

# Razvoj programske aplikacije za vizualizaciju relacija

---

**Vlah, Lorena**

**Undergraduate thesis / Završni rad**

**2016**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:235:836139>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-10-01**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Lorena Vlah  
0036464647

Zagreb, 2016. godina

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

**RAZVOJ PROGRAMSKE  
APLIKACIJE ZA  
VIZUALIZACIJU RELACIJA**

Mentor:

Prof. dr. sc. Neven Pavković, dipl. ing.

Student:

Lorena Vlah  
0036464647

Zagreb, 2016. godina.

Izjavljujem da sam ovaj rad izradila samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.

Zahvaljujem se profesoru Pavkoviću na savjetima i preporučenoj literaturi.

Lorena Vlah



# SADRŽAJ

SADRŽAJ .....	1
SAŽETAK.....	4
1. Uvod.....	6
1.1 Inženjerski objekti i datoteke .....	6
1.2 Matrične metode za prikaz relacija između inženjerskih objekata .....	6
1.3 Programiranje u Visual Basic-u „for applications“ .....	7
1.3.1 Excel aplikacije .....	7
1.3.2 Visual Basic Editor .....	8
1.3.3 Code Window prozor .....	9
1.3.4 Tipovi podataka, varijable i područje varijabli .....	10
1.3.5 Usmjeravanje toka programa uz pomoć naredbi.....	11
1.3.6 Korištenje matrica .....	11
1.3.7 Procedure .....	11
1.3.8 Objekti, metode, svojstva i događaji .....	12
1.4 Objekti u Microsoft Excelu.....	12
1.4.1 Excelov objektni model .....	12
1.4.2 Object Browser .....	13
1.4.3 Forma kao korisničko sučelje u Excelu .....	14
1.4.4 Osnovne kontrole u VBA.....	15
1.5 Treeview kontrola .....	18
1.5.1 Objekt čvor ( <i>Node</i> ) .....	18
1.5.2 Metode TreeView kontrole .....	20
1.5.3 Svojstva TreeView kontrole.....	21
1.5.4 Događaji TreeView kontrole.....	22
2. Struktura aplikacije za vizualizaciju relacija .....	24
2.1 Unaprijed uređeni radni listovi.....	24
2.2 Forma aplikacije za vizualizaciju relacija .....	25
2.2.1 Kartica “Prikaz podataka i unos značajki“ .....	25
2.2.2 Kartica “Manipulacija strukturom podataka“ .....	27
3. Upute za korištenje aplikacije za vizualizaciju relacija.....	31
3.1 Pokretanje aplikacije .....	31
3.2. Popunjavanje i prikaz TreeView kontrole.....	33
3.3 Manipulacija objektima koji su hijerarhijski strukturirani .....	36
3.4 Manipulacija datotekama .....	37
3.5 Prikaz podataka i unos značajki .....	40
4. ZAKLJUČAK.....	43
5. LITERATURA .....	44
PRILOZI.....	45

POPIS SLIKA

Slika 1:Visual Basic Editor .....	9
Slika 2:Code Window .....	10
Slika 3:Object Browser .....	14
Slika 4:Forma aplikacije za vizualizaciju relacija.....	15
Slika 5:Okvir Toolbox sa osnovnim kontrolama .....	16
Slika 6:Kontrole Label i TextBox .....	16
Slika 7:Gumb za pokretanje aplikacije Vizualizacija relacija.....	17
Slika 8:MsgBox.....	18
Slika 9:InputBox .....	18
Slika 10: Primjer TreeView kontrole u MS formi.....	19
Slika 11: Opcija Additional Controls .....	20
Slika 12:Svojstva TreeView kontrole koja možemo postaviti u prozoru Properties VBE Editora .....	22
Slika 13:Primjer realizacije hijerarhijske strukture sklop/komponente pomoću TreeView kontrole u Solidworks-u .....	23
Slika 14:Unaprijed postavljeni radni listovi aplikacije .....	24
Slika 15: Kartica “Prikaz podataka i unos značajki“ .....	25
Slika 16:Radni list “Matrica“ nakon klika na gumb “Prikaži vezu podataka i datoteka u matrici“, s postavljenim vezama .....	26
Slika 17: Radni list “Matrica2“ nakon klika na gumb “Prikaži vezu među podacima u matrici“, s postavljenim vezama .....	26
Slika 18: Kartica “Manipulacija strukturom podataka .....	27
Slika 19: (Lijevo) Struktura radnog list korijenskih članova .....	28
Slika 20: (Desno) Struktura radnog lista s Child čvorovima kojima su Parent čvorovi korijenski čvorovi.....	28
Slika 21: Open Dialog .....	29
Slika 22:MsgBox nakon klika na gumb “Spremi datoteke“ .....	29
Slika 23:Izbornik u kojem je moguće kreirati ikonu za pokretanje aplikacije preko vrpce.....	31
Slika 24: Radni list “Postavke“ .....	32
Slika 25: Kopiranje nekog radnog lista radne knjige u list iz kojeg se čitaju podaci za TreeView .....	33
Slika 26:Radni listovi kreirani od strane korisnika nazvani (redom slijeva) “Fakultet“, “Studiji“, “Glavni_smjerovi“ .....	33
Slika 27: Radni list kreiran od strane korisnika nazvan “Smjerovi“ .....	34
Slika 28:Gumb “Kopiraj u radni list“ i povratni MsgBox .....	34
Slika 29: Nakon što se klikne OK u MsgBox-u “Kopiranje podataka“, pojavljuju se još dvije povratne poruke .....	34
Slika 30:MsgBox gumba “Kopiraj u radni list“ .....	35
Slika 31: Podaci učitani u TreeView kontrolu s omogućenim gumbima za manipulaciju (klikom na čvor Konstrukcijski 1).....	35
Slika 32:Unos još jednog korijenskog čvora (uz FSB) .....	36
Slika 33:Prikaz podataka u TreeView kontroli nakon izmjena učinjenih preko gumbi forme	36
Slika 34:Listovi kreirani klikom na gumb “Spremi podatke“ .....	37
Slika 35:Sadržaj listova kreiranih klikom na gumb “Spremi podatke“ .....	37
Slika 36:Dijaloški okvir Open Dialog.....	38
Slika 37:MsgBox koji se pojavi nakon spremanja datoteka .....	38
Slika 38.Mapa Datoteke projekta u koju su preko aplikacije pohranjene datoteke .....	39
Slika 39: Radni list datoteke nakon unesenih promjena, prvi stupac pohranjuje ključeve, drugi nazive, a ostali pridružene datoteke .....	39

Slika 40: Forma aplikacije nakon promjena.....	40
Slika 41: Kartica "Prikaz podataka i unos značajki".....	41
Slika 42: Veze podataka i datoteka u matrici.....	42
Slika 43: Veze među podacima u matrici.....	42



## SAŽETAK

U ovom radu implementirana je aplikacija, kojoj je osnovna funkcija da omogućí hijerarhijski prikaz podataka, manipulaciju njima i datotekama te zapis u obliku matrice. Korišten je programski jezik Visual Basic „for applications“ i MS Excel. Excel raspolaže objektima specifičnim za tablične prikaze, kao što su radna knjiga, radni list i ćelija. Ćelija je objekt na koji se veže najviše metoda i svojstava koje VBA nudi.

Hijerarhijski prikaz podataka realizira se pomoću TreeView kontrole. Aplikacija se sastoji od unaprijed uređenih radnih listova i forme koja ima dvije kartice. Prva kartica ima ulogu isključivo prikaza TreeView strukture, dodjeljivanja značajki odabranog podatka te prikaz matrica.

Druga kartica omogućuje manipulaciju podacima i datotekama. Kartica u gornjem desnom dijelu sadrži kontrole da manipulaciju radnim listovima i kontrolu za prikaz radnih listova koji su bitni za pravilno učitavanje TreeView kontrole. TreeView kontrola za svoju implementaciju zahtijeva jedinstvene ključeve svojih članova, čvorova (*Nodes*). U ovoj aplikaciji ključevi su kreirani kao skup znakova koji čini ime radnog lista na kojem se podatak nalazi i vrijednost koja se nalazi unutar ćelije, odnosno ime podatka.

Kada se podaci učitaju i hijerarhijski prikažu, moguće je njima manipulirati. To se ostvaruje preko gumbi za dodavanje čvorova nulte razine, dodavanje čvorova na istoj razini, dodavanje podređenog čvora te gumba za uklanjanje čvora.

Za selektirani podatak korisnik može dodavati mu datoteke ili ih ukloniti. Dodavanje datoteka realizirano je pomoću dijaloškog okvira *Open Dialog*.

Svaku promjenu na formi, moguće je spremiti. Spremanjem datoteka, one se zapisuju u odgovarajući radni list i u mapu namijenjenu za njihovu pohranu. Spremanjem promjena na TreeView kontroli, podaci se zapisuju u radne listove prema zadanoj strukturi.

Matricama je moguće definirati i pridružiti veze.

U priloženom kodu nalaze se komentari koji opisuju korištene procedure.

Ključne riječi:

VBA, TreeView, struktura, upute



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



Središnje povjerenstvo za završne i diplomske ispite  
 Povjerenstvo za završne ispite studija strojarstva za smjerove:  
 procesno-energetski, konstrukcijski, brodstrojarski i inženjersko modeliranje i računalne simulacije

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

## ZAVRŠNI ZADATAK

Student: **Lorena Vlah**

Mat. br.: 0036464647

Naslov rada na hrvatskom jeziku: **RAZVOJ PROGRAMSKE APLIKACIJE ZA VIZUALIZACIJU RELACIJA**

Naslov rada na engleskom jeziku: **DEVELOPMENT OF SOFTWARE TOOL FOR RELATION VISUALIZATION**

Opis zadatka:

Proces razvoja vrlo složenih proizvoda i sustava na kojem radi veliki broj ljudi u dugotrajnim projektima iziskuje upotrebu složenih metoda prikaza i manipulacije relacijama između inženjerskih objekata. Inženjerskim objektima pri tome smatramo dokumentaciju (pretežno u digitalnom obliku), 3D modele, fizičke komponente i podsustave, zahtjeve, procese, zadatke, itd. Za prikaz relacija između inženjerskih objekata iz navedenih domena danas se koriste matricne metode. U radu je potrebno razviti programsku aplikaciju koja će omogućiti kreiranje i vizualizaciju relacija između objekata koji su hijerarhijski strukturirani (npr. struktura komponenti proizvoda) i datoteka koje čine skup dokumentacije o proizvodu. Aplikaciju treba realizirati korištenjem Microsoft Excel-a i pripadne verzije Visual Basic-a „for applications“ (VBA).

Programski sustav treba imati sljedeća sučelja odnosno mogućnosti:

1. Unos i manipulaciju objektima (elementima) koji su hijerarhijski strukturirani – realizirati korištenjem tzv. „tree view“ kontrole;
2. Unos i manipulaciju imenima i vezama na datoteke direktno iz strukture direktorija u radni list;
3. Postavljanje i ažuriranje relacija između objekata hijerarhijske strukture i datoteka.

Programski kod i primijenjene metode rješavanja pojedinih segmenata sustava treba detaljno dokumentirati te napraviti upute za korištenje razvijenog programskog sustava.


U radu navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:  
 25. studenog 2015.


Rok predaje rada:  
 1. rok: 25. veljače 2016.  
 2. rok (izvanredni): 20. lipnja 2016.  
 3. rok: 17. rujna 2016.

Predvideni datumi obrane:  
 1. rok: 29.2., 02. i 03.03. 2016.  
 2. rok (izvanredni): 30. 06. 2016.  
 3. rok: 19., 20. i 21. 09. 2016.

Zadatak zadao:

  
 Prof. dr. sc. Neven Pavković

Predsjednik Povjerenstva:

  
 Prof. dr. sc. Igor Balen

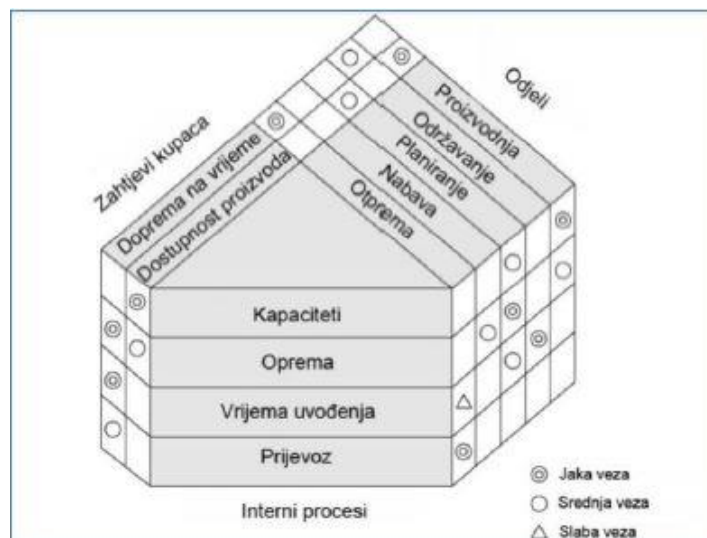
## 1. Uvod

### 1.1 Inženjerski objekti i datoteke

U razvoju složenih proizvoda sudjeluje mnogo ljudi specijaliziranih za određena područja kao što su management i uprava, razvojni inženjeri zaduženi za idejne koncepte, konstrukciju i potpunost tehničke dokumentacije novog proizvoda, tim zadužen za nabavu materijala i standardnih dijelova, inženjeri zaduženi za planiranje i praćenje proizvodnje i drugi. Management i uprava pritom mora voditi računa o budžetu, organizaciji i raspodjeli posla, kontaktu s partnerima, suradnji s kooperacijskim tvrtkama, željama i zahtjevima kupaca te nadzoru svih uključenih u proces nastanka proizvoda. Svi ti nabrojani procesi i zadatci su inženjerski objekti vezani za management i upravu a pripadni podaci pohranjuju se u obliku raznih datoteka. Objekti s kojima rade razvojni inženjeri su postojeći konkurentski proizvodi, funkcije koje proizvod mora ispuniti, skice, koncepti, 3D modeli i sklopovi, tehnička dokumentacija i drugi. Tim objektima također je pridružen niz datoteka. Tim zadužen za nabavu materijala i standardnih dijelova šalje upit na više proizvođača, prikuplja ponude te ih analizira prema cijeni, roku isporuke itd. U svim fazama projekta pojavljuje se jako puno inženjerskih objekata s pripadajućim datotekama koji su u međuovisnosti te se javlja potreba za prikazom njihovih međusobnih odnosa. Objekte i pripadne datoteke potrebno je vizualno prikazati u formi raznih dijagrama matričnog ili hijerarhijskog tipa te omogućiti da se dinamički mijenjaju njihove veze.

### 1.2 Matrične metode za prikaz relacija između inženjerskih objekata

Tijekom razvoja proizvoda inženjeri koriste razne metode koje im služe kao pomagala u planiranju, analizi, koncipiranju, vrednovanju, konstrukcijskoj razradi, planiranju proizvodnje, nabavi materijala te predviđanju ergonomije, eksploatacije i samog recikliranja novog proizvoda. Matrični prikaz omogućuje pojedincu ili timu da na vizualni način utvrdi uzročno-posljedične veze odabраниh kriterija, zahtjeva ili ciljeva, te odredi prioritete u rješavanju problema. Matrični dijagram može prikazivati odnos između dvije, tri, ili četiri skupine informacija. Neki od primjena upotrebe matričnog dijagrama su kad želimo odrediti distribuciju odgovornosti za zadatke među skupinom ljudi (matrica odgovornosti) ili formiranje matričnog prikaza prilikom povezivanja zahtjeva kupaca na elementima procesa. U dosadašnjem studiju pomoću matrice utvrđivali smo koji problemi utječu na koji proizvod ili dio opreme. Matricom se mogu dobro prikazati uzročno-posljedične veze, a za njezino kreiranje potrebni su određeni programski alati.



Slika 1: Matrični dijagram

DSM metoda (*Design structure matrix*) je matrični prikaz relacija. Ova metoda razlikuje se od tradicionalnih metoda za upravljanje projektima jer je usmjerena na jednostavno predočavanje protoka informacija, te na efekte koje ti tokovi imaju na tijek izvođenja cjelokupnog projekta.

U ovom radu matrica je kreirana u jednom radnom listu Microsoft Excela, a na listu relacije moguće je definirati veze. U stupcu matrice nalaze se objekti iz TreeView kontrole koja se prikazuje preko aplikacije, a u retku pripadne datoteke. Korisnik, nakon što je matrica prikazana, može postaviti veze među njezinim članovima. Na datoteke prikazane u matrici je postavljen hiperlink, te se one klikom na polje u kojem se nalaze, otvaraju.

### 1.3 Programiranje u Visual Basic-u „for applications“

#### 1.3.1 Excel aplikacije

Excel je dio programskog paketa Microsoft Office namijenjen unosu, obradi i ispisu većih količina brojčanih podataka smještenih u tablice te zbog toga ima široku primjenu u inženjerstvu. Možemo ga koristiti i kao jednostavnu bazu podataka, za razne proračune te prikaz podataka u obliku grafikona. Do sada je izašlo već mnogo inačica, a aplikacija koju sam izradila u završnom radu implementirana je u verziji Excel 2010. Od verzije do verzije neke kontrole i objekti imaju nove mogućnosti i tehnologije. Sve podržavaju standardni programski jezik Visual Basic „for application“, a razlikuju se uglavnom u raspoloživim objektima, njihovim svojstvima i metodama.

Glavni Excel objekti su radna knjiga, njezini radni listovi te ćelije. Radne listove upotrebljavamo za popisivanje, organiziranje i analizu podataka koji opisuju neku određenu pojavu ili problem, a radnu knjigu čini više radnih listova spremljenih u jednu datoteku.

Excel aplikacije služe kao pomagalo ljudima da efikasno obavljaju svoje poslove, pružaju im mogućnost da što jednostavnije prikažu i analiziraju podatke te njima

manipuliraju. Aplikacije u Excelu se kreiraju na temeljima već postojeće funkcionalnosti Excela, a programski kod piše se u programskom jeziku Visual Basic koji je specijalno dizajniran za brzo projektiranje aplikacija.

Tijekom izrade računalnih programa i aplikacija, potrebno je voditi računa o velikoj količini informacija i pravila neophodnih za ispravno funkcioniranje programa. Proces izrade programa dijelimo u slijedeće faze:

1. Analiza problema
2. Postavljanje modela
3. Izrada algoritma
4. Izrada dijagrama toka
5. Kodiranje programa
6. Prevođenje programa
7. Testiranje programa
8. Dokumentiranje programa
9. Eksploatacija programa

### 1.3.2 Visual Basic Editor

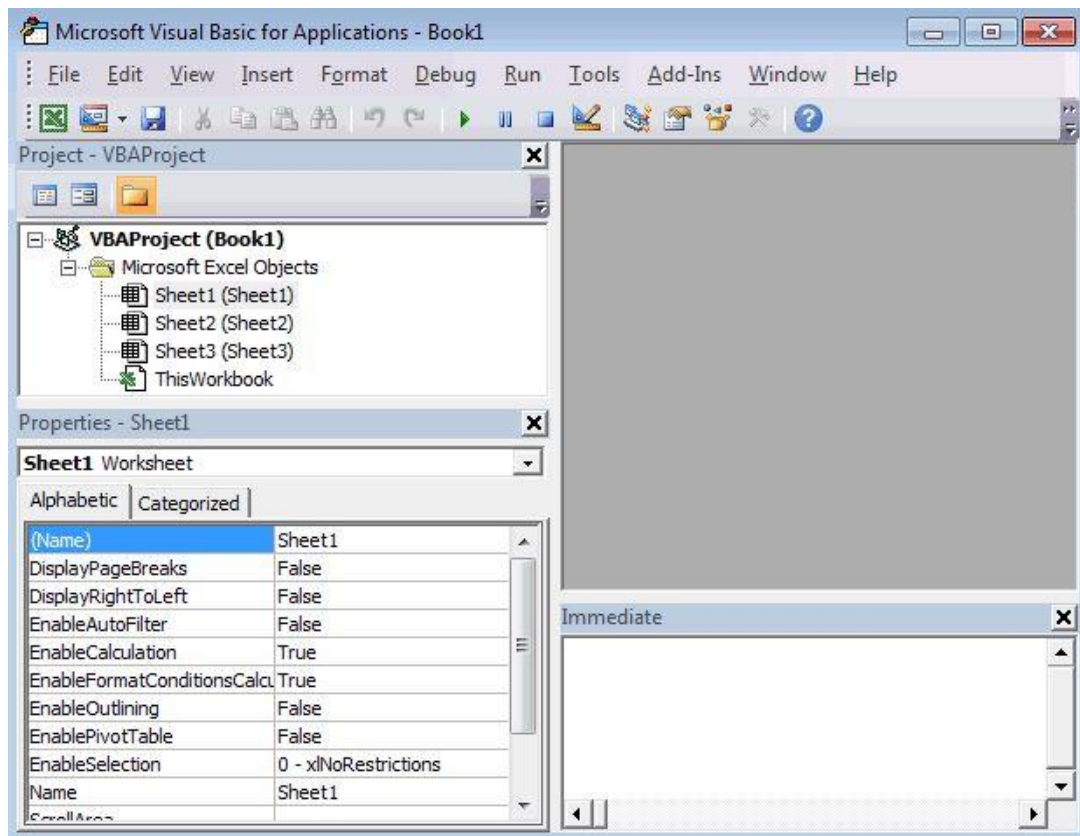
Kod se piše u Visual Basic Editoru (VBE). VBE editor se sastoji od tri prozora a to su prozor *Project*, prozor *Properties* i prozor *Immediate*. Svaku radnu knjigu Project Explorer tretira kao zaseban projekt. Nakon što se u radnoj knjizi kreira određeni broj formi, modula ili klasnih modula, one će biti vidljive u Project Exploreru.

