OBJEKTNO ORIJENTISANO PROGRAMIRANJE

- domaći zadatak broj 3 -

Na programskom jeziku C++ implementirati sistem fajlova na osnovu zadate specifikacije i konceptualnog opisa domena problema.

Konceptualni opis domena problema

Apstraktni objekat u sistemu fajlova (*FSObject*) ima naziv koji može da se pročita i promeni, može da prihvati apstraktnog posetioca i poseduje deskriptor pristupa (*AccessDescriptor*), koji definiše i kontroliše pristup tom objektu i može da se dohvati. Objekat može da se obriše, kopira i da mu se odredi veličina. Fajl (*File*) je objekat sistema fajlova čiji sadržaj je niz bajtova (u memoriji, tj. virtuelnom disku ograničenog kapaciteta). Omogućava čitanje i upis bajtova, kao i dohvatanje trenutne veličine (broj bajtova). Brisanje fajla zahteva i uklanjanje sadržaja. Kopiranje podrazumeva i kopiranje fizičkog fajla. Direktorijum (*Folder*) je objekat sistema fajlova koji sadrži druge objekte. Omogućava dohvatanje sadržanih objekata, dodavanje novog objekta, kao i uklanjanje zadatog objekta. Brisanje direktorijuma podrazumeva i brisanje objekata koje sadrži. Kopiranje direktorijuma vrši kopiranje njegovog sadržaja.

Deskriptor pristupa (*AccessDescriptor*) sadrži spisak naziva dozvoljenih operacija (*FSOperation*) koje mogu da se izvrše objektom, omogućava dodavanje operacije, kao i brisanje operacije iz skupa dozvoljenih. Apstraktna operacija (*FSOperation*) može da se izvrši i ima naziv koji može da se dohvati. Postoje sledeće operacije: a) čitanje fajla (*ReadFile*) – čita sadržaj zadatog fajla kao niz bajtova, b) pisanje u fajl (*WriteFile*) – upisuje zadati sadržaj u zadati fajl, c) pravljenje direktorijuma (*CreateDirectory*) – pravi prazan direktorijum sa zadatim imenom u zadatom direktorijumu, d) pravljenje fajla (*CreateFile*) – stvara prazan fajl sa zadatim imenom u zadatom direktorijumu, e) brisanje objekta (*DeleteObject*) – briše zadati objekat, f) listanje direktorijuma (*ListDirectory*) – vraća neposredno sadržane objekte u zadatom direktorijumu, g) *CopyPaste* – pravi kopiju odabranog objekta i smešta ga u zadati direktorijum, pretraga po imenu (*Search*) – pronalazi sve objekte u podstablu zadatog direktorijuma, čija imena sadrže zadati tekst. Zaštićena operacija (*ProtectedOperation*) je operacija koja „obavija" drugu operaciju i, pre nego što joj delegira izvršavanje, proverava ispunjenost preduslova. U opštem slučaju, operacija može da se izvrši ukoliko postoje adekvatna prava pristupa i ukoliko ima dovoljno prostora na virtuelnom disku. Ukoliko prava pristupa objekta ili objekata nad kojima treba izvršiti operaciju nisu adekvatna, podiže se izuzetak *AccessException*. Efekti neuspele operacije moraju se poništiti i sistem ostaviti u stanju kao da nikad nije ni započela.

Sistem fajlova (*Filesystem*) predstavlja fasadni interfejs sistema fajlova prema ostatku sistema. Sadrži koreni direktorijum i veličinu „virtuelnog diska" u bajtovima na kojem se pravi. Može da vrati veličinu preostalog slobodnog prostora. Sme da postoji najviše jedna instanca sistema fajlova u programu. Omogućava sledeće operacije: pravljenje fajla, pravljenje direktorijuma, čitanje sadržaja fajla, dohvatanje objekta u direktorijumu sa zadatim imenom, dodeljivanje pristupa objektu, ukidanje pristupa objektu, brisanje objekta, pretragu objekta, kao i dohvatanje/otvaranje direktorijuma na zadatoj putanji. Putanja je tekstualna vrednost koja sadrži niz imena direktorijuma razdvojenih znakom /, gde ne na početku koreni direktorijum (/), a na kraju nalazi ime objekta. Primer: /userA/documents/text.

