

Domaća zadaća 3.

Naziv: nadzor promjena strukture direktorija u VT100 terminalu

Naziv projekta: T{i}_{n}_zadaca_3 -> i=seminarska grupa (1,2), n=broj tima u seminarskoj grupi (01,10). Broj tima mora imati vodeću 0 ako je broj manji od 10 (01, 02, ..., 09, 10).

Ishodišni direktorij projekta: T{i}_{n}_zadaca_3

Naziv rješenja: T{i}_{n}_zadaca_3.zip

Prije predavanja projekta potrebno je napraviti Clean na projektu. Zatim cijeli projekt sažeti u .zip (NE .rar) format s nazivom **T{i}_{n}_zadaca_3.zip** i predati u Moodle. Uključiti izvorni kod, primjere datoteke podataka, popunjenu datoteku dokumentacije i popunjen obrazac za zadaću (obje datoteke u ishodišnom direktoriju projekta).

Program se mora izvršavati unutar programa koji omogućava emulaciju VT100 terminala. Nastavnik koristi ConEmu (<http://www.fosshub.com/ConEmu.html>) i zadaće će se izvršavati u tom programu pa je preporuka da ga koristite i studenti.

Program se MORA izvršavati s komandne linije unutar VT100 emulatora pri čemu vrijedi sljedeći format:

putanja1_naziv.programa	brojRedaka
brojStupaca	podjelaOkvira
putanja2_naziv	brSekundi

brojRedaka = 24 - 40

brojStupaca = 80 - 160

podjelaOkvira = V, O

brSekundi = 1 - 120

Npr.

java -jar C:\UzDiz\dkermek_zadaca_3.jar 40 160 O C:\UzDiz\test 30

ili

C:\UzDiz\dkermek_zadaca_3.exe 40 160 O C:\UzDiz\test 30

Program kod pokretanja provjerava upisane argumente, priprema ekran za rad između ostalog tako da dijeli ekran u dva logička dijela (okvira) od kojih je gonji za prikaz podataka i odgovara brojuRedaka, a donji je za unos podataka i zauzima samo jednu red. Okvir za prikaz podataka dijeli se u dva jednaka dijela, prozora (ako je podjelaOkvira V na lijevi (1) i desni (2), ako je podjelaOkvira O na gornji (1) i donji (2)).

Za prikaz podataka u pojedinom prozoru treba koristiti **direktno pozicioniranje na temelju ANSI/VT100 kontrolnih ESC**

kodova (<http://www.termsys.demon.co.uk/vtansi.htm>). Odnosno, sve radnje vezane uz ekran treba obavljati na temelju **ANSI/VT100 kontrolnih ESC kodova**.

Kada ispis stigne do zadnje linije u prozoru, kod sljedećeg prikaza potrebno je obrisati sadržaj prozora, postaviti kursor na početnu poziciju u prozoru i nastaviti prikaz. Sve akcije oko prikazivanja podataka trebaju biti realizirane vlastitim rješenjem. Nije dozvoljeno koristiti biblioteke za prikaz podataka u VT100 terminalu.

Program zatim učitava strukturu direktorija i datoteka počevši od direktorija koji je određen argumentomputanja2_naziv. Ta struktura treba biti organizirana kao logička struktura u kojoj zadani direktorij predstavlja korijenski direktorij. Program treba upamtiti podatke o nazivu, vrsti, vremenu promjene/kreiranja svakog pojedinog direktorija i datoteke, a za datoteke i veličina. Kod direktorija veličina je jednaka zbroju svih njegovih datoteka kao i datoteka njegovih poddirektorija. Ažuriranje veličine direktorija provodi se kod dodavanja svake datoteke koja se nalazi u njegovoj grani strukture. Za pojedinog elementa strukture ne smije se pamti puni naziv tj. putanju nego informacija (referenca) o njegovom roditelju u strukturi. Veza prema fizičkoj strukturi (stvarnoj poziciji na datotečnom sustavu) ostvarena je kroz ishodišnji direktorij koji jedini sadrži svoju apsolutnu adresu.

Prilikom učitavanja strukture u 1. prozoru prikazuju se elementi kako se učitavaju uz naglašavanje hijerarhijskog odnosa uvlačenjem za 2 mjesta i različitom bojom za direktorije i datoteke. Nakon što se kreira pojedini direktorij ili datoteka u strukturi u 2. prozoru se prikazuje broj kreiranih direktorija, broj kreiranih datoteka i ukupna veličina cijele strukture.

