

Specifikacija prvog projekta iz predmeta Softverske komponente - školska
2022/2023. godina

Implementacija komponente za upravljanje skladištem fajlova

Opis projekta:

Osmisliti i implementirati biblioteku (komponentu) koja će se koristiti kao alat za rad sa skladištem fajlova i podržavati skup operacija nad skladištem, kao i pretraživanje skladišta. Potrebno je obezbediti i mogućnost konfigurisanja skladišta. Komponentu treba realizovati tako da bude odvojena specifikacija (API).

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.

Kreiranje skladišta i konfiguracija

- kreiranje skladišta je kreiranje praznog foldera koji će biti korenski direktorijum skladišta, prilikom kreiranja može se zadati konfiguracija, a postoji i podrazumevana konfiguracija,
- konfiguracija obuhvata mogućnost zadavanja veličina skladišta (u bajtovima) koja ne sme biti prekoračena,
- podržati mogućnost da se u konfiguraciji 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 se u konfiguraciji zadaje broj fajlova koji se mogu smestiti u određeni direktorijum,
- konfiguracioni fajl se 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.

Osnovne operacije nad skladištem

- kreiranje direktorijuma na određenoj putanji u skladištu (osmisliti razne načine zadavanja paternna 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,
- 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 u neki lokalni folder na zadatoj putanji (ako se radi o lokalnom skladištu, ta putanja mora biti izvan skladišta),
- preimenovanje fajlova i foldera u skladištu,
- sve putanje se uvek zadaju relativno u odnosu na korenski direktorijum
- ukoliko se neke od operacija ne mogu implementirati nad Google Drive-om obezbediti odgovarajuću obradu ove situacije, na primer da operacija izbacuje izuzetak tipa *UnsupportedOperation*

Pretraživanje skladišta

- implementirati operacije kojima se vrše različite pretrage nad skladištem, na primer:
 - vratiti sve fajlove u zadatom direktorijumu (vraća se naziv i metapodaci),
 - vrati sve fajlove iz nekog direktorijuma,
 - vrati sve fajlove u zadatom direktorijumu i svim poddirektorijumima,
 - vrati fajlove sa određenom ekstenzijom,
 - vrati fajlove koji u svom imenu sadrže, počinju, ili se završavaju nekim zadatim podstringom,
 - vratiti da li određeni direktorijum sadrži fajl sa određenim imenom, ili više fajlova sa zadatom listom imena
 - vratiti u kom folderu se nalazi fajl sa određenim zadatim imenom,
 - obezbediti zadavanje različitih kriterijuma sortiranja, na primer po nazivu, datumu kreiranja ili modifikacije, rastuće ili opadajuće,

- vrati fajlove koji su kreirani/modifikovani u nekom periodu, u nekom direktorijumu,
- omogućiti filtriranje podataka koji se prikazuju za fajlove koji su rezultati pretraga, na primer cela putanja (od korenskog direktorijuma) ili samo naziv, veličina fajla, datum kreiranja, modifikacije, i slično.

Za komponentu specifikacije treba napisati API dokumentaciju kojom je precežno opisano korišćenje operacija komponente, može na srpskom ili na engleskom.

Pored komponenti potrebno je implementirati program koji će se pozivati preko komandne linije i koji će ilustrovati korišćenje komponente za skladište. Program uvek radi samo sa jednom implementacijom skladišta (ili lokalno ili udaljeno) i sa jednim skladištem čija putanja se zadaje kao argument komandne linije prilikom pokretanja skladišta. Ukoliko na toj putanji ne postoji skladište treba kreirati skladište. Zatim korisnik naredbama preko komandne linije poziva operacije nad skladištem (smeštanje fajlova, skidanje fajlova, kreiranje direktorijuma, pretraživanje skladišta,...).

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 su komponente koje predstavljaju konkretne implementacije 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 pakovanje 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 - inicijalizacija skladišta i konfiguracija- 20%, osnovne operacije nad skladištem - 40%, pretraživanje skladišta - 40%, procenti se odnose na specifikaciju i dve implementacije, na primer ako ne uradite pretraživanje, a sve ostalo kompletno, možete dobiti maks 12 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 dipendensija u bild alatu i da domonstrira pokretanje apliakcije iz komandne linije, izvan razvojnog okruženja. 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 poslat na mejllistu i 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.