Apstraktni posetilac (*FSVisitor*) omogućava obilazak sistema fajlova. Može da obradi fajl i direktorijum. Direktorijum se obrađuje tako što se prvo obradi samo objekat direktorijuma, a zatim svi objekti koje on sadrži. Posetilac za pretragu (*SearchVisitor*) omogućava pretragu objekta na osnovu zadatog imena i vraća kolekciju pronađenih objekata. Pretraga ignoriše foldere koje nije moguće čitati.

Za prijavljivanje i oporavak od grešaka koristi se mehanizam izuzetaka. Svi izuzeci su izvedeni iz klase **`exception`**. Postoji opšti izuzetak **`FSException`** koji sadrži objekat greške i opcionu informaciju o drugom izuzetku koji predstavlja uzrok greške (za ulančavanje izuzetaka). Izuzetak **`AccessException`** se koristi da signalizira greške usled neodgovarajućih prava pristupa. Izuzetak

`OperationFailedException` predstavlja opšti izuzetak za signaliziranje greške prilikom izvršavanja neke od operacija. Izuzeci ovog tipa su **`NameCollisionException`**, koji signalizira grešku pri dodavanju objekata pod imenom koje već postoji u direktorijumu, kao **`WriteFailedException`** usled nedostavka prostora na virtuelnom disku.

Napisati glavni program koji testira navedene usluge/operacije sistema fajlova (pozivaju se isključivo operacije fasadne klase). Glavni program treba da pozove funkciju **`void testFilesystem()`**.

Tehnički zahtevi i smernice za izradu rešenja

Opisane koncepte implementirati u vidu odgovarajućih klasa.Glavni program treba da poziva metode/operacije koje obavljaju opisane radnje. Programski kod klasa rasporediti u odgovarajuće **.h** i **.cpp** fajlove.

Napomene:

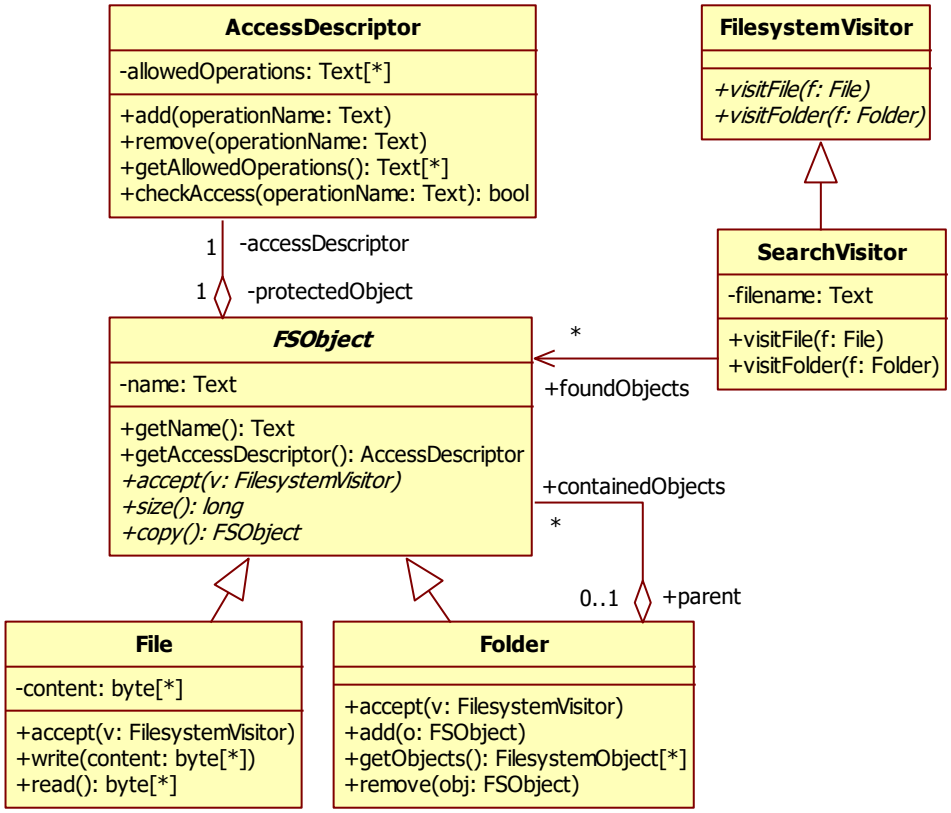
1. Za uspešno odbranjen domaći zadatak potrebno je na odbrani pokazati kod podeljen na odgovarajuće projekte, .h i .cpp fajlove.
   - Klase kojima su implementirani osnovni koncepti treba da budu smeštene u poseban projekat rešenja koji se prevodi kao dinamička biblioteka (**`fs_core.dll`**).
   - Glavni program napisati u posebnom projektu koji se prevodi kao Win32 Console Application (**`filesystem.exe`**) fajl i koji treba povezati sa navedenim bibliotekama.
   - NIJE DOZVOLJENO SMESTITI CEO KOD U JEDAN PROJEKAT ILI CPP fajl!
   - Rešenja će biti testiranja tajnim testovima, pa je potrebno da klase zadovolje opisani interfejs i da se imenuju na engleskom jeziku, kao što je napisano u tekstu zadatka i priloženom UML modelu.