Svaki radni list u nekoj radnoj knjizi biti će tretiran kao zaseban Excel objekt. Forme predstavljaju korisničke obrasce, koji mogu biti dodani u projekt. Moduli predstavljaju spremnike za procedure koje smo napisali. Kada krećemo sa pisanjem programskog koda, prvo stavljamo odgovarajući modul u projekt. Klasni moduli omogućavaju projektiranje vlastitih, prilagođenih objekata. Moduli i forme koje su pridružene nekom objektu posjeduju vlastiti prozor u kojem se vidi odgovarajući VBA kod.

Prozor pod nazivom *Properties* uglavnom se koristi za objekte tipa Microsoft Excel *Objects* i *Forms*. Ovisno o tome što smo selektirali u Project Exploreru ili prilikom dizajniranja neke forme, unutar prozora *Properties* biti će prikazan određeni broj svojstava, koja su pridružena tome objektu i koje možemo analizirati i uređivati tj, mijenjati im sadržaj.

Prozor *Immediate* možemo korisno upotrijebiti za prikazivanje izlaznih rezultata u toku procesa projektiranja, a kada se programski kod nalazi u *Break* režimu možemo pomoću njega vršiti kontrolu stanja objekata. Možemo ga i koristiti za izvršavanje pojedinačnih iskaza (*statements*).

Od ostalih osobina VBE Editora treba izdvojiti Auto Syntax Check, Auto List Members i Auto Quick Info. Auto Syntax Check tokom samog unošenja programskog koda automatski provjerava da li je taj kod u suglasnosti sa zahtijevanom sintaksom programskog jezika Visual Basic „*for applications*“. Opcija Auto List Members nudi popis svojstava i metoda pojedinog objekta (ovisno o tipu), npr. ona zna da objekt radnog lista posjeduje svojstvo *Name*, metodu *Delete*, svojstvo *Visible* i sva ostala svojstva i metode koje su karakteristične za radne listove. Ta opcija na ekranu prikazuje popis raspoloživih svojstava i metoda u toku samog unošenja koda. Karakteristika Auto Quick Info omogućava prikazivanje informacija o funkcijama i njihovim parametrima (njihov pravilan redoslijed i dozvoljeni tip podataka za svaki od parametara).

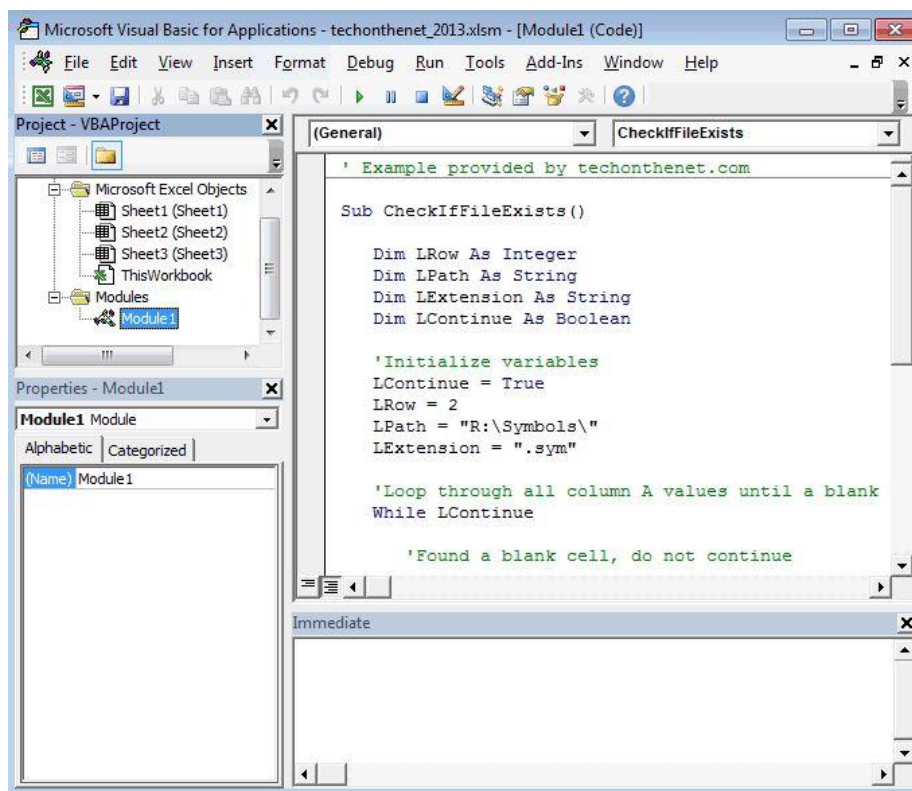


Slika 1: Visual Basic Editor

### 1.3.3 Code Window prozor

Padajuća lista u gornjem desnom kutu je lista objekata (*Object List*) ,a sadrži sve objekte koji se mogu prikazati u Code Window prozoru. Ako se radi o nekom standardnom modulu jedina opcija biti će *General*, a kod komponenata tipa *UserForm* za svaki kontrolni element na *UserForm*-i postojati će po jedna opcija na listi objekata.

Padajuća lista u gornjem desnom kutu je Procedure List. Na njoj će biti prikazane sve procedure iz objekta koji je trenutno selektiran na *Object* listi. Prilikom selektiranja na obje spomenute liste, u Code Window prozoru biti će prikazane odgovarajuće procedure.



Slika 2: Code Window

### 1.3.4 Tipovi podataka, varijable i područje varijabli

Programski kod sastoji se od varijabla, operatora i naredbi. Varijable predstavljaju objekte koje programski kod obrađuje. Primjer varijable u VBA može biti bilo koji broj nad kojim je potrebno izvršiti određene matematičke operacije, tekstualni *string* koji je potrebno urediti, neki radni list, radna knjiga, raspon itd. To znači da na programskoj razini možemo manipulirati svim postojećim objektima Excela s kojima i inače radimo. Za razliku od većine drugih programskih jezika, u VBA je za upotrebu neke varijable dovoljno u programskom kodu navesti samo njezin naziv (ako nije upotrijebljen *Option Explicit*), bez prethodnog deklariranja iste.

Većina tipova podataka korištenih u Visual Basic-u dostupni su i u VBA. Najčešće korištene su *Integer*, *Single Long*, *String*, *Boolean*, *Object* i *Variant*. Prilikom implementacije ove aplikacije koristila sam i *Error* i *Nothing* tipove varijabli. Varijabla tipa *Error* ukazuje da je, unutar neke procedure, došlo do pojave aplikacijski definirane pogreške. Postavljanjem objektne varijable na vrijednost *Nothing* oslobađa se prostor u radnoj memoriji koji je taj objekt zauzimao te ostaje više prostora za druge namjene. Značajnu primjenu ima varijabla *Variant* (promjenljivi tipovi podataka) zbog toga što u sebi može sadržati i matrice (*arrays*) te se koristi za učitavanje podataka s radnog lista. Ako za neku varijablu eksplicitno ne definiramo tip podatka, njoj će automatski biti dodijeljen tip *Variant*.

Područje varijabli predstavlja područje njene vidljivosti i određeno je položajem iskaza o deklaraciji te varijable, kao i upotrebom ključnih riječi *Public*, odnosno *Private*. Prilikom određivanja područja bilo koje varijable, na raspolaganju su nam tri mogućnosti: područje procedure, područje modula i globalno područje. Varijabla na razini procedure možemo

pristupiti isključivo preko programskog koda koji je smješten u istoj proceduri u kojoj su te varijable deklarirane. Varijable deklarirane na početku nekog modula mogu biti varijable na razini modula ili varijable na globalnoj razini. Kod globalnih varijabli iskaz *Dim* zamjenjujemo sa *Public* te se one mogu upotrijebiti pomoću bilo kojeg programskog koda unutar projekta.

### 1.3.5 Usmjeravanje toka programa uz pomoć naredbi

Naredne su pokretačka snaga VBA programskog jezika kojima, između ostalog, možemo u svojim programima implementirati petlje i donositi logičke odluke. Programske petlje koriste se radi višestrukog i uzastopnog izvršavanja određenog dijela programskog koda. Postoje fiksne petlje, čiji je broj izvršavanja poznat i konačan, te promjenljive petlje, koje su oslonjene na neki logički izraz na osnovu čijeg izračunavanja program donosi odluku da li izvršavanje petlje treba nastaviti ili ne. Primjer fiksnih programskih petlji često korištenih u VBA programiranju su *For...Next* te *For...Each*, a *Do...Loop* petlja može imati više varijanti te je zbog toga nazivamo promjenljivom petljom.

Za grananje programskog koda koristimo *If...Then* iskaz. Grananje je proces selektivnog izvršavanja određenih dijelova koda u ovisnosti od vrijednosti danog izraza. Najčešće se koristi *If...Then* iskaz koji ispituje jedan jedini izraz *i*, ukoliko je on istinit, izvršava jednu jedinu naredbu. Grananje programskog koda za više od dvije različite grane ostvarujemo pomoću *If...Then...ElseIf* iskaza ili korištenjem *Select Case* iskaza.

Operatori predstavljaju elemente programskog jezika, koje možemo upotrijebiti kako za uspoređivanje tako i za promjenu vrijednosti drugih programskih elemenata na određeni način. Matematički operatori primjenjuju se za izvršavanje matematičkih operacija. Operatori uspoređivanja u kombinaciji sa *And*, *Is*, *Not* i *Or* koriste se kako bismo u programski kod implementirali logičke zakone s ciljem određivanja trenutka u kome treba prekinuti izvršavanje neke petlje ili kao sastavni dio *If Then* iskaza.

### 1.3.6 Korištenje matrica

Matrica (*array*) je bilo koja grupa indeksiranih elemenata sa istim tipom podataka. Uz pomoć matrica možemo deklarirati jednu jedinu varijablu, koja u sebi sadrži veliki broj pojedinačnih vrijednosti kojima pristupamo putem indeksa. Pri deklariranju matrice potrebno je definirati i njezinu dimenziju. Ukoliko eksplicitno ne definiramo opseg polja, biti će kreirana dinamička (*dynamic*) matrica, čije se dimenzije mogu mijenjati u toku samog izvršavanja programskog koda. Prvi element matrice je označen s indeksom 0. Upotrebom iskaza *Option Base 1* na početku nekog modula, prvi element VBA matrice početi će indeksom 1. Za definiranje raspona koristi i pomoćna riječ *To* unutar deklaracije matrice.

### 1.3.7 Procedure

Svaka procedura predstavlja skup iskaza, koji izvršavaju jedan ili više zadataka. Razlikujemo dvije vrste procedura, potprograme (*subroutine*) i funkcije. Osnovna razlika je u tome što funkcije mogu vratiti neku vrijednost procesu koji ih je pozvao. Potprogrami predstavljaju najmanji element programa koji uopće može biti izvršen i ne moraju eksplicitno vraćati nikakvu konkretnu vrijednost. Jednom kreirani potprogrami mogu biti pozvani ili izvršavani od strane drugih potprograma navođenjem naziva odgovarajućeg potprograma na



željenoj lokaciji unutar pozivajućeg potprograma. U tu svrhu možemo upotrijebiti iskaz *Call*, ali u VBA ona nije obavezna. Funkcija izvodi određeni zadatak i vraća vrijednost proceduri iz koje je ta funkcija pozvana. Prilikom deklariranja funkcija potrebno je deklarirati i tip podatka koji će biti dodijeljen njenoj izlaznoj vrijednosti. Ako ne navedemo tip podatka, funkcija će vraćati varijablu tipa *Variant*.

### 1.3.8 Objekti, metode, svojstva i događaji

Objekt je spoj programskog koda i podataka povezanih u jednu cjelinu. Oni predstavljaju stavke (*items*) kojima možemo manipulirati uz pomoć programskog koda. Objekt sadrži u sebi metode i svojstva koji su članovi (*members*) datog objekta. Najznačajniji objekti u stvaranju Windows aplikacije su obrasci (*forms*) i kontrole. Svojstva i polja predstavljaju informacije smještene u objekt. Svojstva (*properties*) predstavljaju karakteristike kojima se opisuje pripadni objekt. Postupci ili metode (*methods*) su akcije koje neki objekt može poduzeti, izvršiti. Objekt može reagirati na neke događaje kao što su klik, pomicanje miša ili pritisak na tipku tipkovnice. Način reakcije određuje se programskim kodom postupka (events) vezanog za taj događaj.

## 1.4 Objekti u Microsoft Excelu

### 1.4.1 Excelov objektni model

Glavni Excel objekti su Excel aplikacija u cjelini (*Application object*), radna knjiga (*Workbook object*), radni list (*Worksheet object*) te raspon (*Range object*).

Svi članovi (*Members*) objekta *Application* mogu se podijeliti u četiri grupe, a to su: članovi orijentirani na sadržaj koji se prikazuje na ekranu, članovi namijenjeni bržem i jednostavnijem radu u Excelu (*ActiveCell*, *ThisWorkbook*, *Selection* Objekt, i drugi), članovi za operacije s datotekama te članovi koji sadrže informacije o sistemu. Značajne metode su *GetOpenFileName* i *GetSaveAsFileName* koje omogućavaju brz pristup dijaloškim okvirima *Open* i *Save As*, koji se koriste u gotovo svim Windows aplikacijama.

*Workbook* objekt predstavlja jednu radnu knjigu. Osim svojstva i metoda za mnoge objekte postoji definiran skup akcija na koje ti objekti daju odgovor, jer su tako projektirani. Sve akcije koje određeni objekt prepoznaje nazivamo događajima (*events*). Kao primjer navela bih objekt *Workbook* koji sadrži događaj *Open*. Kad god dođe do otvaranja neke radne knjige, VBA će provjeriti da li u programskom kodu postoji dio koji je pridružen događaju *Open* i automatski ga izvršiti. Događajima možemo lako pristupiti tako da unutar VBA Editora s padajuće liste *Object* odaberemo opciju *Workbook*, te s padajuće liste *Procedures/Events* odaberemo događaj za koji želimo pisati programski kod. Kolekcijski objekt *Workbooks* predstavlja sve trenutno otvorene radne knjige u Excelu i posjeduje metode pomoću kojih se može otvoriti nova radna knjiga (*Add* metoda), otvoriti neka od postojećih radnih knjiga ili zatvoriti bilo koja od radnih knjiga koje su trenutno otvorene. Svojstva objekta *Workbooks* su *Item*, koji koristimo radi pristupa nekoj konkretnoj stavki unutar kolekcije, *Count*, koje vraća ukupan broj stavki u danoj kolekciji, *Path*, *Saved*, *ReadOnly* i mnogi drugi.

Objekt *Worksheet* upotrebljava se često, a u najvećem broju slučajeva koristimo ga kao *Parent* svojstvo da bismo došli do nekog raspona (*Range object*). *Worksheets* objekt

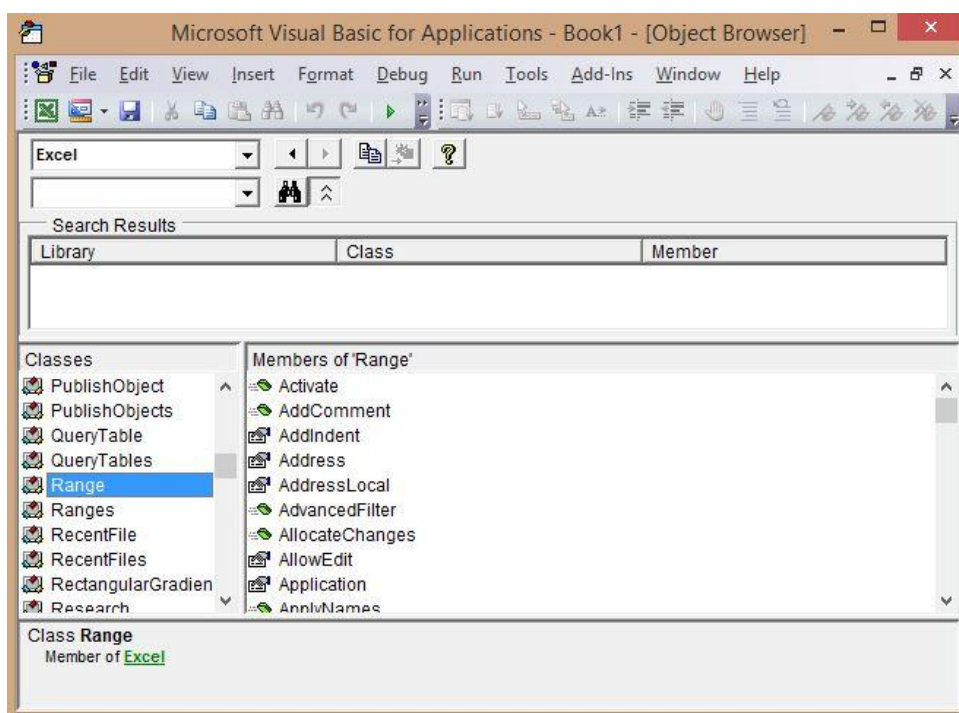
predstavlja jedno od svojstava objekta *Workbook* te ga ne možemo upotrijebiti ako prethodno ne navedemo i pripadni *Workbook*. Određenom radnom listu možemo pristupiti i tako da mu unutar VBA Editora, u prozoru *Properties*, definiramo kodni naziv (*Name*). Uobičajeni zadatci koji se rade nad objektom radnog lista su njihovo dodavanje, brisanje, kopiranje, premještanje, izmjena svojstva vidljivosti, postavljanje i uklanjanje zaštite radnog lista te se za realizaciju istog koristi niz svojstava, metoda i događaja pridruženih radnim listovima. Neke od tih su *Type*, *Count*, *Move*, *Copy*, *Activate*, *Change*, *FollowHyperlink*, itd.

Najvažniji objekt po pitanju programskog upravljanja Excelom je raspon (*Range*) koji sadrži puno više svojstava i metoda od objekata koje sam dosad spomenula. Osnovne radnje nad objektom *Range* su kreiranje referenci prema određenim opsezima ćelija, navigacija po radnom listu i upravljanje ulazno/izlaznim podacima. Opseg (*Range*) može predstavljati samo jednu, ili proizvoljan broj ćelija na radnom listu. Do željenog raspona možemo doći na mnogo načina. Neki od njih su primjena svojstva *ActiveCell*, *Range* ili *Selection*, objekta *Application* ili primjena svojstva *Cells*, *Columns*, *Range* ili *Rows*, objekta *Worksheet*. Svojstvo *Cells* vraća opseg koji predstavlja sve ćelije danog radnog lista, a određenoj ćeliji pristupamo *dodavajući* svojstvo *Item*. Sve stavke unutar kolekcijskog objekta *Cells* raspoređene su u obliku jedne dvodimenzionalne matrice, čime je omogućeno definiranje pojedinačnih ćelija, korištenjem odgovarajućih oznaka reda i stupca. Neka od svojstava objekta *Range* su *Value*, *Address*, *Font*, *HorizontalAlignment*, *Color*, *Offset*, *End* i druga. Pokretljivost podataka i manipulacija istim postiže se korištenjem metoda *Cut*, *Copy*, *Paste*, *Sort*, *Find* i *Replace*.

Visual Basic „*for Applications*“ sadrži veliki broj objekata, od objekata za vizualni dojam (npr. *Font*, *Interior*, *Border*) i objekata vezanih uz grafikone do objekata koji su osnovni elementi kontrola kao što su *Button* ili *Node* korišten u *TreeView* kontroli.

## 1.4.2 Object Browser

Tokom projektiranja Excel aplikacija u VBA, programerska interakcija sa Excelom sastoji se od manipulacije ili korištenja različitih Excelovih objekata. Excel nudi ogroman broj objekata kojima možemo manipulirati. Jedan od načina na koji možemo vidjeti koje sve objekte imamo na raspolaganju je koristeći tzv. *Object Browser*. Možemo ga upotrijebiti za pregled informacija o raspoloživim bibliotekama objekata, za pregled neke specifične biblioteke objekata, kao i za pregled biblioteke objekata iz našeg vlastitog projekta. Pod informacijama se smatra lista svih objekata u određenoj biblioteci, kao i lista svih članova za svaki od tih objekata. Kada odaberemo datoteku Excel možemo vidjeti npr. za objekt *Worksheet* listu svih članova, ikone koje predstavljaju metode (*methods*), događaje (*events*) te svojstva (*properties*).

