).

Sledi opis funkcionanosti komponente za upravljanje skladištem fajlova.

Inicijalizacija skladišta i korisnici

- inicijalizacija skladišta (može se implemenitrati kao kreiranje praznog foldera koji će biti korenski direktorijum skladišta),
- osmisliti da se komponentom podrži postojanje korisnika skladišta i napraviti konekciju na skladište sa odgovarajućim nalogom,
- svaki korisnik treba da ima korisničko ime i lozinku
- svaki korisnik ima privilegije, to su privilegija za snimanje fajlova, za preuzimanje, brisanje fajlova i pregled fajlova u skladištu, privilegije mogu da važe za celo skladište ili samo za neke foldere, mora da postoji neka logična raspodela privilegija, na primer ne može korisnik da ima privilegiju za brisanje, a da nema za pregled, ovo je potrebno programski iskontrolisati prilikom zadavanja privilegija,
- jedan korisnik kreira skladište i on ima sve privilegije i jedini on može da kreira ostale korisnike i da im dodeljuje privilegije,
- u implementaciji svake operacije nad skladištem vrši se kontrola privilegija,
- korisnik može da se diskonektuje sa skladišta,

- uvek samo jedan korisnik može da bude konektovan na skladište,
- podaci o korisnicima, njihovo korisničko ime i lozinka, kao i podaci o privilegijama čuvaju se u json fajlu u korenskom direktorijumu skladišta, potrebno je osmisлити odgovarajući format,
- kod implementacije udaljenog skladišta, ovi korisnici skladišta nemaju veze sa formalnom autentifikacijom na Google Drive.

Operacije nad skladištem

- kreiranje direktorijuma i praznih fajlova na određenoj putanji u skladištu (osmisлити razne načine zadavanja paternā kreiranja direktorijuma, kao u bash-u na primer `mkdir s{1..20}` kreira 20 direktorijuma pod imenom s1 do s20),
- smeštanje fajlova (jednog ili više) na određenu putanju u skladištu, putanja koja se prosleđuje treba da bude nezavisna od operativnog sistema, obraditi neke specifične situacije, na primer ako direktorijumi zadati putanjom ne postoje u skladištu,
- brisanje fajlova i direktorijuma iz skladišta,
- pregled sadržaja skladišta - podržati razne pretrage skladišta, na primer vrati sve nazive fajlove u direktorijumu, vrati nazive svih direktorijuma u nekom direktorijumu, vrati fajlove po imenu u direktorijumu i svim poddirektorijumima, vrati fajlove sa određenom ekstenzijom, vrati nazive fajlova sortirano po nazivu, datumu kreiranja ili modifikacije, vrati nazive fajlova koji su kreirani/modifikovani u nekom periodu, u nekom direktorijumu i sl.,
- premeštanje fajlova iz jednog direktorijuma u drugi,
- preuzimanje fajlova iz skladišta - zadaje se putanja koja može biti putanja do direktorijuma ili do fajla i odgovarajući element se preuzima iz skladišta,
- ukoliko se neke od ovih operacija ne mogu implementirati nad Google Drive-om obezbediti odgovarajuću obradu ove situacije, na primer da operacija izbaci izuzetak tipa *UnsupportedOperation*

Konfiguracija skladišta

- omogućiti da se zada veličina skladišta (u bajtovima) koja ne sme biti prekoračena,
- podržati mogućnost da se zadaju ekstenzije fajlova koji se ne mogu skladištiti (na primer može da se definiše da skladište ne prihvata exe fajlove) i vraćanje greške ako neko pokuša da uploaduje fajl sa ekstenzijom koja se ne prihvata,
- obezbediti da može se zada broj fajlova koji se mogu smestiti u određeni direktorijum,

- konfiguraciju skladišta može da radi samo korisnik koji je kreirao skladište,
- konfiguracioni fajl se takođe smešta u korenski direktorijum skladišta,
- prilikom izvršavanja operacija nad skladištem proveriti da li su zadovoljeni kriterijumi iz konfiguracije, na primer u operaciji koja snima fajl u skladište proveriti da li će njegovim dodavanjem biti prekoračena veličina i u koliko hoće, ne treba dozvoliti skladištenje.