13.12.2017. godine sa predmeta

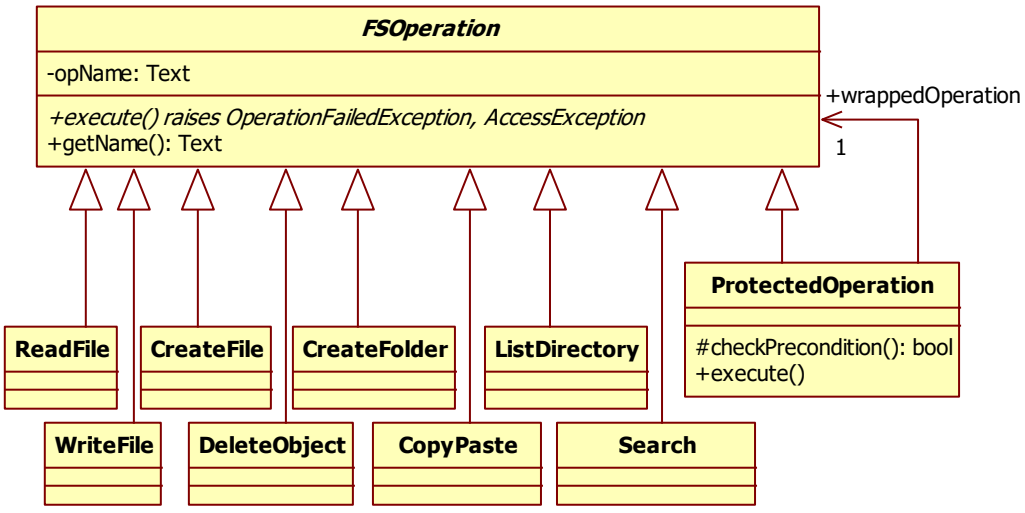
DODATAK

UML model opisanog sistema

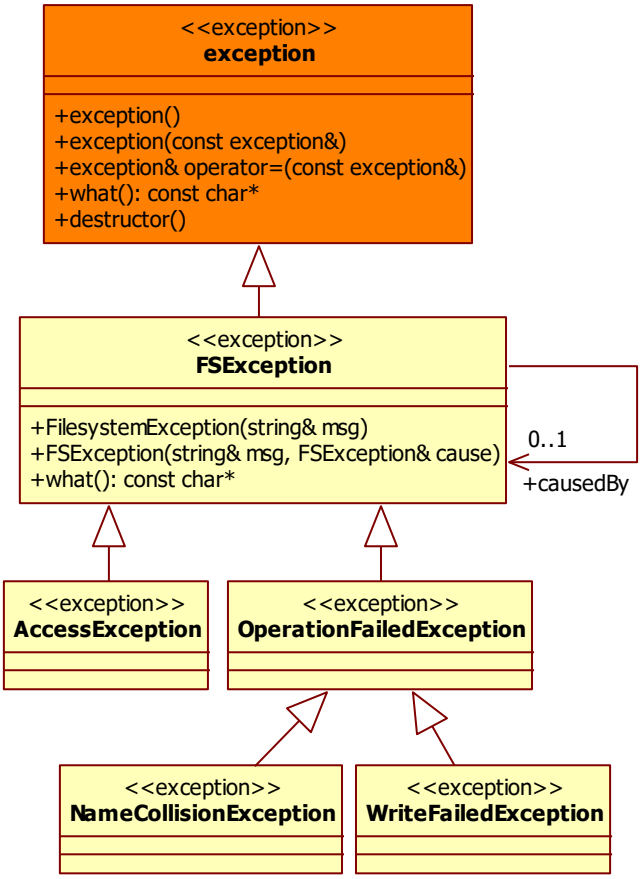
Konceptualni opis domena je nastao na osnovu datog UML modela sistema. Od studenata se očekuje da koriste i tekstualni opis i UML model kao relevantne izvore informacija za izradu domaćeg zadatka.