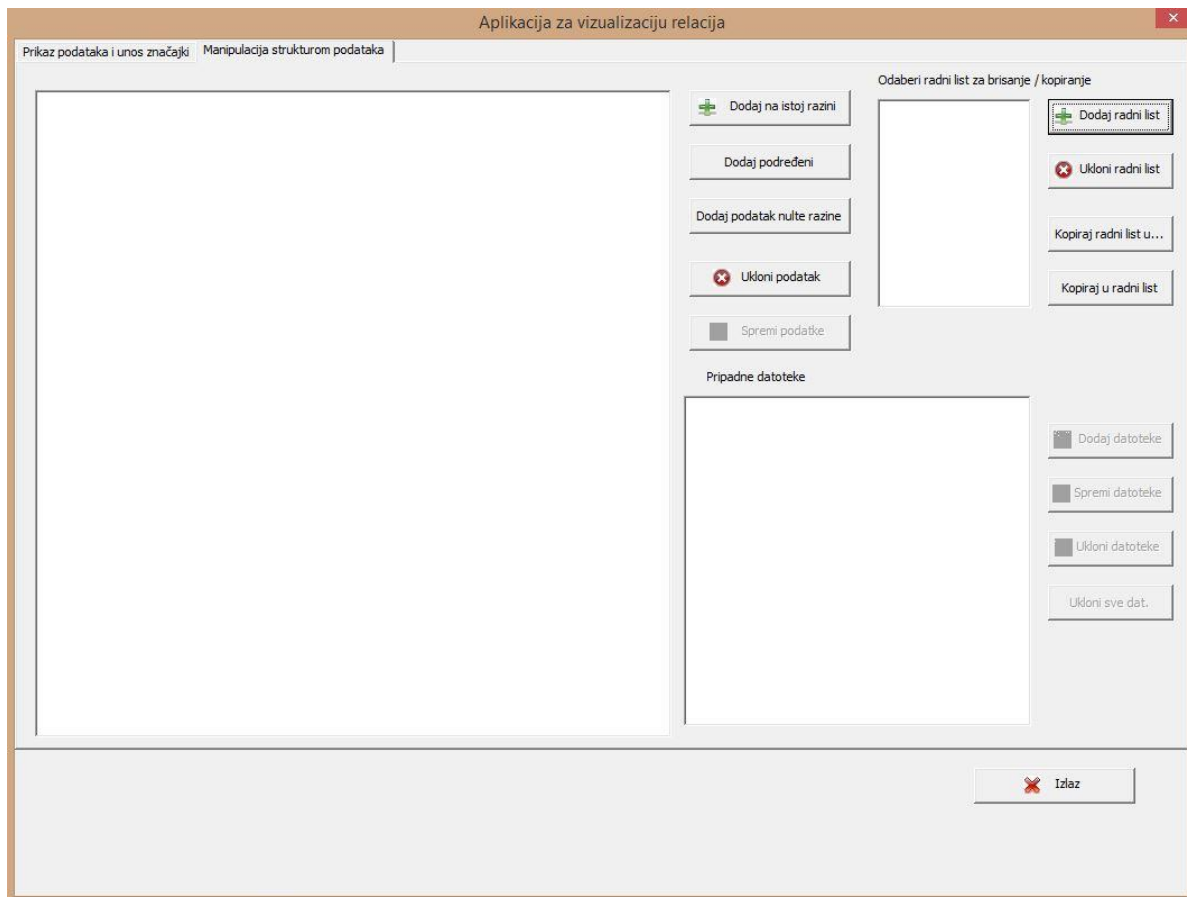


Slika 3: Object Browser

### 1.4.3 Forma kao korisničko sučelje u Excelu

Svaka aplikacija zahtjeva neku vrstu korisničkog sučelja (*Interfaces*). Jedna od prednosti upotrebe Excela kao razvojne platforme za izradu aplikacija je ta što se aplikacije projektiraju na temeljima već postojeće Excelove funkcionalnosti. Korisničko sučelje omogućava krajnjim korisnicima da po želji usmjeravaju tok izvršavanja neke aplikacije. Zadatci koje korisničko sučelje mora izvršiti su prikupljanje ulaznih podataka (*input*) od korisnika, upravljanje postavkama (*Settings*) i konfiguracijom aplikacije, prikazivanje izlaznih rezultata (*output*) i omogućavanje kretanja (navigacije) po aplikaciji. Korisničko sučelje može biti zasnovano na radnom listu (*worksheet-based user interface*), pomoću komandnih paleta i pomoću korisničkih formi. Kontrole formi na radnom listu su one kontrole koje su kreirane upotrebom palete alata *Forms* (*View->Toolbars->Forms*).

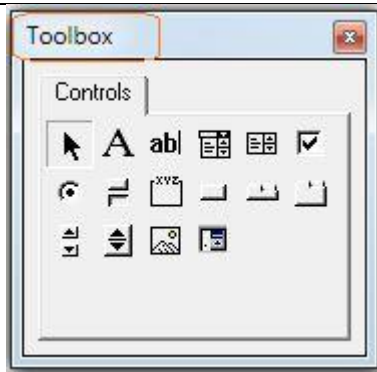
Kao korisničko sučelje za prikaz aplikacije za vizualizaciju relacija korištena je forma. Forma ili obrazac je u početku pravokutni dio zaslona, koji se može izmijeniti po potrebi, a služi za prikaz ili unos podataka. Svaki novi projekt otvara se s jednom formom koja nosi naziv *Form1*. Većina svojstava forme vezana je za namještanje njezina izgleda, a postavljanje tih svojstava omogućuje prozor *Properties*. Vrijednost svojstava može se odrediti i programskim kodom, što znači da će svojstva biti namještena pri izvođenju aplikacije (*Run Time*). To su npr. određivanje oblika okvira, boja pozadine i drugi.



Slika 4: Forma aplikacije za vizualizaciju relacija

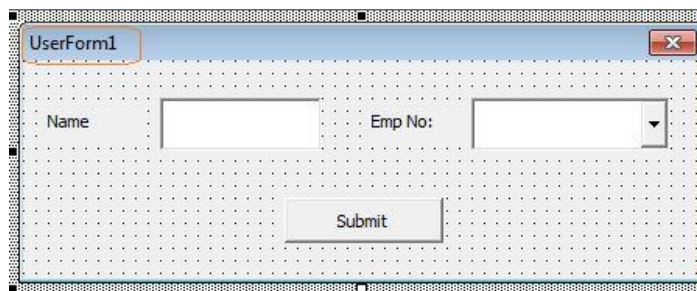
#### 1.4.4 Osnovne kontrole u VBA

Svaki objekt definiran je svojom klasom (*class*). Klasa može poslužiti za stvaranje proizvoljnog broja objekata. U Windows aplikaciji kontrola u alatnom okviru (*toolbox*) je klasa. Njezinim povlačenjem na obrazac (*form*) stvara se objekt koji je primjerak te klase. Klasa je opisana preko svojih svojstava (*properties*), polja (*fields*), postupaka (*methods*) i događaja (*events*).



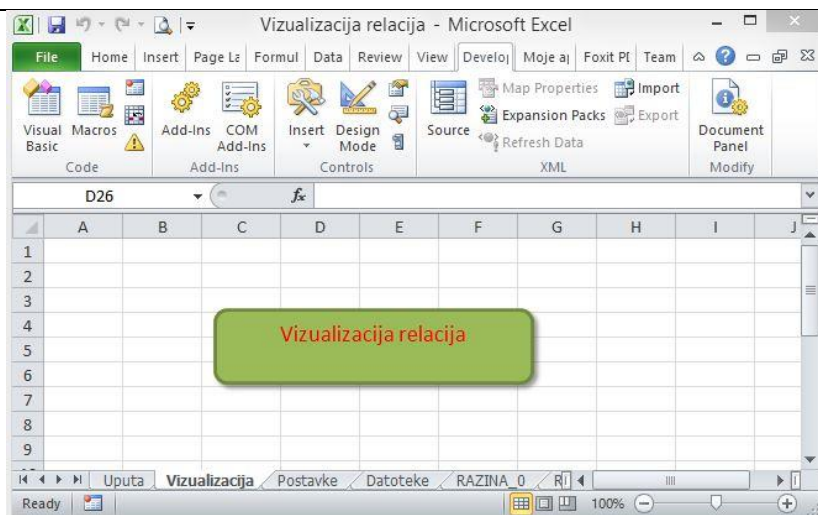
Slika 5: Okvir Toolbox sa osnovnim kontrolama

Kontrole za prikazivanje i uređivanje teksta su *Label* i *TextBox*. Natpis (*Label*) koristi se za prikaz teksta, a tekstni okvir (*TextBox*) za upis teksta. Najznačajnije svojstvo za obje kontrole je *Text*. Vrijednost mu odgovara prikazanom ili upisanom sadržaju. Najčešće se pojavljuju u paru pri čemu sadržaj svojstva *Text* kontrole *Label* ukazuje na sadržaj koji treba biti upisan u tekstni okvir.



Slika 6: Kontrole *Label* i *TextBox*

Naredbeni gumb (*Button*) upotrebljava se da bi se klikom na njega zadala neka naredba, odnosno poduzela neka akcija. Zbog toga se programski kod koji bi trebao odrediti tu akciju nalazi u događaju *Click* ove kontrole. Gumbi (*Buttons*) predstavljaju najčešće korištenu vrstu kontrola. Formatiranje gumba vrši se opcijama koje su smještene na kartici *Properties* unutar okvira za dijalog *Format Control*.



Slika 7: Gumb za pokretanje aplikacije Vizualizacija relacija

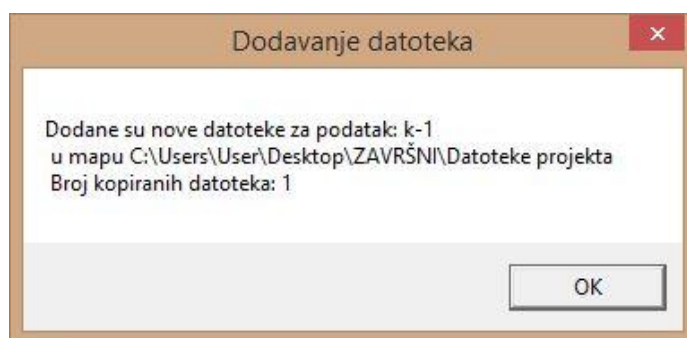
Odabir s popisa ostvaruje se pomoću popisnog okvira (*ListBox*) ili kombiniranog okvira (*ComboBox*). Svrha popisnog okvira je omogućavanje odabira nekog objekta s popisa. Kombinirani okvir sastoji se iz dva dijela: tekstualnog okvira i padajućeg popisa. Zbog toga kombinirani okvir, za razliku od popisnog, ima svojstvo *Text*. Svojstvo tekst dopušta upis proizvoljnog teksta pri čemu popis s padajuće liste predlaže njegov sadržaj. Bez obzira na navedenu razliku obje kontrole se upotrebljavaju za izbor s njihova popisa. Zbog navedene sličnosti jedan dio svojstava im je isti. Najznačajnija svojstva odnose se na zbirku objekata (*Items*), odabrani objekt (*SelectedItem*) i njegov redni broj (*SelectedIndex*). Prvi objekt u zbirci ima redni broj (*index*) 0.

*MultiPage* je kontrola koja omogućuje kreiranje dijaloškog okvira s više kartica na kojima su druge kontrole. Najviše korištena svojstva *MultiPage* kontrole su *Style* i *Value*.

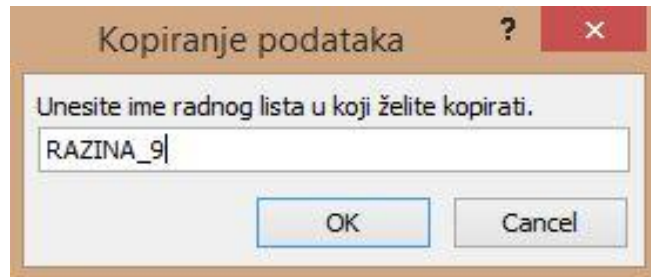
Neka svojstva, metode i događaji karakteristični su samo za određenu kontrolu i ona određuju njezinu namjenu i način uporabe, no puno je veći broj onih koji su identični za sve kontrole (npr. *Add* ili *Click*).

Na okvir *Toolbox* možemo, po potrebi postaviti i dodatne kontrole, tako da na nju kliknemo desnim klikom miša i odabremo opciju *Additional Controls*. Na taj način u okvir *Toolbox* dodala sam kontrolu *TreeView* koja omogućuje hijerarhijski prikaz podataka, što je i glavni zadatak ovog rada.

U aplikacijama se često koriste funkcije *MsgBox* i *InputBox*. *MsgBox* prikazuje poruku u dijaloškom okviru, a nakon što korisnik klikne na neki od ponuđenih gumba, ona vraća cijeli broj. Obavezan parametar funkcije je *Prompt* kao *string*, a optimalan je gumb, ili više njih te naslov (*Title*).



Funkcija *InputDialog* prikazuje upit u dijaloškom okviru, a nakon što korisnik unese tekst ili pritisne gumb, vraća *string* u kojem je pohranjen sadržaj upisan u tekstualni okvir. Obavezan parametar funkcije je *Prompt* kao *string*, a optimalni su *Title*, *Default Response*, *XPos* i *YPos* za usmjeravanje dijaloškog okvira na ekranu.



Slika 9:InputDialog

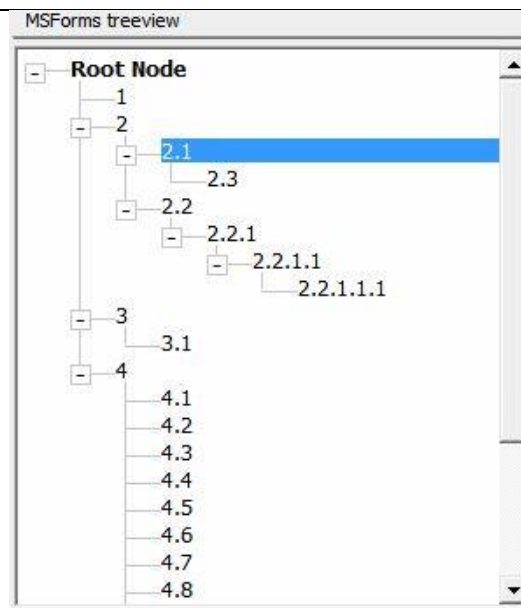
## 1.5 Treeview kontrola

### 1.5.1 Objekt čvor (Node)

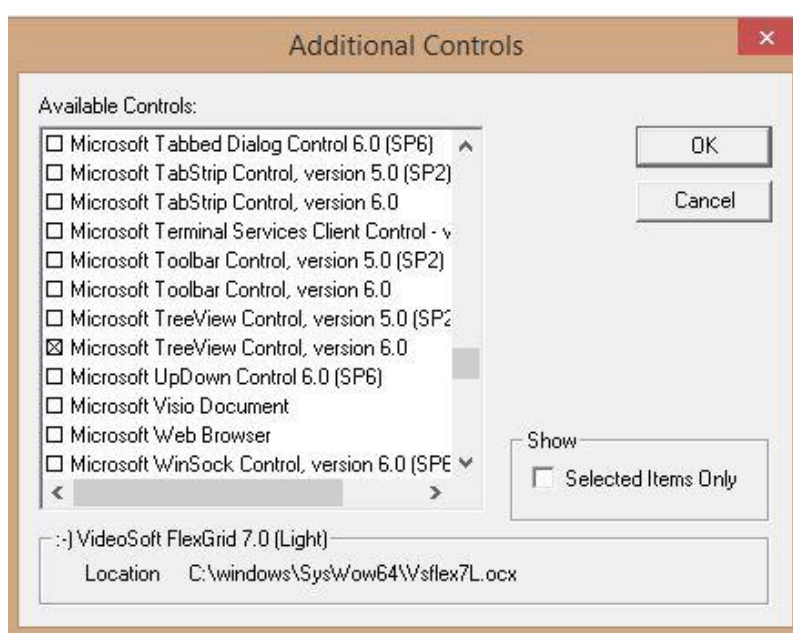
Zadatak TreeView kontrole je prikaz podataka koji su hijerarhijski povezani. Kontrola se sastoji od čvorova (*Nodes*). Ona omogućava korisnicima pregled podataka, odnosno objekata, i brzo utvrđivanje kako su ti objekti povezani.

Slika 10. prikazuje primjer TreeView kontrole prikazane u formi. Korijski čvor (*Root*) često je ime projekta te je tim čvorom definiran početak stabla. Možemo vidjeti da je nekoliko čvorova razvedeno ispod *Root Node*. Uvlaka i linije koje povezuju čvorove pokazuju da su čvorovi 1,2,3 i 4 tzv. *Child* („dijete“) čvorovi od korijskog čvora. Čvor 2 je roditeljski čvor čvorovima 2.1 i 2.2. *Child* čvorovi istog roditeljskog (*Parent*) čvora nazivaju se *Siblings*.

Prema početnim postavkama MS Excela kontrola TreeView ne nalazi se u okviru *Toolbox*, te je potrebno postaviti dodatne kontrole. Desnim klikom miša na paleti *Toolbox* potrebno je odabrati opciju *Additional Controls* i dodati TreeView kontrolu u *Toolbox*.



Slika 10: Primjer TreeView kontrole u MS formi





Članovi `TreeView` kontrole su čvorovi, tzv. *Node Object*, a sadržani su u kolekcijском objektu *Nodes*. Kao što se skup podataka pohranjuje kao zbirka u kontrolu *ListBox*, tako se i kolekcija čvorova pohranjuje u `TreeView` kontrolu. Do podataka koji se odnose na određeni čvor, član stabla, može se doći njegovim indeksom koristeći zapis:

```
TreeView1.Nodes(index)
```

gdje je „*index*“ broj koji identificira čvor u zbirci, ili tražeći željeni čvor preko vrijednosti ključa (*Key*) koristeći zapis:

```
TreeView1.Nodes("key")
```

gdje je „*key*“ *string* vrijednost *Key* svojstva traženog čvora.

## 1.5.2 Metode `TreeView` kontrole

Čvorove `TreeView` kontrole možemo dodavati i ukloniti iz strukture. Dodajemo ih koristeći metodu *Add* kolekcijског objekta *Nodes*. Sintaksa za dodavanje je sljedeća:

```
TreeView1.Nodes.Add(relative, relationship, key, text, image, selectedimage)
```

Svi argumenti *Add* metode su neobavezni. Argument *Relative* daje ključ ili indeks postojećeg čvora u stablu na kojeg će se novi dodati a, parametar *Relationship* kaže gdje će se novi čvor smjestiti u odnosu na *Relative* čvor. Ako *Relative* i *Relationship* argumenti nisu navedeni, novi čvor će se smjestiti nakon svih ostalih čvorova na toj razini.

Vrijednosti *Relationship* argumenta su sljedeće:

**-*twFirst* (0)** ,

Novi čvor se nalazi na istoj razini stabla kao i „*relative*“. Taj čvor će biti dodan kao prvi čvor te razine.

**-*twLast* (1)** ,

Novi čvor se nalazi na istoj razini stabla kao i „*relative*“ čvor i biti će dodan nakon zadnjeg postojećeg čvora na istoj razini kao i „*relative*“.

**-*twNext* (2)** ,

Novi čvor će biti postavljen na istoj razini u stablu kao i „*relative*“ čvor, odmah iza njega, slijedeći ga.

**-*twChild* (4)** ,

Novi čvor će biti *Child* čvor od „*relative*“ čvora.

Svojstvo ključa ( *Key property*) identificira novi čvor u stablu. Ako se unese kao argument, mora biti jedinstven niz znakova, što znači da se ne smije ponoviti ista vrijednost ključa za dva ili više čvorova. Ključ se koristi za dohvaćanje i traženje nekog čvora, ukoliko mu indeks nije poznat. Posljednja tri argumenta *Add* metode definiraju izgled novog čvora. Tekst koji se pojavljuje definiran je kao tekstualni argument i pripada tipu *string*.

### 1.5.3 Svojstva TreeView kontrole

Brojna svojstva za kontrolu *TreeView* i *Node* objekte definiraju izgled stabla i omogućuju pristup čvorovima unutar stabla. Mnogi od njih služe na navigaciju kroz stablo, a najčešće korišteni su:

#### **-Child**

Svojstvo *Child* vraća referencu na prvi *Child* čvora, kao u sljedećem primjeru:

```
Dim objNode as Node  
objNode = TreeView1.Nodes(1).Child
```

Varijabla *objNode* postaje jednaka prvom *Child* čvoru, odnosno čvoru s indeksom 1. Mogli smo postaviti reference izravno na *Child*:

```
TreeView1.Nodes(1).Child.Text = "Ovo je prvi čvor."
```

Ovaj dio koda mijenja tekst prvog *Child* čvora u odnosu na čvor s indeksom 1.