Za komponentu specifikacije treba napisati API dokumentaciju kojom je precežno opisano korišćenje komponente.

Pored komponenti potrebno je implementirati program koji će se pozivati preko komandne linije i koji će ilustrovati korišćenje komponente za skladište. Program se pokreće sa zadatom putanjom za korenski element skladišta, ukoliko na toj putanji ne postoji skladište treba izvršiti inicijalizaciju skladišta i kreirati jednog glavnog korisnika (unose se korisničko ime i lozinka preko komandne linije). Ukoliko postoji skladište na zadatoj putanji, unosi se korisničko ime i lozinka i vrši se konekcija na skladište. Zatim korisnik naredbama preko komandne linije poziva operacije nad skladištem (snimanje fajlova, skidanje fajlova, kreiranje direktorijuma, pretraživanje skladišta). Glavni korisnik može da kreira druge korisnike sa određenim privilegijama.

Program treba da bude implementiran tako da radi sa specifikacijom komponente, i jednom implementacijom, ali da bude implementiran tako da poziva samo elemente specifikacije.

Prilagoditi implementaciju komponenti i aplikacije tako da komponente koje predstavljaju konkretne implementacije mogu da budu runtime dependensi programu za komandnu liniju (slično kao drajver za konkretnu bazu podataka u JDBC-u).

Pakovanje komponenti i programa za komandnu liniju, specifikacija dipendensija, kao i generisanje dokumentacije treba da bude automatizovano nekim build alatom.

Preporučeni jezik za implementaciju je Java, a alati za build Apache Maven ili Gradle.

Raspodela poena:

1. Dobro osmišljena specifikacija, kvalitet implementacije specifikacije - 10p
2. Implementacija dve komponente koje implementiraju specifikaciju
 - lokalno skladište - 5
 - udaljeno skladište - 5

(raspodela poena po zahtevima - inicijalzacija skladišta i korisnici - 30%, operacije nad skladištem 50%, konfiguracija - 20%, procenti se odnose na specifikaciju i dve implementacije, na primer ako ne uradite konfiguraciju, a sve ostalo kompletno,

možete dobiti maks 16 poena)

3. Kvalitetna dokumentacija za specifikacionu komponentu - 1p

4. Implementacija programa za komandnu liniju - 4p

5. Automatizacija pakovanja biblioteka i programa za komandnu liniju, generisanje dokumentacije korišćenjem build alata - 5p

Napomene:

Ukoliko tim odluči da ne radi program za komandnu liniju, potrebno je napraviti neki testni program bez komunikacije sa korisnikom, program treba da ilustruje rad implementiranih komponenti, ovaj program ne nosi poene, ali se mora napraviti da bi se projekat ocenio.

Bild alatom treba automatizovati pakovanje aplikacije za komandnu liniju tako da ona može bez dodatnih podešavanja da se pokreće izvan razvojnog okruženja (Eclipse, IntelliJ), proces pakovanja i pokretanja aplikacije se demonstrira na odbrani. Na odbrani student mora da pokaže razumevanje bild procesa i konfiguracije dependencija u bild alatu. Opisani zahtevi su uslov da bi student ostvario poene na 5. tački.

Rok za predaju i odbrana:

Projekat se radi u timu od dvoje i rok za završetak je termin odbrane, projekat se donosi na odbranu i pokazuje ispitivaču, nije potrebno slati ranije.

Projekat se brani u kolokvijumskoj nedelji, nedelju dana pre termina odbrane biće organizovana prijava timova za odbranu, a raspored odbrane će biti objavljen na materijalima. Studenti mogu, ako žele da projekat brane i ranije u dogovoru sa asistentima.

Članovi tima se posebno ocenjuju prema procenjenom uloženom trudu.