

Rad sa modelom drveta, primer učitavanja podataka o fajl sistemu

Zadatak je podeljen u nekoliko segmenata. U svakom od njih su detaljno obrazloženi koraci potrebni za implementaciju. Studentu se sugeriše korišćenje online sadržaja i pretraga potrebnih fragmenata koda.

1. Kreirati početnu stranicu tj. osnovni okvir, koji je korišćen u ranijim vežbama. Postarati se da okvir bude centriran i veličine 500 x 500 tačaka.
2. Napisati funkciju koja rekurzivno učitava fajl sistem tj. C particiju i istestirati je pozivom kroz main funkciju, bez pokretanja grafičkog interfejsa:
 - a. Da bi se ovo uradilo potrebno je koristiti klasu **File** koja ima vrlo koristan metod **listFiles**. Pored te funkcije potrebno je uočiti i svojstva File-a, a to su Name, Path, isDirectory, lastModified i slično..
 - b. Ispisati rezultat na standardni izlaz u bilo kom redosledu, bitno je samo pobrinuti se da funkcija radi.
 - c. Poboľjšati funkciju u smislu dodavanja parametra maksimalne dubine obilaska fajl sistema. Tako da npr. ako postavimo max. dubinu na 1 i primenimo na particiju C:// dobijamo izlaz poput:
 - i. Program Files
 - ii. Documents and Settings
 - iii. Temp
 - iv. ...
3. (Vraćamo se na GUI)
 - a. Kreirati komponentu **JScrollPane** i dodati je u centralni deo kreiranog okvira, na kojem je postavljen BorderLayout.
 - b. Kreirati komponentu **JTree** i ne dodavati je nigde, već metodom **setViewportView** iz komponente JScrollPane postaviti pogled na kreirano drvo. Zarad testiranja kontrole JTree prilikom pozivanja njenog konstruktora proslediti kolekciju Vector, koja sadrzi nekoliko imena studenata.
 - c. Sada primeniti metodologiju sa ranijih časova, a to je rad sa modelima:
 - i. JTree ima podrazumevani model koji se naziva **DefaultTreeModel**. Naslediti ga i novi model nazvati npr. **FSTreeModel**. Pobrinuti se da postoji odgovarajući podrazumevani konstruktor, ne zaboraviti nasleđivanje konstruktora kroz poziv super();
 - ii. Kreirati klasu **FSTreeNode** koja nasleđuje **DefaultMutableTreeNode**, gde ova default varijanta predstavlja osnovni čvor koji se koristi u radu sa JTree. Nama je potreban donekle prilagođen jer imamo dodatnih informacija, a to su:
 1. naziv
 2. putanja
 3. jeDirektorijum
 4. datumPoslednjeIzmene
 - iii. U okviru modela napraviti novi konstruktor koji prihvata 2 argumenta: FSTreeNode i int koji redom predstavljaju koreni čvor (u našem slučaju onaj sa C://) i maksimalnu dubinu do koje učitavamo.

- d. Prilagoditi ranije napisanu funkciju za čitanje fajl sistema, tako da umesto ranijeg argumenta File, sada prihvata FSTreeNode koji u sebi već sadrži pohranjene informacije o putanji i slično, pa se iz toga mogu izlistati poddirektorijumi i fajlovi. Prilagođenu funkciju pozvati odmah u konstruktoru modela.
 - e. JTree komponenti pridružiti novo kreirani model sa npr. FSTreeNode-om, koje cilja na "C:/" i ima dubinu 2. Pokrenuti i istestirati aplikaciju. Probat i nakon toga sa dubinom 3.. Šta je problem, diskusija...
4. Poboljšavanje:
- a. Ideja je ne "keširati" sve informacije prilikom kreiranja GUI-ja, jer fajl sistem običi rekurzivno predstavlja ozbiljan trošak. Kao i kod list i slider modela, i ovde postoje određene akcije i njihovi oslušivači koje možemo iskoristiti. JTree objektu možemo pridružiti tzv. **TreeExpansionListener** i redefinisati njegov hendler **treeExpanded**, tako da prilikom klika na deo drveta ažuriramo samo taj deo drveta.
 - b. Korišćenjem reference na **event** objekat **getPath().getLastPathComponent()**, uzimamo poslednji kliknut čvor, a zatim je taj čvor potrebno proslediti funkciju koju definišemo nad FSTreeModel-om, koja će se npr. zvati **ucitajCvor(FSTreeModel cvor)**.
 - c. Gore navedena funkcija će ažurirati samo deo drveta koji predstavlja decu prosleđenog čvora i na kraju pozivamo kao i ranije kod **JList** signal koji inicira renderovanje samo tog dela drveta **fireTreeNodesInserted**.