#### **-FirstSibling**

Vraća referencu na prvi *Sibling* čvor navedenog čvora. *FirstSibling* je čvor na istoj razini kao i navedeni čvor.

#### **-LastSibling**

Vraća referencu na posljednji *Sibling* čvor navedenog čvora. *LastSibling* je čvor na istoj razini kao i navedeni čvor.

#### **-Parent**

Vraća referencu na *Parent* čvor navedenog čvora.

#### **-Next**

Utvrđuje i vraća čvor neposredno nakon određenog čvora.

#### **-Previous**

Utvrđuje i vraća čvor koji prethodi navedenom čvoru.

#### **-Root**

Vraća korijenski čvor, najvišu razinu u stablu navedenog čvora.

#### **-SelectedItem**

Vraća referencu na čvor koji je trenutno odabran u *TreeView* kontroli.

Za kontrolu, ponašanje i izgled stabla mogu se koristiti i neka dodatna svojstva kao što su *Children*, *Selected*, *Expanded*, *FullPath*, *PathSeparator*, *LineStyle* i *Sorted*.

(Name)	frmDemo
BackColor	<input type="checkbox"/> &H8000005&
BorderColor	<input type="checkbox"/> &H80000012&
BorderStyle	0 - fmBorderStyleNone
Caption	
ControlTipText	
Cycle	0 - fmCycleAllForms
Enabled	True
Font	Tahoma
ForeColor	<input type="checkbox"/> &H80000012&
Height	246
HelpContextID	0
KeepScrollBarsVisible	3 - fmScrollBarsBoth
Left	12
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Picture	(None)
PictureAlignment	2 - fmPictureAlignmentCenter
PictureSizeMode	0 - fmPictureSizeModeClip
PictureTiling	False
ScrollBars	0 - fmScrollBarsNone
ScrollHeight	0
ScrollLeft	0
ScrollTop	0
ScrollWidth	0
SpecialEffect	2 - fmSpecialEffectSunken
TabIndex	2
TabStop	True
Tag	
Top	120
Visible	True
Width	222
Zoom	100

Slika 12: Svojstva TreeView kontrole koja možemo postaviti u prozoru Properties VBE Editora

#### 1.5.4 Događaji TreeView kontrole

Događaje (*Events*) TreeView kontrole koristimo za radnje poduzete od strane korisnika ili za obradu neke akcije uzrokovane izvršavanjem koda. Uz standardne, *Click* i *DoubleClick*, TreeView kontrola ima slijedeće događaje:

##### **-Collapse**

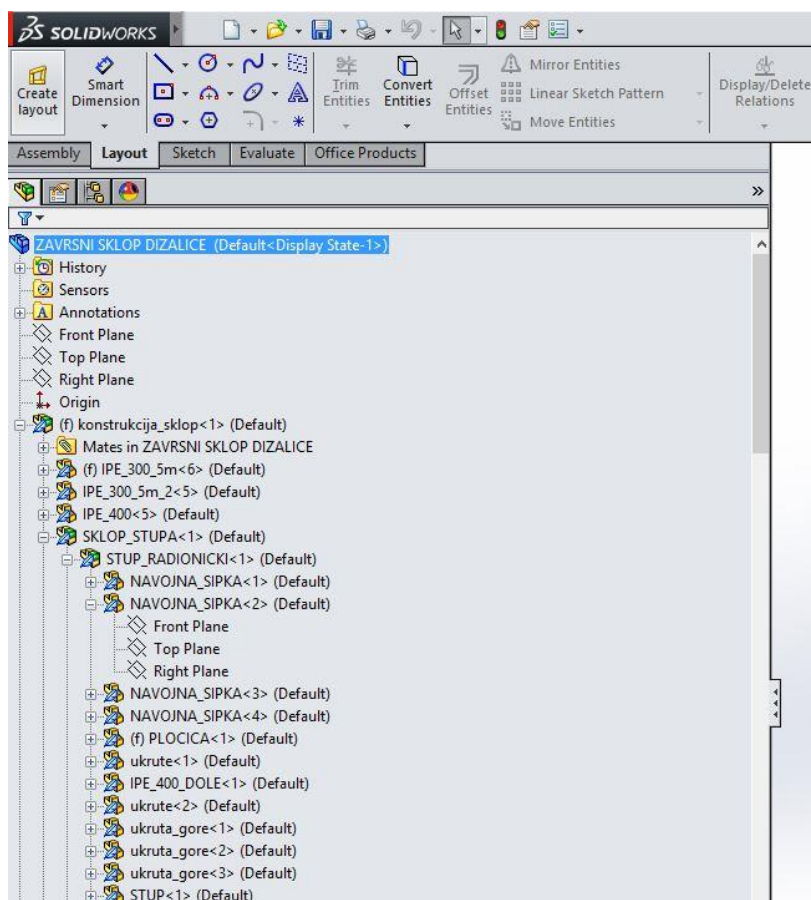
Collapse se generira svaki put kada je *Expanded* svojstvo čvora postavljeno na *False* i kad korisnik klikne na - oznaku s namjerom zatvaranja neke otvorene grane.

##### **-Expand**

Expand se generira kada se klikne na oznaku + s ciljem da raširimo granu, odnosno kada je *Expanded* svojstvo postavljeno na *True*. To svojstvo ima i jedan argument, objekt *Node* koji želimo "otvoriti", tj. detaljno prikazati sadržaj cijele podgrane.

##### **-NodeClick**

*NodeClick* je događaj od strane korisnika, a do njega dolazi klikom miša na neki čvor koji je pritom argument događaj



Slika 13: Primjer realizacije hijerarhijske strukture sklop/komponente pomoću TreeView kontrole u Solidworks-u

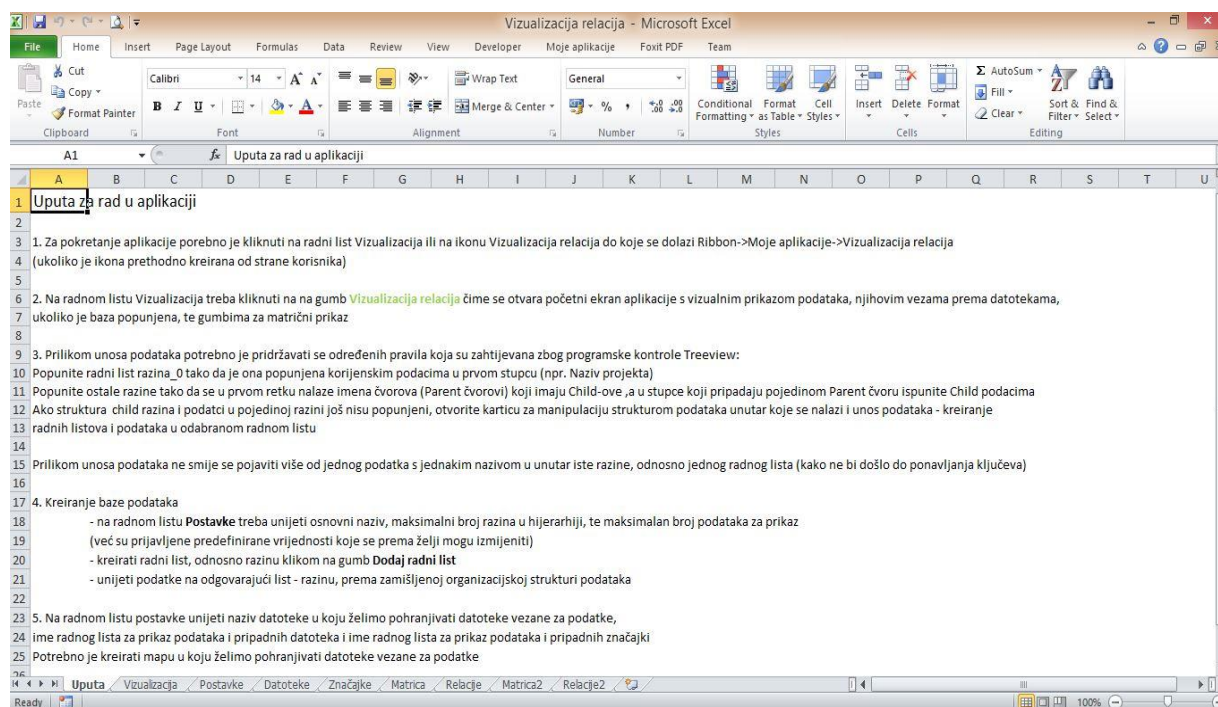
## 2. Struktura aplikacije za vizualizaciju relacija

### 2.1 Unaprijed uređeni radni listovi

Kada korisnik preuzme aplikaciju (xslm datoteku pod nazivom Vizualizacija relacija) i pokrene, prikazati će se nekoliko unaprijed uređenih radnih listova, potrebnih za rad aplikacije. Prvi radni list je "Uputa" na kojem su navedena pravila kojih se treba pridržavati kako bi aplikacija pravilno radila. Na radnom listu "Vizualizacija" nalazi se gumb "Vizualizacija relacija", čijim klikom se pokreće aplikacija. Korisnik može pokrenuti aplikaciju i preko ikone na vrpici (*Ribbon*), preko kartice "Moje aplikacije" čije umetanje je detaljnije opisano u odjeljku 5.

Na radnom listu "Postavke" potrebno je podesiti naziv radnih listova (koji će se pomoću opcija u formi kreirati) za unos podataka koji će se unositi u hijerarhijski objekt TreeView, maksimalni broj razina, maksimalni broj podataka za prikaz, ime mape za spremanje datoteka (koja je unaprijed kreirana kao podmapa mape u kojoj se nalazi xslm datoteka Vizualizacija relacija), naziv radnog lista za spremanje datoteka, naziv radnog lista za unos značajki, odabrati boju zaglavlja matrica te boju unutrašnjosti matrica. Ukoliko je korisnik izmijenio nazive lista za prikaz datoteka i lista za prikaz značajki, potrebno je izmijeniti imena radnih listova koji nose naziv "Datoteke" i "Značajke", koji su postavljeni kao početni nazivi, ili u tu svrhu dodati nove radne listove koji nose nazive navedene u radnom listu "Postavke".

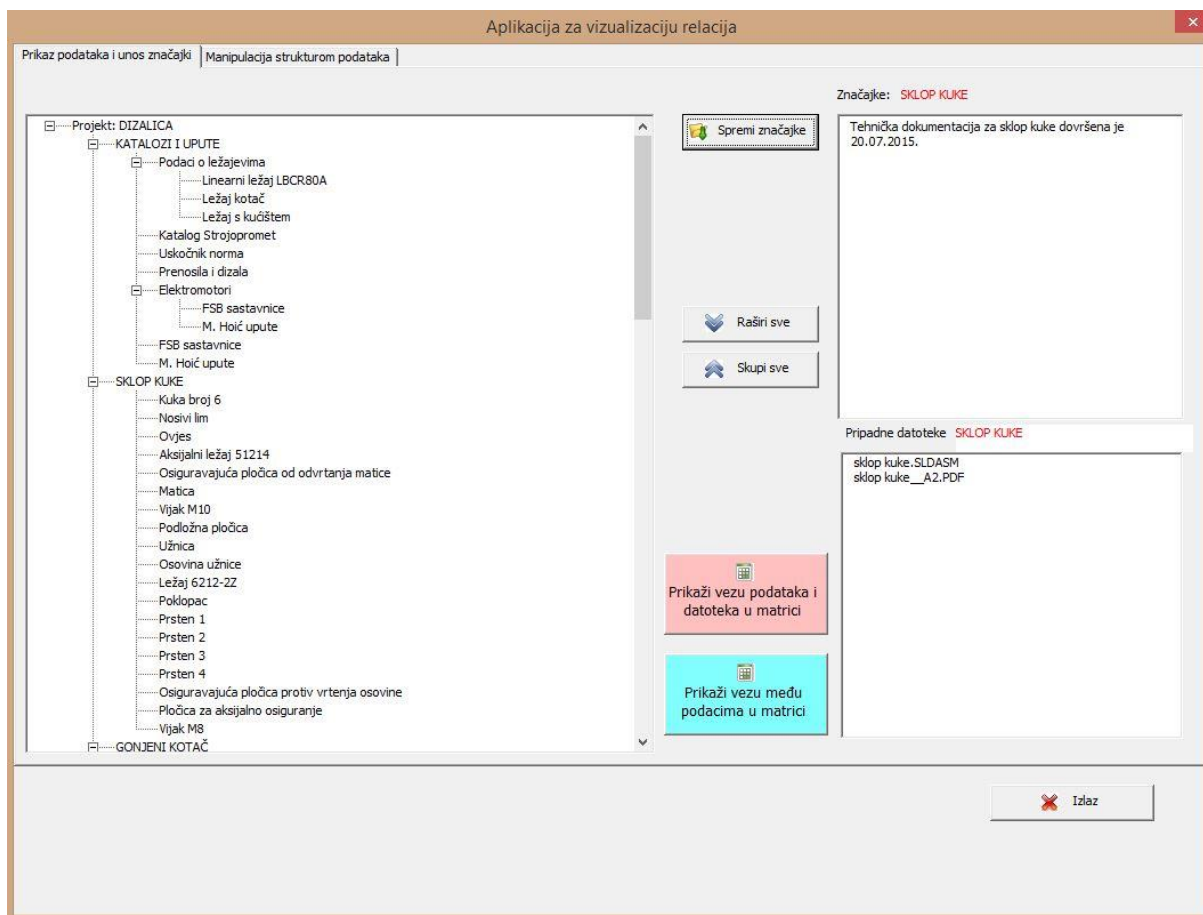
Pri preuzimanju aplikacije unaprijed su određeni i listovi "Matrica", "Relacije", "Matrica2" i "Relacije2". U listovima "Matrica" i "Matrica2" prikazati će se određene matrice klikom na gumbе u aplikaciji. U listovima "Relacije" i "Relacije2" korisniku je omogućeno definiranje veza koje može postavljati po pripadajućim matricama prema uputama navedenim u 5.



Slika 14: Unaprijed postavljani radni listovi aplikacije

## 2.2 Forma aplikacije za vizualizaciju relacija

### 2.2.1 Kartica “Prikaz podataka i unos značajki“



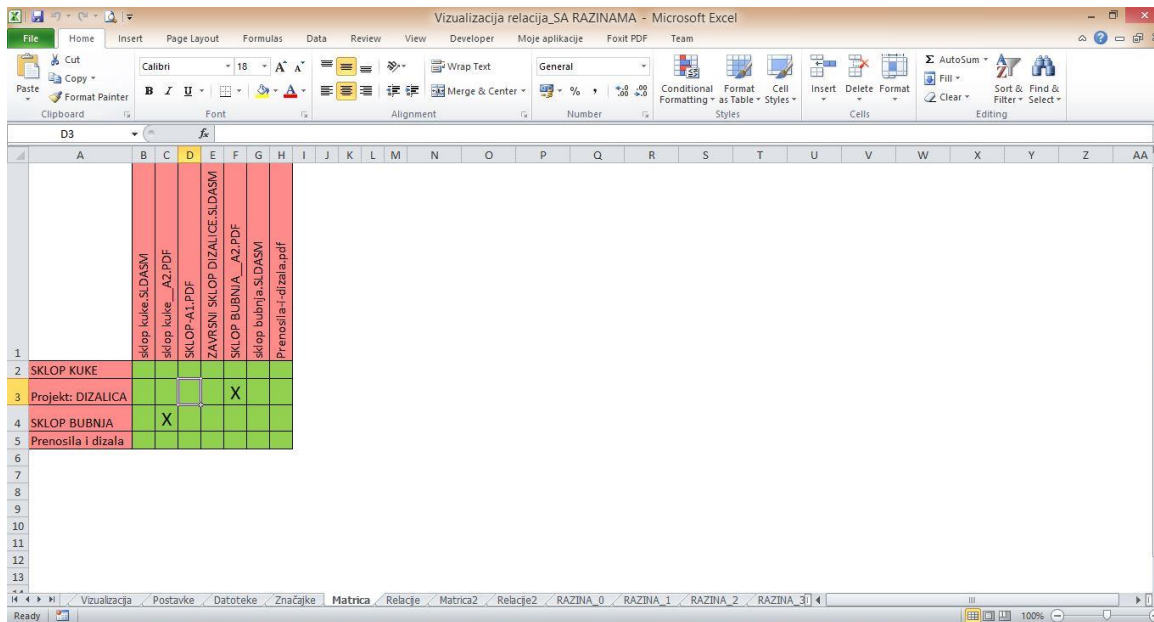
Slika 15: Kartica “Prikaz podataka i unos značajki“

Forma se sastoji od dvije kartice (*tab*). Prva kartica je “Prikaz podataka i unos značajki“. Sastoji se od *TreeView* kontrole, gumbi “Spremi značajke“, “Raširi sve“, “Skupi sve“, “Prikaži vezu podataka i datoteka u matrici“, “Prikaži vezu među podacima u matrici“ i “Izlaz“, dvije pune labela “Značajke:“ i “Pripadne datoteke:“, dvije prazne labela u kojima se ispisuje naziv selektiranog čvora, *TextBox* kontrole te *ListBox* kontrole. Na ovoj kartici je isključivo hijerarhijski prikaz strukture zadane u radnim listovima, a manipulacija strukturom nalazi se na drugoj kartici, kao i kreiranje radnih listova za unos podataka. Gumb “Raširi sve“ otvara i prikazuje sve razine, grane *TreeView* kontrole te omogućuje prikaz svih podataka. Gumb “Skupi sve“ zatvara stablo te ostaju vidljivi samo korijenski čvorovi (*Root*).

Klikom na pojedini podatak u *TreeView* kontroli, njegovo se ime ispisuje unutar spomenutih praznih labela. Istim događajem (selektiranjem čvora) u *ListBox* kontroli ispisuju se datoteke koje pripadaju selektiranom podatku.

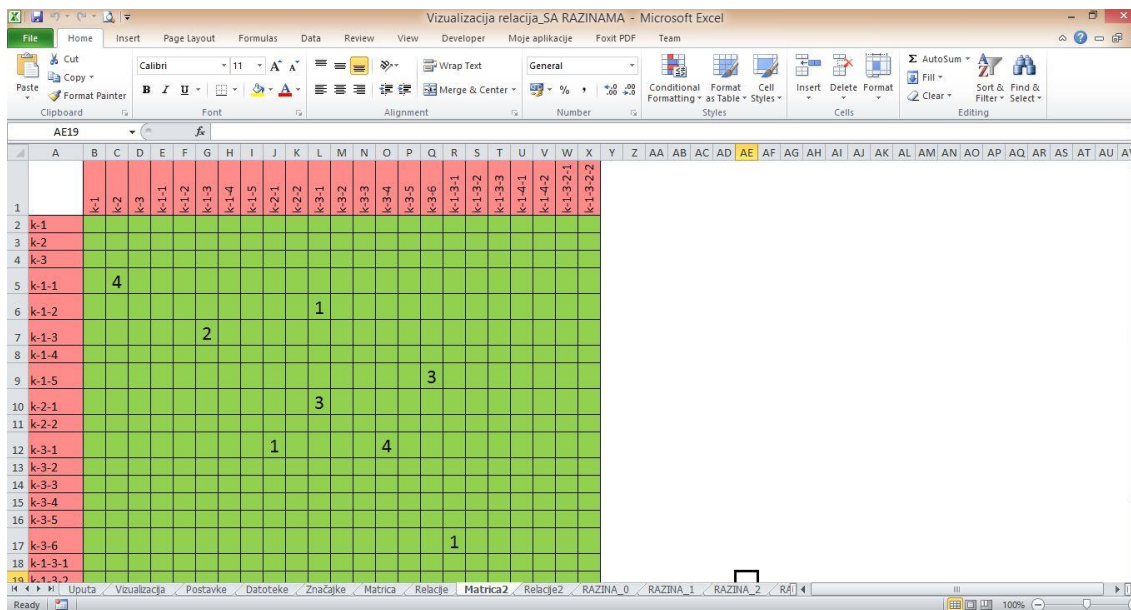
U *TextBox* kontroli omogućeno je da korisnik unese značajke podatka koji je odabrao u *TreeView* kontroli te ih spremi u radni list “Značajke“ klikom na gumb “Spremi značajke“. Prilikom spremanja u radni list, u prvi stupac će se zapisati ime podatka čije značajke spremamo, a u drugi stupac unesene značajke.

Klikom na gumb “Prikaži vezu podataka i datoteka u matrici“ forma se zatvara i otvara se radni list “Matrica“. Na listu je prikazana matrica čiji su retci podaci kojima smo dodavali ili uklanjali neke datoteke ( na kartici “Manipulacija strukturom podataka“ ) te sve promjene spremili, a stupci su datoteke zapisane u radnom listu “Datoteke“, one koje su preostale nakon svih promjena. Na datoteke je postavljen hyperlink (Slika 16).