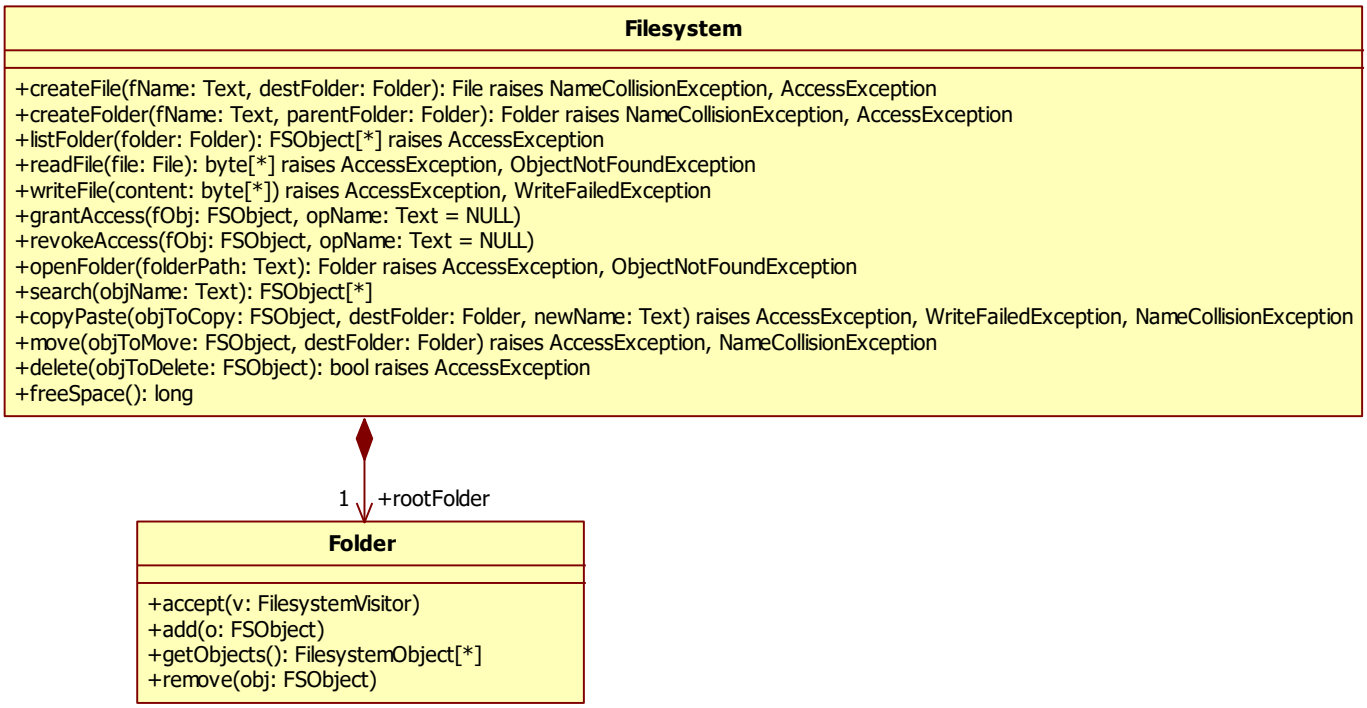
Slika 1 Filesystem objects.



Slika 2 Filesystem operations.



Slika 3 Filesystem exceptions.



Slika 4 Filesystem facade interface.