Za prolaženje po strukturi treba koristiti **isključivo** GOF uzorak Iterator.

Nakon toga program čeka komandu koja se unosi tipkovnicom u okviru za unos podataka. Ukoliko je dretva u stanju izvršavanja tada se unos obavlja u zelenoj boji. A ako je dretva u stanju prekida tada se unos obavlja u žutoj boji. Komanda može biti:

- 1 - ispis ukupnog broja direktorija i datoteka u strukturi (prikaz u 1. prozoru)
- 2 - ispis sadržaja strukture direktorija i datoteka uz prikaz naziva, vremena (formatiranog u HR obliku), veličina (u formatu 999.999.999 B) (prikaz u 1. prozoru)
- 3 - izvršavanje dretve (prikaz u 1. prozoru)
- 4 - prekid izvršavanja dretve (prikaz u 1. prozoru)
- 5 - ispis informacija o svim spremljenim stanjima (redni broj i vrijeme spremljenja) (prikaz u 1. prozoru)
- 6 n - postavljanje stanja strukture na promjenu s rednim brojem n čime ono postaje novo trenutno stanje strukture (prikaz u 1. prozoru)
- 7 m - uspoređivanje trenutnog stanja strukture i promjene s rednim brojem m (prikaz u 1. prozoru)
- 8 - ponovno učitavanje strukture uz poništavanje svih spremljenih stanja strukture (prikaz kao i kod inicijalnog učitavanja strukture)
- 9 - dodana vlastita funkcionalnost (prikaz u 1. prozoru)
- Q - prekid rada programa.

Argument programa `brSekundi` određuje ukupno vrijeme jednog ciklusa provjere stanja strukture u kojem dretva uspoređuje trenutno stanje pojedinog elementa strukture i njenog stvarnog predstavnika u datotečnom sustavu. Ako je došlo do bilo koje promjene u odnosu na prethodni sadržaj strukture potrebno je u 2. prozoru ispisati informaciju (vrijeme provjere, putanja, naziv elementa, vrsta promjene). Na kraju je potrebno spremati postojeće stanje strukture, samo ako je bilo promjene unutar strukture. Ako nije bilo promjene ispisuje se na 1. prozoru da nije bilo promjene (vrijeme, tekst). Dretva NE smije biti prekinuta ako je u aktivnom stanju ciklusa kada provjerava stanje strukture.

Treba pronaći prikladne GOF i POSA1 uzorke dizajna za opisane probleme. Za rad s korisničkim sučeljem treba se koristiti MVC ili PAC uzorak dizajna. Potrebno je dodati vlastitu funkcionalnost projektu tako da se koriste GOF uzorak Visitor i POSA1 uzorak Leyers. To znači da je potrebno dodati u dokumentaciju projekta opis funkcionalnosti koja nije zadana u opisu zadaće i koja će se realizirati GOF uzorkom Visitor i POSA1 uzorkom Leyers.

U ishodišnom direktoriju projekta treba priložiti datoteku dokumentacije `T{i}_n_zadaca_3.{doc|pdf}` u kojoj se na 1. stranici nalazi objašnjenje razloga odabira pojedinog uzorka dizajna (max 1 str A4, font min 10) a na 2. stranici dijagram rješenja (max 1 str A4, font min 10).

U ishodišnom direktoriju projekta treba priložiti datoteku `{LDAP_korisničko_ime}_obrazac_za_zadacu_3.{doc|pdf}` u kojoj se nalazi popunjen **NOVI** obrazac za zadaću. Uskoro će biti objavljen obrazac.

Voditelj tima treba predati uz projekt i obje dokumentacije. Član tima treba predati samo popunjen **NOVI** obrazac za zadaću (bez projekta i objašnjenja odabira) uzoraka.

Napomena: Ne smiju se koristiti ugrađene osobine odabranog programskog jezika za realizaciju funkcionalnosti pojedinih uzoraka dizajna.

Jezici implementacije: C#, Java. Isti koji je odabran u anketi za programski jezik za zadaće.

Razvojni alati: NetBeans (8.*), Visual Studio 201* (preporučuje se 2013)