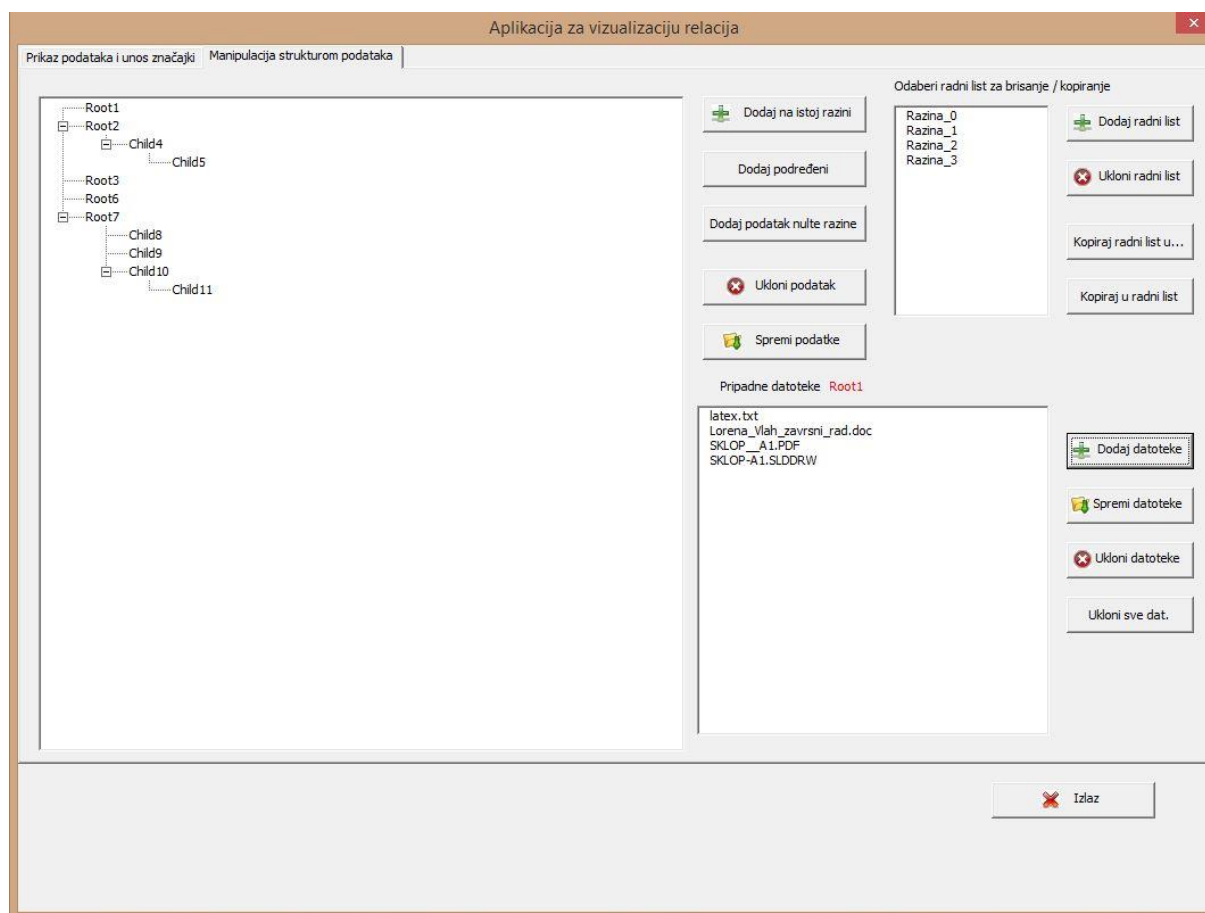
Slika 16: Radni list “Matrica“ nakon klika na gumb “Prikaži vezu podataka i datoteka u matrici“, s postavljenim vezama

Klikom na gumb “Prikaži vezu među podacima u matrici“ forma se zatvara i otvara se radni list “Matrica2“. Na listu je prikazana matrica kojoj su retci i stupci svi podaci iz TreeView kontrole. Dvoklikom na bilo koje polje u unutrašnjosti matrice pojavi se padajući izbornik u kojem odabiremo jednu od veza koje smo definirali na listu “Relacije2“.



Slika 17: Radni list “Matrica2“ nakon klika na gumb “Prikaži vezu među podacima u matrici“, s postavljenim vezama

## 2.2.2 Kartica “Manipulacija strukturom podataka”



Slika 18: Kartica “Manipulacija strukturom podataka”

Druga kartica je “Manipulacija strukturom podataka“. Sastoji se od TreeView kontrole, gumbi “Dodaj radni list“, “Ukloni radni list“, “Kopiraj radni list u...“, “Kopiraj u radni list“, “Dodaj na istoj razini“, “Dodaj podređeni“, “Dodaj podatak nulte razine“, “Ukloni podatak“, “Spremi podatke“, “Dodaj datoteke“, “Spremi datoteke“, “Ukloni datoteke“, “Ukloni sve dat.“, dvije pune labele “Pripadne datoteke“ i “Odaberi radni list za brisanje / kopiranje“, jedne prazne labele u kojoj se ispisuje naziv selektiranog čvora, te dvije *ListBox* kontrole, jedna za prikaz radnih listova koji sadrže podatke za hijerarhijski prikaz, a druga za prikaz datoteka.

Da bi mogao unijeti podatke, korisnik mora kreirati radne listove. Klikom na gumb “Dodaj radni list“ kreira se radni list koji nosi naziv unesen u radni list “Postavke“ a programski mu se doda znak “\_“ i broj. Manipulacija unesenim radnim listovima omogućena je gumbima “Ukloni radni list“, “Kopiraj radni list u...“ i “Kopiraj u radni list“. Ukoliko želimo ukloniti neki od radnih listova s *Listbox* popisa, trebamo selektirati list u *ListBox*-u i kliknuti na odgovarajući gumb. Ako list koji uklanjamo već ima neki sadržaj, aplikacija će poslati prozor *MsgBox* sa upozorenjem da će i postojeći sadržaj biti obrisan. Ukoliko korisnik, nakon što označi neki list u *ListBox* kontroli, klikne na gumb “Kopiraj radni list u...“, pojaviti će se prozor *InputBox* u koji je potrebno upisati ime radnog lista u koji želimo kopirati sadržaj radnog lista selektiranog u *ListBox*-u. Gumb “Kopiraj u radni list“ otvara *InputBox* u koji upisujemo bilo koji radni list koji imamo kreirani u radnoj knjizi, čiji sadržaj



želimo kopirati u neki od radnih listova u *ListBox*-u (koji su razine *TreeView* kontrole). Prvo je potrebno označiti u *ListBox*-u list u koji kopiramo sadržaj, zatim kliknuti na gumb “Kopiraj u radni list“, te u *InputBox* okvir unesti ime lista čiji sadržaj kopiramo. Aplikacija šalje povratne informacije o kopiranom sadržaju.

Ako korisnik već ima u nekim radnim listovima upisane podatke koje želi prikazati u *TreeView* kontroli, oni obavezno moraju slijediti zadanu strukturu, koja je prikazana na slici 19 ukoliko se radi o korijenskim čvorovima, odnosno na slici 20 za *Child* čvorove.

	A	B	C
1	k-1		
2	k-1-1		
3	k-1-2		
4			
5			
6			
7			
8			
9			

Slika 19: (Lijevo) Struktura radnog list korijenskih članova

	A	B	C
1	k-1	k-2	k-3
2	k-1-1	k-2-1	k-3-1
3	k-1-2	k-2-2	k-3-2
4	k-1-3		k-3-3
5	k-1-4		k-3-4
6	k-1-5		k-3-5
7			k-3-6
8			
9			
10			
11			

Slika 20: (Desno) Struktura radnog lista s *Child* čvorovima kojima su *Parent* čvorovi korijenski čvorovi

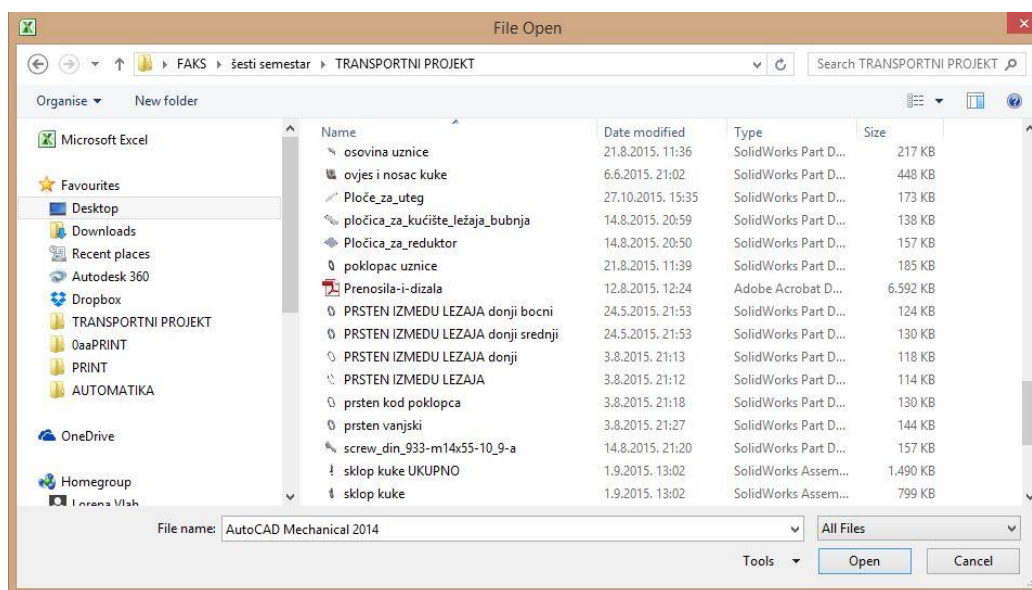
U jednom radnom listu ne smije doći do članova koji imaju isto ime, vrijednost, zbog toga što ključ, potreban za učitavanje sadržaja u *TreeView* kontrolu, mora biti jedinstven. U ovom radu ključevi su kreirani kao skup znakova, ime radnog lista na kojem se član nalazi i vrijednost njegove ćelije. „Nulti“ radni list ima sve članove, korijenske čvorove, ispisane u prvom stupcu (Slika 19). „Razina 1“ mora u prvom retku imati imena samo onih korijenskih članova koji imaju *Child* čvorove, a u retcima ispod *Root* čvora trebaju biti nazivi pripadnih *Child* čvorova. Kao primjer takve strukture priložena je slika 19. No, da bi se podaci iz nekih radnih listova kreiranih u trenutno aktivnoj radnoj knjizi mogli učitati u *TreeView* kontrolu, moraju biti u radnim listovima koji nose naziv kako programski kod zahtijeva. Zbog toga je potrebno sadržaj radnog lista sa unaprijed popunjenim podacima kopirati u radne listove kreirane unutar same forme aplikacije. To činimo preko gumba “Kopiraj u radni list“. U tom slučaju će se željeni sadržaj automatski učitati u *TreeView* kontrolu prilikom pokretanja aplikacije na gumb “Vizualizacija relacija“.

Drugi način unosa podataka je direktno iz forme aplikacije. Podaci uneseni/izbrisani preko forme, nakon klika na gumb “Spremi podatke“ biti će, prema zadanim pravilima strukture radnih listova, učitani u odgovarajuće radne listove. Moguće je iz forme popunjavati prazne radne listove i one koji već imaju postojeći sadržaj. Dodavanje korijenskih čvorova vrši se klikom na gumb “Dodaj podatak nulte razine“. Pritom se otvara *InputBox* u koji je potrebno unijeti naziv novog čvora. Gumb “Dodaj podatak“ omogućuje dodavanje podatka

koji je na istoj razini kao i odabrani, selektirani čvor, a naziv mu se dodjeljuje također preko *InputBox* prozora. Na gumb “Dodaj podređeni“ dodajemo Child čvor selektiranom čvoru.

Klikom na gumb “Spremi podatke“ novi podaci, kao i ostale promjene, zapisuju se u odgovarajuće radne listove.

Moguće je odabranom čvoru dodavati i brisati datoteke. Klikom na neki čvor, njegovo ime ispiše se u prethodno praznoj labeli, a u *ListBox*-u, koji se nalazi ispod te labele, ispišu se datoteke koje već pripadaju selektiranom čvoru. Na gumb “Dodaj datoteke“ otvara se dijaloški okvir *Open Dialog*.



Slika 21: *Open Dialog*

U *Open Dialog* okviru moguće je odabrati više datoteka. Klikom na gumb “Spremi datoteke“ dodane datoteke spremaju se u mapu koju smo prethodno kreirali na računalu kao podmapu mape u kojoj se nalazi i aplikacija. Ime mape u kojoj želimo pohranjivati dodane datoteke postavljamo na radnom listu “Postavke“. Nakon klika na gumb “Spremi datoteke“ pojaviti će se prozor kao na slici 22.



Slika 22: *MsgBox* nakon klika na gumb “Spremi datoteke“

U *MsgBox*-u će se ispisati podatak na koji smo dodali datoteke, lokacija gdje su one pohranjene, te broj spremljenih datoteka. Promjena će biti vidljiva i na radnom listu “Datoteke“. Ako smo odabrali podatak koji već ima pridružene neke datoteke, nove datoteke

zapisati će se u radni list "Datoteke", u redak koji pripada odabranom čvoru, nakon već postojećih datoteka. U radnom listu "Datoteke" u prvi stupac zapisuju se ključevi čvorova koji imaju datoteke, u drugi stupac nazivi tih čvorova (koji su nam potrebni za kreiranje članova matrice), a preostali stupci sadrže nazive datoteka. Ukoliko želimo ukloniti datoteke, trebamo u *ListBox*-u selektirati podatke koje želimo ukloniti i kliknuti na gumbe "Ukloni datoteke" ili "Ukloni sve dat.". Klikom na gumb "Spremi" pojaviti će se prozor s obavijesti da su datoteke uklonjene, a primjena je ažurirana i na radnom listu "Datoteke".

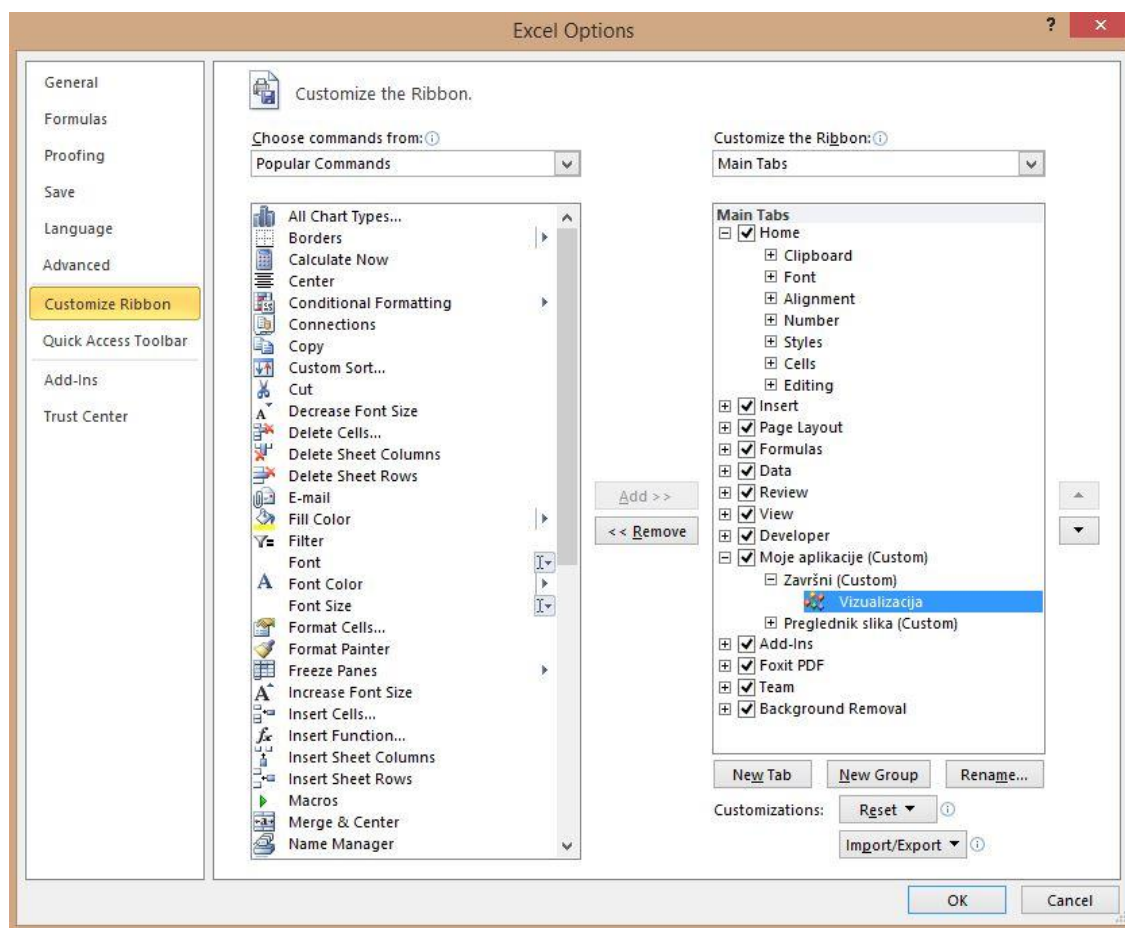
Gumb "Izlaz" vidljiv je u formi u svakom trenutku. Za razliku od toga gumbi za manipulaciju podacima i datotekama postanu vidljivi tek nakon što je selektiran neki član hijerarhijskog stabla. Klikom miša na gumb "Izlaz" aplikacija se zatvara i briše iz memorije.

### 3. Upute za korištenje aplikacije za vizualizaciju relacija

#### 3.1 Pokretanje aplikacije

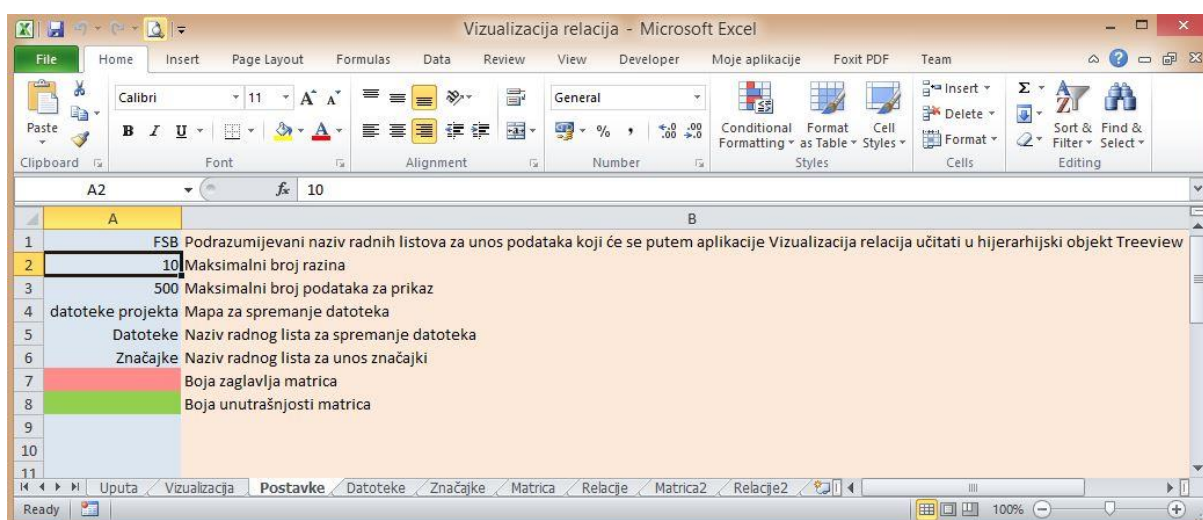
Kako bi mogao pokrenuti aplikaciju, korisnik treba imati u svom MS Excelu omogućeno otvaranje *macro*-a . Potrebno je dodati Developer karticu na vrpcu (*Ribbon*). To činimo klikom na *File*→*Options*→*Customize Ribbon*. Na desnoj strani pojaviti će se labela *Customize the Ribbon* te *ComboBox* izbornik u kojem odabremo *Main Tabs*. U *CheckBox* kontroli označimo da želimo prikaz *Developer* kartice na vrpici. U *Developer* kartici u skupini *Code* odaberemo *Macro Security* i selektiramo opciju *Enable all macros*.

Nakon što korisnik preuzme i pokrene aplikaciju, u radnom listu “Postavke“ postavi tražene podatke, pokreće formu aplikacije na gumb “Vizualizacija relacija“. Aplikaciju je moguće pokrenuti i preko ikone u vrpici, ukoliko je prethodno kreirana ikona u izborniku *Main Tabs* kao na slici 23.



Slika 23: Izbornik u kojem je moguće kreirati ikonu za pokretanje aplikacije preko vrpce

Prije nego što uđemo u formu, potrebno je na listu "Postavke" ispuniti ćelije (prvi stupac) koje traže naziv radnih listova za unos podataka koji će se preko aplikacije Vizualizacija relacija učitati u hijerarhijski objekt TreeView, maksimalni broj razina, maksimalni broj podataka za prikaz, ime mape za spremanje datoteka (ta mapa mora biti kreirana kao podmapa mape u kojoj se nalazi xlsx datoteka Vizualizacija relacija), naziv radnog lista za spremanje datoteka, naziv radnog lista za unos značajki. Također je potrebno odabrati boje zaglavlja i unutrašnjosti matrica. Ukoliko je korisnik u listu "Postavke" promijenio nazive već postavljenih listova "Datoteke" i "Značajke", potrebno je ručno promijeniti nazive tih listova u one koji su sada postavljeni u listu "Postavke". Listovi koji su unaprijed kreirani i prikazuju se pokretanjem xlsx datoteke Vizualizacija relacija su "Upute", "Vizualizacija", "Postavke", "Datoteke", "Značajke", "Matrica", "Relacije", "Matrica2" i "Relacije2".

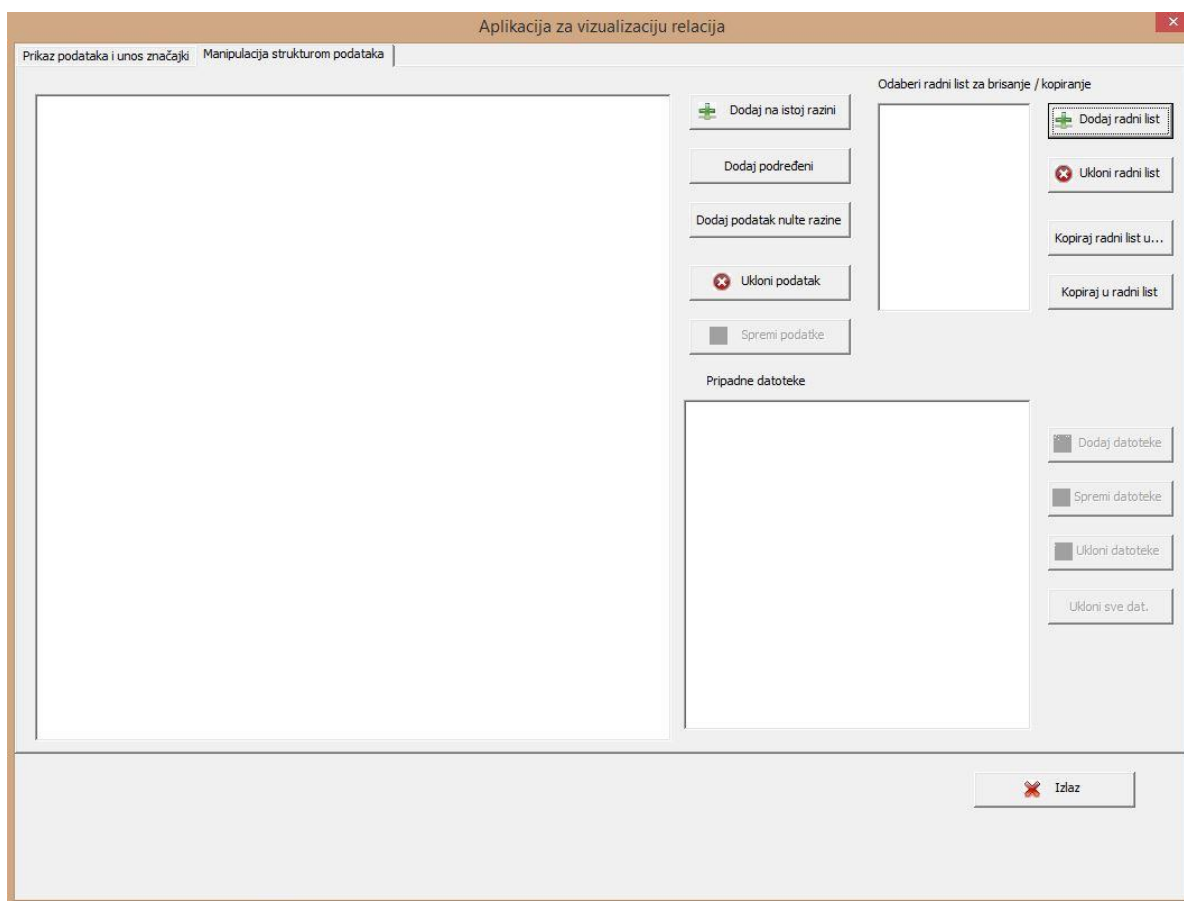


Slika 24: Radni list "Postavke"

Ukoliko korisnik ne želi sve podatke učitati preko forme, direktno preko TreeView kontrole, već ima podatke na naknadno kreiranim radnim listovima i želi prvo vidjeti njihov hijerarhijski prikaz u stablu, mora slijediti određena pravila strukture tih listova i sljedeće upute. Razina koja je popunjena korijenskim podacima mora imati te podatke unesene u prvi stupac, jedan iz drugoga kao što je prikazano na slici 19. Oni članovi te razine koji imaju *Child* podatke, trebaju na idućoj razini (radnom listu) biti ispisani u prvi redak. Svaki član tog retka mora u stupcu ćelije kojoj pripada imati jedan za drugim ispisane *Child* podatke (Slika 20). Takva struktura radnog lista vrijedi za sve razine koje je korisnik sam unio u Excel, a samo ime tih radnih listova je nebitno zbog toga što se tek unutar forme pravilno kreiraju radni listovi kojima punimo TreeView kontrolu. Bitno je napomenuti da unutar jednog radnog lista s podacima ne smije doći do ponavljanja naziva podatka, zbog toga što su ključevi članova kojima punimo TreeView kreirani tako da sadržavaju ime radnog lista podatka te njegov sadržaj. Ključevi moraju biti jedinstvene vrijednosti.

### 3.2. Popunjavanje i prikaz TreeView kontrole

Formu aplikacije za vizualizaciju relacija pozivamo klikom na gumb “Vizualizacija relacija” koji se nalazi na radnom listu “Vizualizacija” ili pomoću ikone na vrpci (Ribbon) ukoliko smo ju kreirali. TreeView prikaz je prazan pa se odmah otvara kartica “Manipulacija strukturom podataka” kako bismo “napunili” kontrolu za hijerarhijski prikaz podataka.



Slika 25: Kopiranje nekog radnog lista radne knjige u list iz kojeg se čitaju podaci za TreeView

Gumbi “Spremi podatke” i oni vezani uz manipulaciju datotekama su zaključani sve dok se ne selektira neki podatak iz stabla. Kao primjer u uputama za korištenje aplikacije, u radnoj knjizi kreirano je nekoliko radnih listova čiji je sadržaj prikazan na sljedećim slikama.

C10		
	A	B
1	FSB	
2		
3		
4		
5		
6		

D19		
	A	B
1	FSB	
2	Preddiplomski	
3	Diplomski	
4		
5		
6		

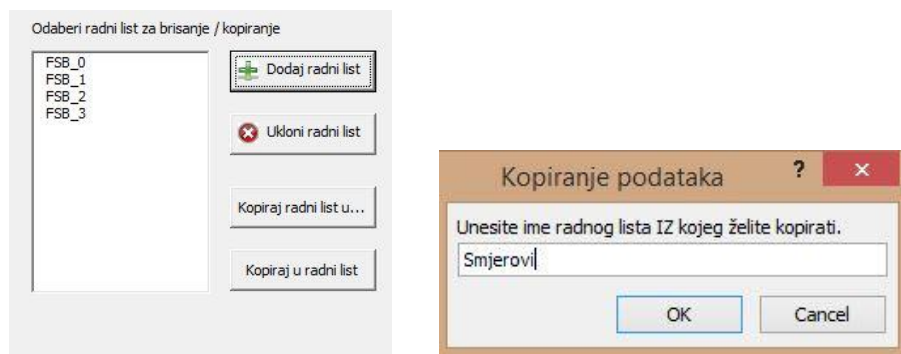
L24			
	A	B	C
1	Preddiplomski	Diplomski	
2	Strojarstvo1	Strojarstvo2	
3	Brodogradnja1	Brodogradnja2	
4	Zrakoplovstvo1	Zrakoplovstvo2	
5			
6			
7			

Slika 26: Radni listovi kreirani od strane korisnika nazvani (redom slijeva) “Fakultet”, “Studiji”, “Glavni\_smjerovi”

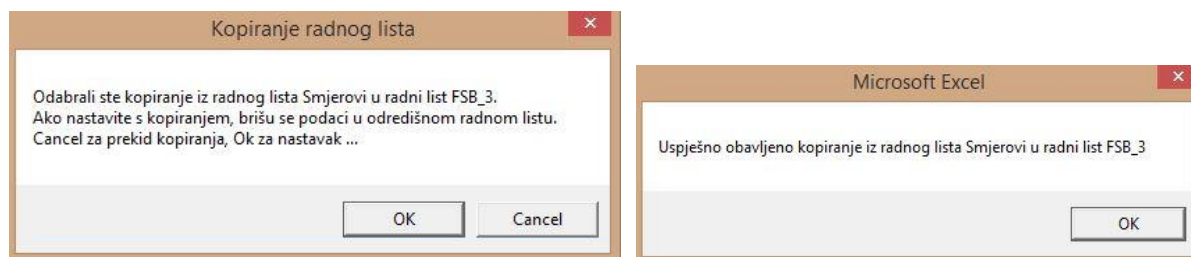
	A	B
1	Strojarstvo1	Strojarstvo2
2	Brodostrojarski1	Brodostrojarski2
3	Industrijsko inženjerstvo i menadžment1	Industrijsko inženjerstvo i menadžment2
4	Inž. modeliranje i rač. simulacije1	Inž. modeliranje i rač. simulacije2
5	Inženjerstvo materijala1	Inženjerstvo materijala2
6	Konstruktivski1	Konstruktivski2
7	Mehatronika i robotika1	Mehatronika i robotika2
8	Procesno-energetski1	Procesno-energetski2
9	Proizvodno inženjerstvo1	Proizvodno inženjerstvo2
10	Računalno inženjerstvo1	Računalno inženjerstvo2
11		

Slika 27: Radni list kreiran od strane korisnika nazvan "Smjerovi"

Potrebno je sadržaje tih radnih listova kopirati u radne listove kojima su programski dodijeljeni nazivi. Na radnom listu "Postavke" unesen je naziv radnih listova za unos podataka „FSB“. Na gumb "Dodaj radni list" potrebno je dodati četiri nova lista, koji će se ispisati u *ListBox* kontroli. Npr. trebamo kopirati sadržaj radnog lista "Smjerovi" u radni list FSB\_3 koji smo kreirali klikom na gumb "Dodaj radni list". Prvo u *ListBox* kontroli odabremo list u koji želimo kopirati sadržaj, dakle radni list FSB\_3. Zatim kliknemo na gumb "Kopiraj u radni list". Otvori se *InputBox* u koji upišemo ime lista iz kojeg kopiramo, u ovom slučaju to je radni list "Smjerovi". Kada kliknemo na OK javiti će se dvije povratne poruke. Isti postupak trebalo bi ponoviti i za ostale razine, odnosno listove "Fakultet", "Studiji", i "Glavni smjerovi". Kada su listovi svih razina ispunjeni sadržajem po definiranoj strukturi, potrebno je izaći iz aplikacije i ponovno ju pokrenuti na gumb "Vizualizacija relacija". Sada je *TreeView* kontrola učitana i prikazana.

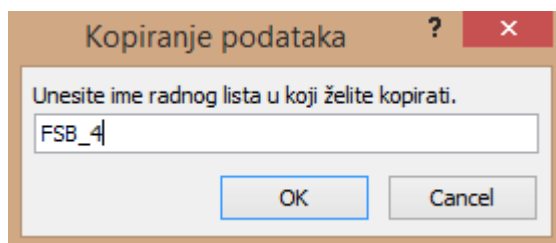


Slika 28: Gumb "Kopiraj u radni list" i povratni MsgBox



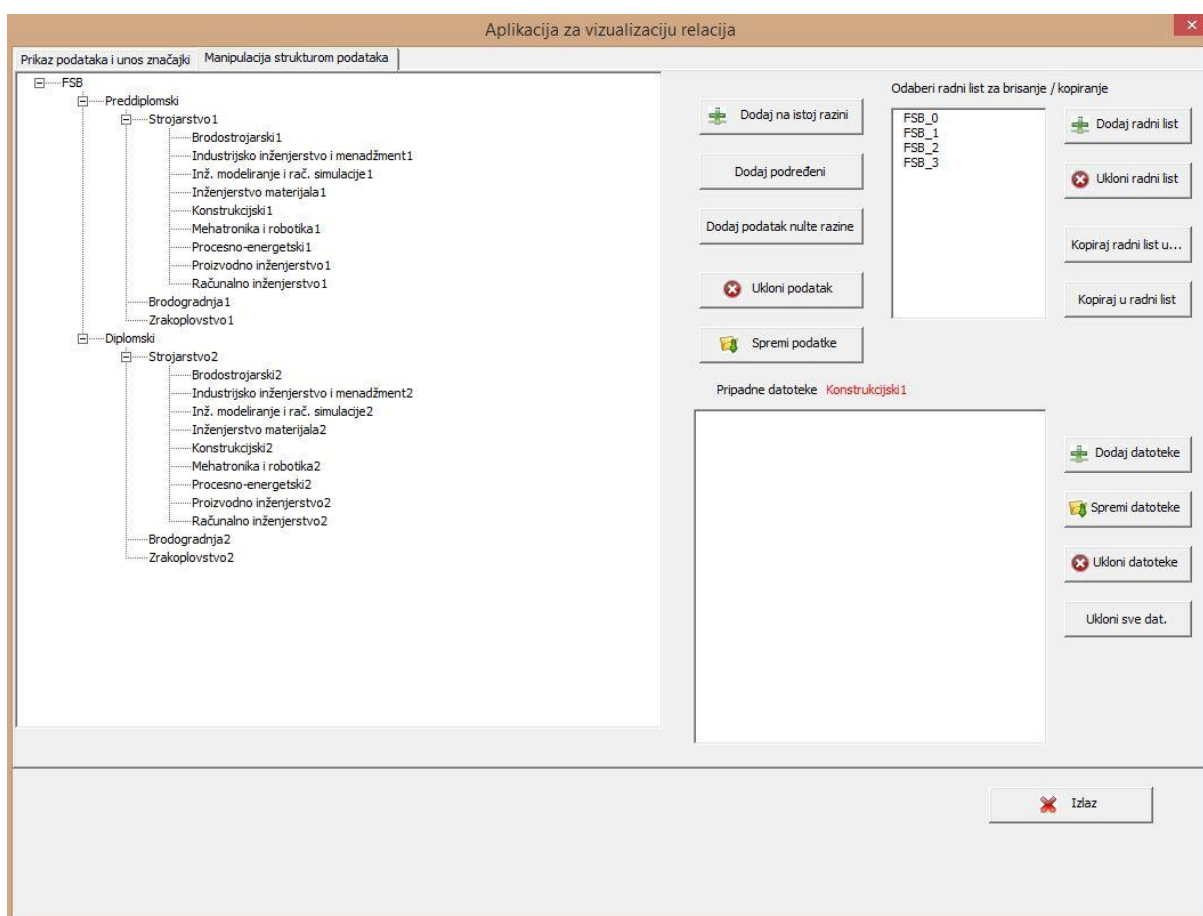
Slika 29: Nakon što se klikne OK u MsgBox-u "Kopiranje podataka", pojavljuju se još dvije povratne poruke

Radne listove možemo ukloniti na gumb “Ukloni radni list“, te kopirati sadržaj iz jednog u drugi na gumb “Kopiraj radni list u...“. Za brisanje lista potrebno je kliknuti na list koji želimo obrisati u ListBox kontroli, te na gumb za brisanje. Za kopiranje sadržaja iz nekog radnog lista u ListBox-u u drugi, koji može biti neki iz ListBox-a ili neki drugi, potrebno je selektirati list iz kojeg kopiramo, zatim gumb “Kopiraj u radni list“. Otvoriti će se InputBox u koji je potrebno upisati ime radnog lista u koji kopiramo. Ako upišemo ime nekog lista koji još ne postoji, on će se samim time što smo ga upisali u InputBox, kreirati.



Slika 30:MsgBox gumba “Kopiraj u radni list“

Ukoliko korisnik zna princip po kojem su dodijeljeni nazivi kreiranim radnim listovima iz programskog koda, može odmah svoje listove s podacima nazvati pravilno, te pokrenuti formu i na taj način učitati podatke (imerazine\_0 je prvi, itd.)

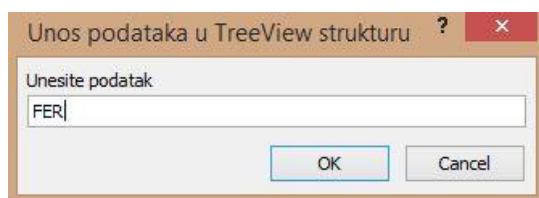


Slika 31: Podaci učitani u TreeView kontrolu s omogućenim gumbima za manipulaciju (klikom na čvor Konstrukcijski 1)



### 3.3 Manipulacija objektima koji su hijerarhijski strukturirani

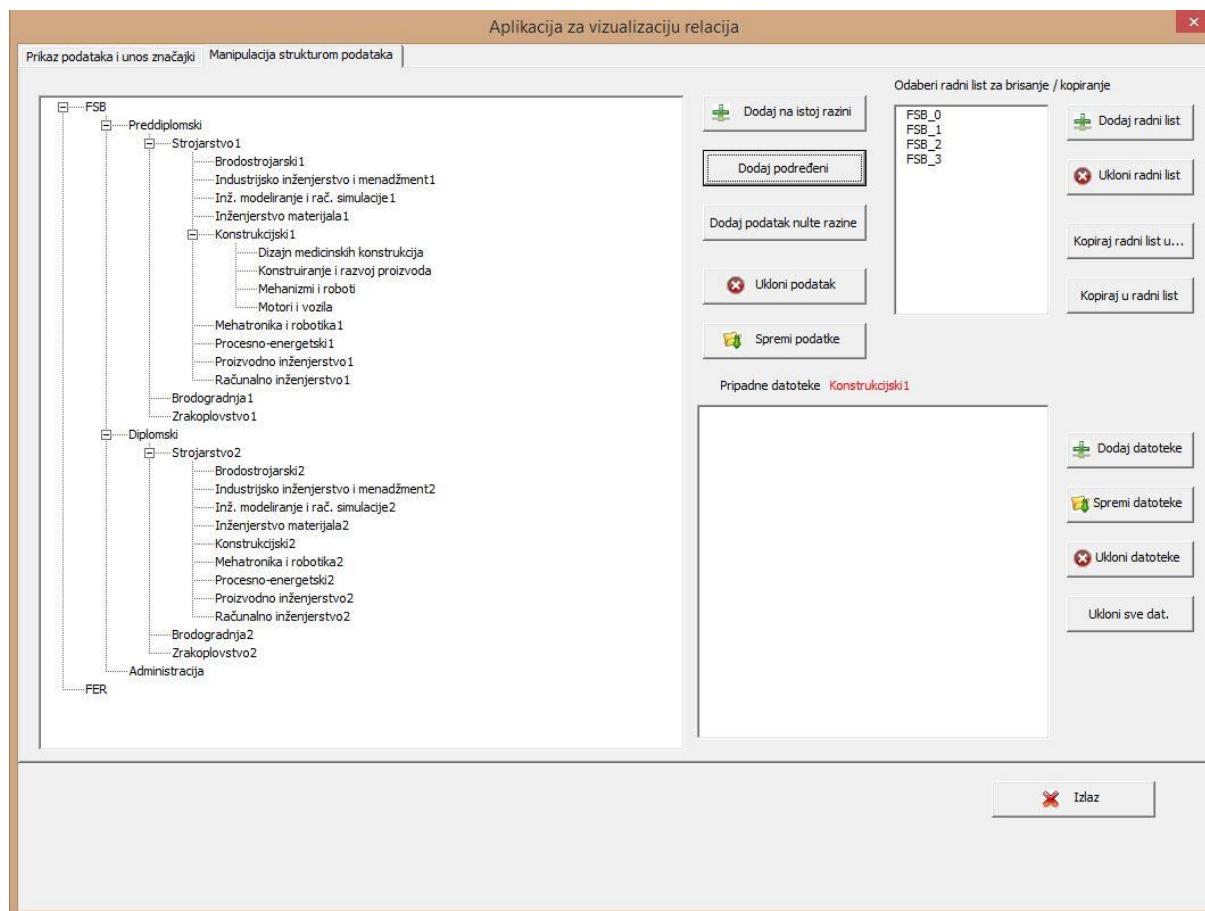
Treeview smo mogli popuniti i direktno iz forme. Tome služe gumbi “Dodaj podatak nulte razine”, “Dodaj na istoj razini” i “Dodaj podređeni”. Označimo bilo koji čvor na razini korijenskih podataka i kliknemo na gumb “Dodaj podatak nulte razine”. Pojaviti će se prozor *InputBox* u koji upisujemo ime novog čvora, te klikom na OK dodajemo čvor.



Slika 32: Unos još jednog korijenskog čvora (uz FSB)

Na isti se način dodaje čvor unutar neke postojeće razine na gumb “Dodaj na istoj razini”. U ovom primjeru selektiran je čvor “Preddiplomski” a klikom na gumb u istoj razini dodan je čvor “Administracija”.

Na gumb “Dodaj podređeni” dodajemo *Child* čvor selektiranom čvoru. U ovom primjeru na čvor “Konstrukcijski1” dodani su podsmjerovi. Na kraju svih unosa Treeview prikaz izgleda kao na slici 33.



Slika 33: Prikaz podataka u TreeView kontroli nakon izmjena učinjenih preko gumbi forme

Klikom na gumb “Spremi podatke“ radni listovi iz kojih su podaci učitani u *TreeView* prazne se i učitaju se podaci iz trenutnog stabla, zadržavajući zadanu strukturu radnih listova. U ovom primjeru pojaviti će se pet radnih listova sa sadržajem kao na slikama.

The image shows three screenshots of spreadsheet lists. The first, labeled I19, has columns A and B with rows 1 (FSB) and 2 (FER) highlighted. The second, labeled E11, has columns A and B with rows 1 (FSB), 2 (Preddiplomski), 3 (Diplomski), and 4 (Administracija) highlighted. The third, labeled L24, has columns A and B with rows 1 (Preddiplomski, Diplomski), 2 (Strojarstvo1, Strojarstvo2), 3 (Brodogradnja1, Brodogradnja2), and 4 (Zrakoplovstvo1, Zrakoplovstvo2) highlighted.

Slika 34: Listovi kreirani klikom na gumb “Spremi podatke“

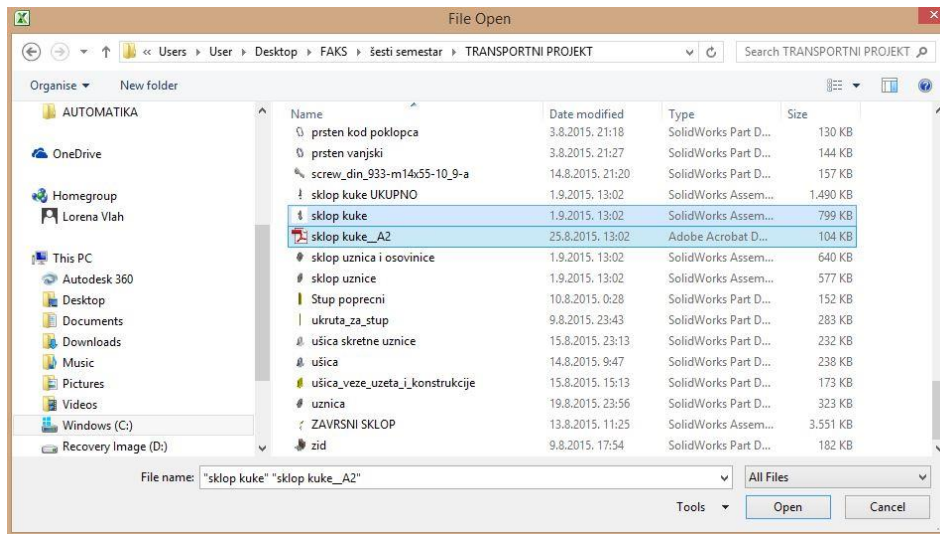
The image shows two screenshots of spreadsheet lists. The first, labeled D18, has columns A and B with rows 1 to 10 highlighted, containing various engineering and construction terms. The second screenshot shows a list with column A and rows 1 to 9 highlighted, containing terms like Konstrukcijski1, Dizajn medicinskih konstrukcija, and Motori i vozila.

Slika 35: Sadržaj listova kreiranih klikom na gumb “Spremi podatke“

### 3.4 Manipulacija datotekama

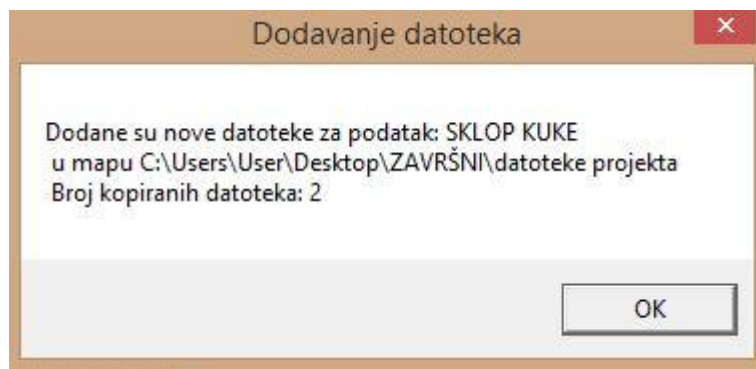
Kako bih napisala upute za manipulaciju datotekama, poslužila sam se projektom Portalni prijenosnik zbog toga što je to jedan od projekata koji je zahtijevao veliki broj datoteka, a time i njihov prikaz te pripadnost pojedinoj komponenti.

Nakon što se podaci prikazu u *TreeView* kontroli, potrebno je kliknuti na podatak za koji želimo dodati datoteke. Pritom se otključavaju gumbi za manipulaciju datotekama. U ovom primjeru to je čvor SKLOP KUKE, čiji naziv vidimo u labeli iznad *ListBox* kontrole za prikaz manipulacije datotekama. Klikom na gumb “Dodaj datoteke“ otvara se dijaloški okvir *Open Dialog*.

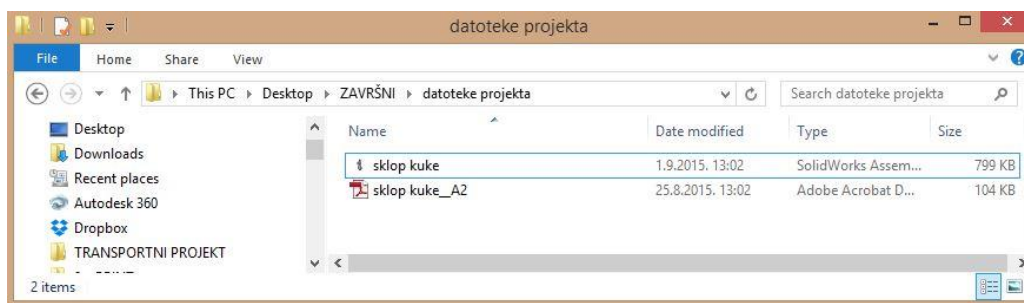


Slika 36: Dijaloški okvir Open Dialog

Moguće je selektirati više datoteka u jednom koraku. Datoteke su dodane u pripadni *ListBox*. Klikom miša na gumb “Spremi datoteke” one su spremljene u mapu “Datoteke projekta”, koju smo kreirali kao jednu od mapa podmape u kojoj se nalazi i xlsx datoteka Vizualizacija relacija, te u radni list “Datoteke”. U radnom listu “Datoteke” klikom na gumb “Spremi datoteke” u prvi stupac zapisani su ključevi onih podataka kojima su dodavane ili oduzimane datoteke, odnosno oni članovi koje je korisnik selektirao kako bi dodao ili oduzeo datoteke. U drugom stupcu su nazivi podataka. Ukoliko korisnik više puta radi promjene na istom podatku, nakon spremanja neće se ponovno ispisati ključ i naziv čvora na kojem su rađene promjene, već će se pretražiti prvi stupac i promjene će se vezati na već postojeće ime tog čvora. Nakon dodavanja pojaviti će se prozor *MsgBox* sa ispisanom porukom na koji podatak su dodane datoteke, putanja gdje su one sada pospremljene i njihov broj (Slika 37.)



Slika 37: MsgBox koji se pojavi nakon spremanja datoteka

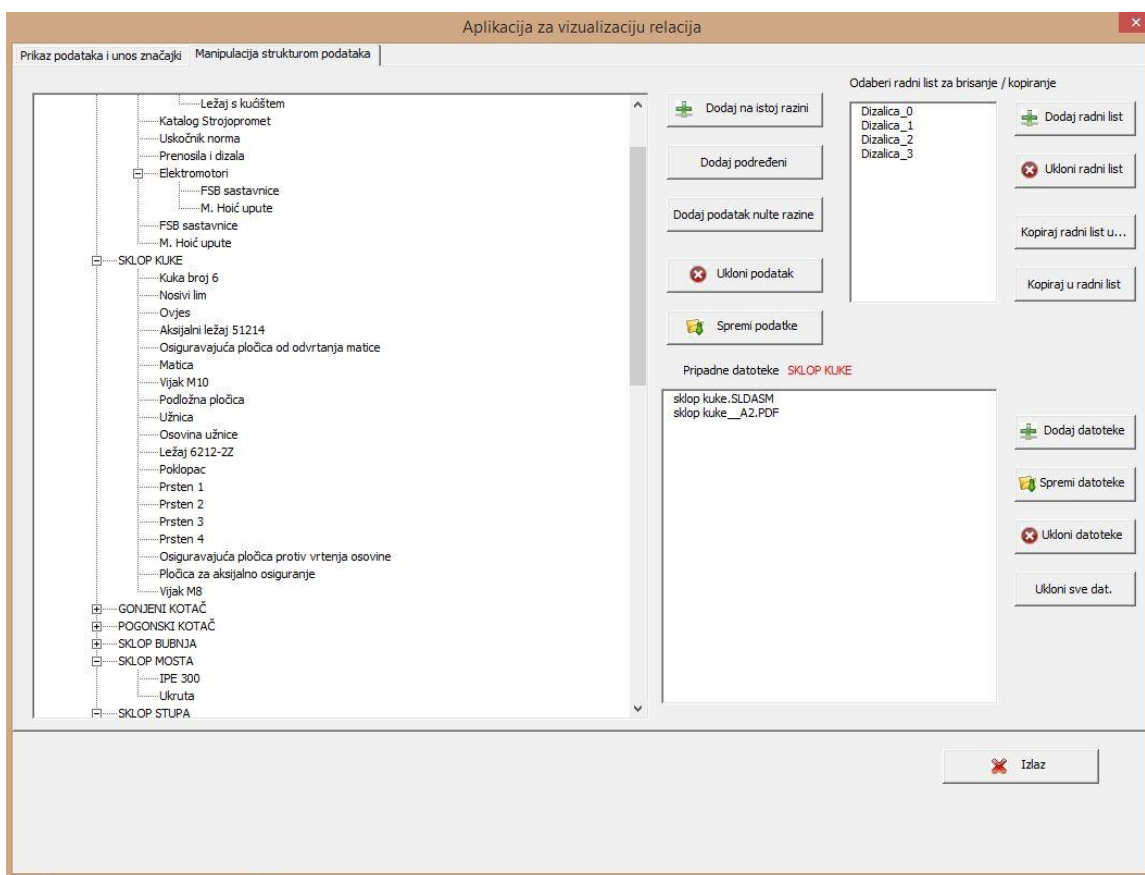


Slika 38. Mapa Datoteke projekta u koju su preko aplikacije pohranjene datoteke

Pomoću gumba “Ukloni datoteke“ uklanjamo datoteke selektirane u *Listbox*-u (mogući *multiselect*), a pomoću “Ukloni sve dat.“ odmah uklanjamo sve datoteke iz *ListBox*-a. Promjene je moguće spremiti na gumb “Spremi datoteke“.

	A	B	C	D
1	Dizalica_0_Dizalica	Dizalica	ZAVRSNI SKLOP DIZALICE.SLDASM	SKLOP-A1.PDF
2	Dizalica_1_SKLOP KUKE	SKLOP KUKE	sklop kuke.SLDASM	sklop kuke_A2.PDF
3	Dizalica_1_SKLOP BUBNJA	SKLOP BUBNJA	sklop bubnja.SLDASM	SKLOP BUBNJA_A2.PDF
4	Dizalica_2_Prenosila i dizala	Prenosila i dizala	Prenosila-i-dizala.pdf	
5	Dizalica_2_Čelna ploča 1	Čelna ploča 1	čeona ploča.SLDPRT	
6	Dizalica_2_Vijenac bubnja	Vijenac bubnja	vijenac bubnja.SLDPRT	
7				

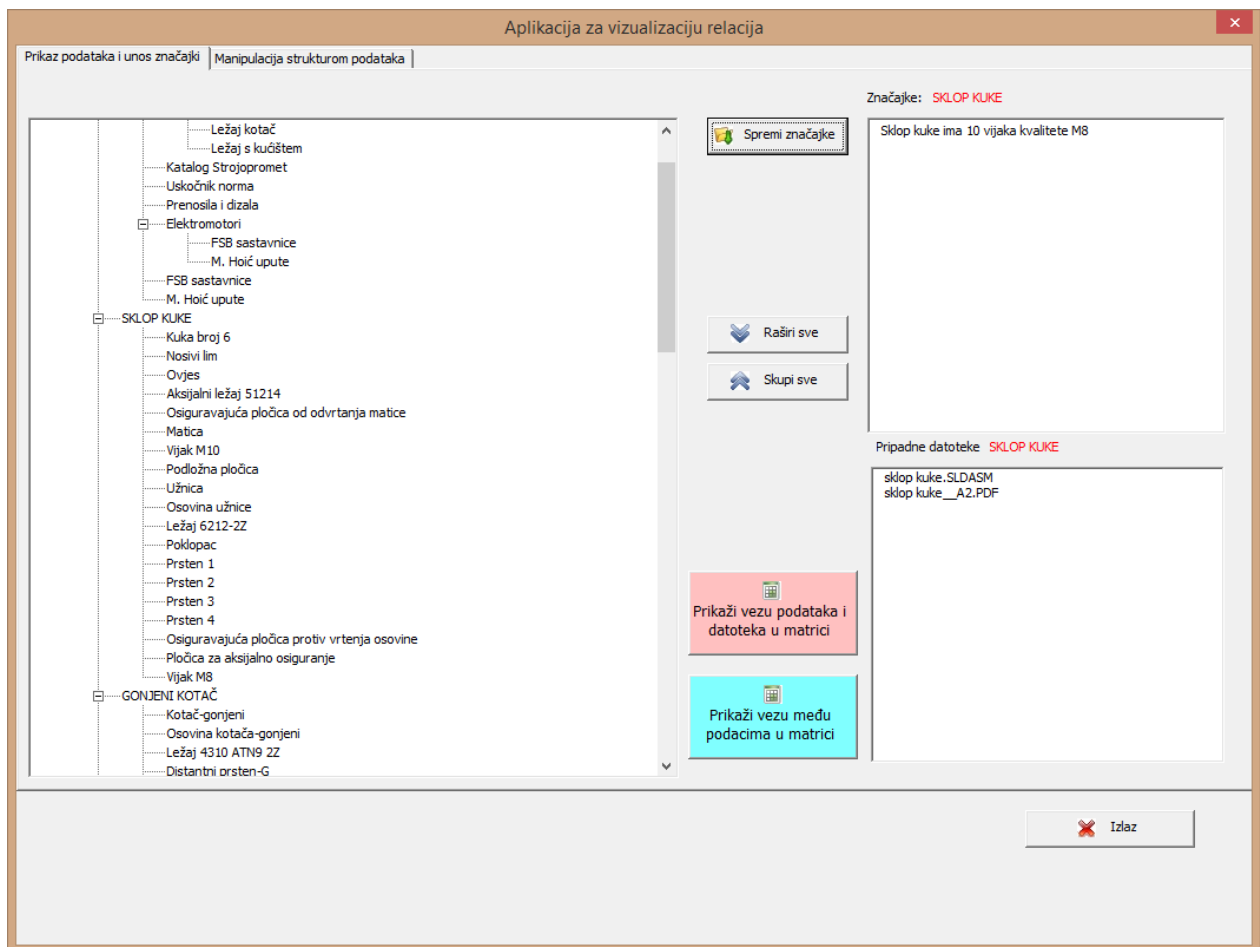
Slika 39: Radni list datoteke nakon unesenih promjena, prvi stupac pohranjuje ključeve, drugi nazive, a ostali pridružene datoteke



Slika 40: Forma aplikacije nakon promjena

### 3.5 Prikaz podataka i unos značajki

Kartica “Prikaz podataka i unos značajki“ sastoji se od prikaza TreeView kontrole, gumba za njezino potpuno skupljanje, gumba za potpuno grananje (otvaranje), ListBox kontrole sa datotekama selektiranog čvora, TextBox kontrole za unos značajki selektiranog podatka te dva gumba za matrični prikaz. Klikom miša na neki podatak, uključuje se gumb “Spremi značajke“. Kada u TextBox upišemo značajke vezane za podatak, i spremimo ih, one će se zapisati u radni list “Značajke“. U prvi stupac tog lista zapisuje se ključ, u drugi ime podatka, a u treći značajka. Klikom na gumb “Prikaži vezu podataka i datoteka u matrici“, forma se zatvara, a otvara se radni list “Matrica“ sa kreiranom matricom sa podacima i datotekama koje su zapisane u radnom listu “Datoteke“. Na radnom listu “Relacije“ korisnik upisuje veze i oznake tih veza koje se spremaju u padajući izbornik. Kada se napravi dupli klik na neko polje u unutrašnjosti matrice, otvara se padajući izbornik, te se klikne na neku od veza. Ta veza se zapiše u polje. Klikom na polje u kojem je upisana pojedina datoteka, ona se otvara, odnosno postavljen je *hyperlink*. Klikom na gumb “Prikaži vezu među podacima u matrici“, zatvara se forma, a otvara se radni list “Matrica 2“. Veze među podacima postavljaju se preko radnog lista “Relacije2“.



Slika 41: Kartica "Prikaz podataka i unos značajki"

	A	B	C	D	E	F	G	H	I	J	K	
1		ZAVRSNI SKLOP DIZALICE.SLDASM		SKLOP-A1.PDF	sklop kuke.SLDASM	sklop kuke_A2.PDF	sklop bubnja.SLDASM	SKLOP BUBNJA_A2.PDF	Prenosila-i-dizala.pdf	čeaona ploča.SLDPRRT	vijenac bubnja.SLDPRRT	
2	Dizalica							X				
3	SKLOP KUKE											
4	SKLOP BUBNJA	X				X						
5	Prenosila i dizala											
6	Čelna ploča 1											
7	Vijenac bubnja											
8												
9												
10												
11												
12												
13												

Slika 42: Veze podataka i datoteka u matrici

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE				
1		Dizalica	KATALOZI I UPUTE	SKLOP KUKE	GONJENI KOTAČ	POGONSKI KOTAČ	SKLOP BUBNJA	SKLOP MOSTA	SKLOP STUPA	SKRETNA UŽNICA 1	SKRETNA UŽNICA 2	OSOVINA UŠICE	OSIGURANJE OD VRTENJA	GUMENI GRANIČNIK	PLOČICA ZA PODESITI VISINU BUBNJA	MATICA M16	MATICA M20	MATICA M14	MATICA M12	PODLOŽNA PLOČICA 1	PODLOŽNA PLOČICA 2	PODLOŽNA PLOČICA 3	PODLOŽNA PLOČICA 4	VIJAK M14	VIJAK M20	VIJAK M10	NAVOJNA ŠPIKA	PODLOŽNA PLOČICA 5	MATICA M30	VIJAK S PODLOŠKOM M12	MATICA S PODLOŠKOM M12				
2	Dizalica																																		
3	KATALOZI I UPUTE																																		
4	SKLOP KUKE																																		
5	GONJENI KOTAČ																																		
6	POGONSKI KOTAČ																																		
7	SKLOP BUBNJA																																		
8	SKLOP MOSTA																																		
9	SKLOP STUPA																																		
10	SKRETNA UŽNICA 1																																		
11	SKRETNA UŽNICA 2																																		
12	OSOVINA UŠICE																																		
13	OSIGURANJE OD VRTENJA																																		
14	GUMENI GRANIČNIK																																		
15	PLOČICA ZA PODESITI VISINU BUBNJA																																		
16	MATICA M16																																		
17	MATICA M20																																		
18	MATICA M14																																		
19	MATICA M12																																		

Slika 43: Veze među podacima u matrici

## 4. ZAKLJUČAK

Programski jezik *Visual Basic „for applications“* alat je kojim se mogu programirati mnoge aplikacije koje služe kao pomagala u inženjerskom radu. Excel, kao dio najčešće korištenog uredskog alata Microsoft Office, posjeduje objekte kao što su radna knjiga, radni list, ćelija, tablice i mnogi drugi. VBA omogućuje manipulaciju tim objektima s puno više opcija nego što je to već realizirano u samom Excelu. Excel objekti posjeduju razne metode, svojstva i događaje koji programiranje čine jednostavnijim i lakše shvatljivim nego što je to u nekim drugim programskim jezicima. Željeni algoritam ostvaruje se pomoću objekata i naredbi.

Kroz studij smo se već u kolegijima vezanim za razvoj proizvoda susreli s mnogim prednostima matričnog prikaza među podacima. Takav prikaz omogućuje dobar pregled veza među podacima, sprječava moguće pogreške, upućuje na utjecaj jedne komponente na drugu te sam razvoj proizvoda čini efikasnijim.

Aplikacije poput Vizualizacije relacija mogu olakšati i ubrzati sam proces upravljanja i razvoja složenih projekata, s velikim brojem komponenata, velikim brojem datoteka vezanih na njih te mnogo ljudi odgovornih za određene funkcije projekta.

Glavni zadatak ovog rada bio je vizualizacija podataka koji su hijerarhijski strukturirani, što je zahtijevalo implementaciju kontrole koja omogućuje hijerarhijski prikaz u obliku stabla, tzv. *TreeView* kontrola. *TreeView* kontrola posjeduje objekte, čvorove, čija je manipulacija omogućena preko svojstava, metoda i događaja svojstvenih baš objektu čvor stabla (*Node*).



## **5. LITERATURA**

- [1] M. Hansen, Steven: Excel 2003 Programming with VBA, SYBEX, 2004.
- [2] Nastavni materijali s kolegija Napredna inženjerska informatika
- [3] <https://msdn.microsoft.com>

## **PRILOZI**

- I. Priložen kod
- II. CD-R disc

## Option Explicit

```
' obavezna deklaracija svih varijabli

' parametri aplikacije
' vrijednosti se postavljaju na radnom listu Postavke
' podrazumijevani naziv radnih listova koji
' pripadaju hijerarhijskoj strukturi za prikaz
Private sLevel As String
' broj razina
Private iLevels As Integer
' broj podataka za prikaz u strukturi
Private iNodes As Long
' mapa u koju spremamo odabrane datoteke
Private sFilePath As String
' naziv radnog lista za spremanje datoteka
Private sFiles As String
' naziv radnog lista za unos značajki
Private sProperties As String
' polje radnih listova s podacima za TreeView
Private arrDataWorkSheets() As String
' polja za manipulaciju svojstvima pojedinih node-ova
Private arrName() As String
Private arrKeyRoot() As String
Private arrKeyParent() As String
Private arrKeyChild() As String
' polje odabranih datoteka povezanih s podacima iz baze podataka
Private arrFiles() As String
Private iFiles As Integer
Private bAddFile As Boolean
Private bDelFile As Boolean

Private Sub cmdAddChild_Click()
'gumb za dodavanje podređenih podataka (Child)

    Dim sSheetName As String
    Dim sNextSheet As String
    Dim sText As String
    Dim iNodes As Long
    Dim sKeyParent As String
    Dim sKeyChild As String
    Dim sDefaultText As String
    Dim nParent As Node
    Dim i As Integer

    On Error Resume Next

    If TreeView2.SelectedItem Is Nothing Then
        Exit Sub
    End If
```

```

sKeyParent = TreeView2.SelectedItem.key
Set nParent = TreeView2.Nodes(sKeyParent)
sSheetName = TreeView2.SelectedItem.Tag
i = 0
Do While arrDataWorkSheets(i) <> sSheetName
    i = i + 1
Loop
sNextSheet = arrDataWorkSheets(i + 1)
sDefaultText = "Child"
ThisWorkbook.Worksheets(sNextSheet).Activate
iNodes = TreeView2.Nodes.Count
sText = Application.InputBox(Prompt:="Unesite podatak", Title:="Unos podataka u TreeView strukturu", Default:=sDefaultText &
Cnt(iNodes + 1))
If sText <> "False" Then
    sKeyChild = sNextSheet & "_" & sText
    TreeView2.Nodes.Add sKeyParent, tvwChild, sKeyChild, sText
    TreeView2.Nodes(iNodes + 1).Tag = sNextSheet
End If
End Sub

```

```

Private Sub cmdAddSheet_Click()
'dodavanje radnog lista
    Dim i As Long
    Dim ws0 As Worksheet
'dodavanje radnih listova iza zadnjega
    Call AddSheetAfterLast
    lstSheets.Clear
    i = 0
    For i = LBound(arrDataWorkSheets) To UBound(arrDataWorkSheets)
        lstSheets.AddItem (arrDataWorkSheets(i))
    Next
End Sub

```

```

Private Sub cmdCollapse_Click()
'prikaže sve Node-ove na TreeView1
Dim n As Node
    For Each n In TreeView1.Nodes
        n.Expanded = False
    Next
End Sub

```

```

Private Sub cmdCopy_Click()
' Kopiranje podataka iz odabranog radnog lista u drugi
'U InputBoxu se upisuje naziv radnog lista u koji kopiramo

    Dim SheetCopy As String
    Dim SheetPaste As String
    Dim shtNew As Worksheet
    Dim shtLast As Worksheet
    Dim Src As Worksheet

```

```
Dim Dst As Worksheet
Dim lastrow As Long, r As Range
Dim CopyRange As Range
Dim bSelected As Boolean
Dim i As Integer
Dim RetType As Integer
```

```
On Error Resume Next
```

```
bSelected = False
```

```
i = 0
```

```
For i = IstSheets.ListCount - 1 To 0 Step -1
```

```
    If bSelected = False Then
```

```
        bSelected = IstSheets.Selected(i)
```

```
        SheetCopy = IstSheets.List(i)
```

```
    End If
```

```
Next
```

```
If bSelected Then
```

```
    With ThisWorkbook
```

```
        Set shtLast = .Worksheets(.Worksheets.Count)
```

```
    End With
```

```
    'korištenje Excel inbutbox metode
```

```
    SheetPaste = Application.InputBox(Prompt:="Unesite ime radnog lista u koji želite kopirati.", Title:="Kopiranje podataka", _
```

```
    Default:=sLevel & CInt(iLevels - 1))
```

```
    If SheetPaste = "False" Then
```

```
        ' click na cancel
```

```
        Exit Sub
```

```
    End If
```

```
    'poruka za kopiranje/brisanje sadržaja radnog lista u koji se kopira
```

```
    RetType = MsgBox("Odabrali ste kopiranje iz radnog lista " & SheetCopy _
```

```
    & " u radni list " & SheetPaste & ". " & vbCrLf _
```

```
    & "Ako nastavite s kopiranjem, brišu se podaci u odredišnom radnom listu." _
```

```
    & vbCrLf & "Cancel za prekid kopiranja, Ok za nastavak ...", vbOKCancel, "Kopiranje radnog lista")
```

```
    If RetType = 2 Then
```

```
        ' click na Cancel
```

```
        Exit Sub
```

```
    Else
```

```
        ThisWorkbook.Worksheets(SheetPaste).UsedRange.Columns.Delete
```

```
        ThisWorkbook.Worksheets(SheetPaste).UsedRange.Rows.Delete
```

```
    End If
```

```
    ' ako odredišni radni list ne postoji, kreira se automatski sa upisanim nazivom
```

```
    If Not WorksheetExists(ThisWorkbook, SheetPaste) Then
```

```
        With ThisWorkbook
```

```
            'metoda za dodavanje radnog lista
```

```
            Set shtNew = Worksheets.Add(After:=shtLast)
```

```
            shtNew.Name = SheetPaste
```

```
        End With
```

```
    End If
```

```
    ' izvorišni radni list je neki od listova s podacima (bira se iz ListBoxa)
```

```
    Set Src = ThisWorkbook.Worksheets(SheetCopy)
```

```

' odredišni radni list je onaj kojeg zadamo u InputBox-u
Set Dst = ThisWorkbook.Worksheets(SheetPaste)
lastrow = Src.Cells(Cells.Rows.Count, "A").End(xlUp).Row
' kopiranje svih ispunjenih ćelija u rasponu
For Each r In Src.Range("A1:A" & lastrow)
    If r.Value <> "999999**" Then
        If CopyRange Is Nothing Then
            Set CopyRange = r.EntireRow
        Else
            Set CopyRange = Union(CopyRange, r.EntireRow)
        End If
    End If
Next r
If Not CopyRange Is Nothing Then
    CopyRange.Copy Dst.Range("A1")
End If
MsgBox "Uspješno obavljeno kopiranje iz radnog lista " & SheetCopy & " u radni list " & SheetPaste
Else
    MsgBox "Neuspješno kopiranje jer nije odabran radni list."
End If
End Sub

```

```

Private Sub cmdCopyIn_Click()
' Kopiranje podataka iz odabranog radnog lista u proizvoljni radni list, zadaje se u InputBoxu,
'u svrhu zaštite prije promjene podataka
    Dim SheetCopy As String
    Dim SheetPaste As String
    Dim Src As Worksheet
    Dim Dst As Worksheet
    Dim lastrow As Long
    Dim r As Range
    Dim CopyRange As Range
    Dim bSelected As Boolean
    Dim i As Integer
    Dim RetType As Integer

    On Error Resume Next

    bSelected = False
    i = 0
    For i = IstSheets.ListCount - 1 To 0 Step -1
        If bSelected = False Then
            bSelected = IstSheets.Selected(i)
            SheetPaste = IstSheets.List(i)
        End If
    Next
    If bSelected Then
        'korištenje Excel inbutbox metode
        ' radni list IZ kojeg želimo kopirati navodimo u InputBox-u
    End If
End Sub

```

```
SheetCopy = Application.InputBox(Prompt:="Unesite ime radnog lista IZ kojeg želite kopirati.", Title:="Kopiranje podataka",
```

```
Default:="LIST_X")
```

```
If SheetPaste = "False" Then
```

```
    ' click na cancel
```

```
    Exit Sub
```

```
End If
```

```
RetType = MsgBox("Odabrali ste kopiranje iz radnog lista " & SheetCopy _
```

```
& " u radni list " & SheetPaste & ". " & vbCrLf _
```

```
& "Ako nastavite s kopiranjem, brišu se podaci u odredišnom radnom listu." _
```

```
& vbCrLf & "Cancel za prekid kopiranja, Ok za nastavak ...", vbOKCancel, "Kopiranje radnog lista")
```

```
If RetType = 2 Then
```

```
    ' click na Cancel
```

```
    Exit Sub
```

```
Else
```

```
    'klik na OK u MsgBoxu, najprije se briše sadržaj na odredišnom radnom listu
```

```
    ThisWorkbook.Worksheets(SheetPaste).UsedRange.Columns.Delete
```

```
    ThisWorkbook.Worksheets(SheetPaste).UsedRange.Rows.Delete
```

```
End If
```

```
' izvorišni radni list je neki od listova s podacima
```

```
Set Src = ThisWorkbook.Worksheets(SheetCopy)
```

```
' odredišni radni list je onaj koji je odabran u ListBox-u
```

```
Set Dst = ThisWorkbook.Worksheets(SheetPaste)
```

```
lastrow = Src.Cells(Cells.Rows.Count, "A").End(xlUp).Row
```

```
' kopiranje svih ispunjenih ćelija u rasponu
```

```
For Each r In Src.Range("A1:A" & lastrow)
```

```
    If r.Value <> "9999999**" Then
```

```
        If CopyRange Is Nothing Then
```

```
            Set CopyRange = r.EntireRow
```

```
        Else
```

```
            Set CopyRange = Union(CopyRange, r.EntireRow)
```

```
        End If
```

```
    End If
```

```
Next r
```

```
If Not CopyRange Is Nothing Then
```

```
    CopyRange.Copy Dst.Range("A1")
```

```
End If
```

```
MsgBox "Uspješno obavljeno kopiranje iz radnog lista " & SheetCopy & " u radni list " & SheetPaste
```

```
Else
```

```
    MsgBox "Neuspješno kopiranje jer nije odabran radni list."
```

```
End If
```

```
End Sub
```

```
Private Sub cmdDodajDat_Click()
```

```
    'dodavanje datoteka za podatak koji je odabran u TreeViewu
```

```
    ' dohvati lokaciju foldera za spremanje datoteka
```

```
    'kopija datoteka se sprema u mapu koja je unaprijed definirana kao mapa s datotekama projekta
```

```
Dim FilePath As String
```

```
Dim fName As String
```

```
Dim fName1 As String
```

```
Dim iFiles As Integer
Dim i As Integer
Dim k As Integer
Dim Return Type As Integer
Dim SlashPos As Integer
Dim bAdded As Boolean
```

```
bAdded = True
```

```
FilePath = NavigateFromWorkBookPath()
```

```
FilePath = FilePath & sFilePath
```

```
'dijalog za izbor datoteka
```

```
With Application.FileDialog(msoFileDialogOpen)
```

```
    'inicijalizira, postavlja putanju s kojom hoćemo početi
```

```
    .InitialFileName = FilePath
```

```
    .Filters.Clear
```

```
    'omogućuje izbor datoteke bilo kojeg formata
```

```
    .Filters.Add "All Files", "*.*"
```

```
    .AllowMultiSelect = True
```

```
End With
```

```
' google
```

```
' prikaži dijalog za izbor datoteka
```

```
iFiles = Application.FileDialog(msoFileDialogOpen).Show
```

```
' odabrane datoteke zapamti u polje arrFiles,uzimaju se u obzir i već postojeće datoteke
```

```
'omogućen je izbor više datoteka odjednom
```

```
If iFiles <> 0 Then
```

```
    ReDim arrFiles(1 To Application.FileDialog(msoFileDialogOpen).SelectedItems.Count + IstDatoteke2.ListCount)
```

```
    i = 1
```

```
    For i = 1 To Application.FileDialog(msoFileDialogOpen).SelectedItems.Count
```

```
        fName = Application.FileDialog(msoFileDialogOpen).SelectedItems(i)
```

```
        If fName <> "" Then
```

```
            'InstrRev očita poziciju znaka,potraži ga u stringu,a Mid je , počevši od neke
```

```
            'pozicije do kraja uzmi dio stringa
```

```
            SlashPos = InstrRev(fName, "\")
```

```
            fName1 = Mid(fName, SlashPos + 1)
```

```
            k = 0
```

```
            bAdded = True
```

```
            ' dodajemo datoteku na popis samo ako već nije prije dodana
```

```
            For k = 0 To IstDatoteke2.ListCount - 1
```

```
                If IstDatoteke2.List(k) = fName1 Then
```

```
                    bAdded = False
```

```
                    arrFiles(k + 1) = ""
```

```
                End If
```

```
            Next
```

```
            If bAdded Then
```

```
                bAddFile = True
```

```
                arrFiles(k + 1) = fName
```

```
                IstDatoteke2.AddItem (fName1)
```

```
            End If
```

```
        End If
```



Next

End If

End Sub

Private Sub cmdDodajPodatak\_Click()

'dodavanje podataka na istoj razini s odabranim čvorom u TreeView

Dim sSheetName As String

Dim sNextSheet As String

Dim sText As String

Dim iNodes As Long

Dim sKey As String

Dim sKeyParent As String

Dim sKeyChild As String

Dim sDefaultText As String

Dim nParent As Node

Dim i As Integer

Dim bRoot As Boolean

On Error Resume Next

If TreeView2.SelectedItem Is Nothing Then

bRoot = True

sKey = sLevel & "0"

sSheetName = sLevel & "0"

sDefaultText = "Root"

Else

bRoot = False

sKey = TreeView2.SelectedItem.key

sSheetName = TreeView2.SelectedItem.Tag

Set nParent = TreeView2.Nodes(sKey).Parent

sKeyParent = nParent.key

sDefaultText = "Child"

End If

ThisWorkbook.Worksheets(sSheetName).Activate

'broji čvorove

iNodes = TreeView2.Nodes.Count

'InputBox za unos podatka koji dodajemo u TreeView

sText = Application.InputBox(Prompt:="Unesite podatak", Title:="Unos podataka u TreeView strukturu", Default:=sDefaultText &

CInt(iNodes + 1))

If sText <> "False" Then

If bRoot Then

'dodavanje korijenskog čvora

TreeView2.Nodes.Add key:=sKey & "\_" & Trim(sText), text:=Trim(sText)

TreeView2.Nodes(iNodes + 1).Tag = sSheetName

Else

'dodavanje Child čvora

sKeyChild = sSheetName & "\_" & sText

TreeView2.Nodes.Add sKeyParent, twwChild, sKeyChild, sText

TreeView2.Nodes(iNodes + 1).Tag = sSheetName

End If

End If

End Sub

Private Sub cmdDodajRoot\_Click()

'gumb za dodavanje korijenskog čvora

'ima drugačije parametre nego dodavanje Child čvora

Dim c As Range

Dim sText As String

Dim iNodes As Long

On Error Resume Next

ThisWorkbook.Worksheets(sLevel & "0").Activate

iNodes = TreeView2.Nodes.Count

sText = Application.InputBox(Prompt:="Unesite podatak", Title:="Unos podataka u TreeView strukturu", Default:="ROOT" &

Clnt(iNodes + 1))

TreeView2.Nodes.Add key:=sLevel & "0" & "\_" & Trim(sText), text:=Trim(sText)

TreeView2.Nodes(iNodes + 1).Tag = sLevel & "0"

End Sub

Private Sub cmdExpand\_Click()

'prikaže sve čvorove (raširi)

Dim n As Node

For Each n In TreeView1.Nodes

n.Expanded = True

Next

End Sub

Private Sub cmdIzlaz\_Click()

'izbaci formu iz memorije

Unload Me

End Sub

Private Sub cmdRemoveSheet\_Click()

'brisanje radnog lista koji se odabere u ListBoxu

Dim bDeleted As Boolean

Dim ReturnTyp As Integer

Dim i As Integer

If lstSheets.ListIndex = -1 Then

MsgBox "Odaberite barem jedan radni list za brisanje!"

Else

bDeleted = False

For i = lstSheets.ListCount - 1 To 0 Step -1

If lstSheets.Selected(i) = True Then

If ThisWorkbook.Worksheets(lstSheets.List(i)).Range("A1") = "" Then

bDeleted = DeleteSheet(ThisWorkbook.Worksheets(lstSheets.List(i)), True)

lstSheets.RemoveItem (i)

Else

ReturnTyp = MsgBox("Radni list ima sadržaj. Želite li ipak obrisati radni list? ", vbYesNoCancel + vbExclamation, "Potvrda brisanja")

```

        If Return Type = 6 Then
            bDeleted = DeleteSheet(ThisWorkbook.Worksheets(IstSheets.List(i)), True)
            IstSheets.RemoveItem (i)
        End If
    End If
End If
Next
End If
ReDim arrDataWorkSheets(0 To iLevels)
'ponovno napuni polje sa nazivima radnih listovima, nakon brisanja
Call PuniArrWorkSheets
End Sub

```

```
Private Sub cmdShowMatrix2_Click()
```

```
'prikaz matrice, podaci iz TreeView kontrole su u zaglavlju redaka i stupaca
```

```

Dim MainCell As Range
Dim MainCellM As Range
Dim i As Long
Dim j As Long
Dim k As Long
Dim nRow As Long
Dim nCol As Long
Dim TNode As Node

```

```
nRow = 0
```

```
nCol = 0
```

```
i = 0
```

```
j = 0
```

```
k = 0
```

```
' matrica će se prikazati tek kad se cijela ispuni i formatira
```

```
Application.ScreenUpdating = False
```

```
ThisWorkbook.Worksheets("Matrica2").UsedRange.Columns.Delete
```

```
ThisWorkbook.Worksheets("Matrica2").UsedRange.Rows.Delete
```

```
'dodavanje podataka u prvi red u radni list matrica2
```

```
' ćelija od koje kreće popunjavanje redaka i kolona
```

```
Set MainCellM = ThisWorkbook.Worksheets("Matrica2").Range("B1:B1")
```

```
' preuzimanje naziva datoteka iz radnog lista Datoteke
```

```
For Each TNode In TreeView1.Nodes
```

```
' i-ti redak, 1. stupac - uzimamo tekst podataka koji su upisani u Treeview strukturi
```

```
MainCellM.Offset(i + 1, -1).Value = TNode.text
```

```
'formatiranje teksta - font
```

```
With MainCellM.Offset(i + 1, -1).Font
```

```
.Name = "Calibri"
```

```
.Size = 12
```

```
.Underline = xlUnderlineStyleNone
```

```
.ThemeColor = xlThemeColorLight1
```

```
.TintAndShade = 0
```

```
.ThemeFont = xlThemeFontNone
```

```
End With
```

'Bojanje ćelije

```
With MainCellM.Offset(i + 1, -1).Interior  
    .Color = Worksheets("Postavke").Range("A7").Interior.Color
```

End With

'obrub

```
MainCellM.Offset(i + 1, -1).Borders(xlDiagonalDown).LineStyle = xlNone  
MainCellM.Offset(i + 1, -1).Borders(xlDiagonalUp).LineStyle = xlNone
```

```
With MainCellM.Offset(i + 1, -1).Borders(xlEdgeLeft)
```

```
    .LineStyle = xlContinuous  
    .ColorIndex = 0  
    .TintAndShade = 0  
    .Weight = xlThin
```

End With

```
With MainCellM.Offset(i + 1, -1).Borders(xlEdgeTop)
```

```
    .LineStyle = xlContinuous  
    .ColorIndex = 0  
    .TintAndShade = 0  
    .Weight = xlThin
```

End With

```
With MainCellM.Offset(i + 1, -1).Borders(xlEdgeBottom)
```

```
    .LineStyle = xlContinuous  
    .ColorIndex = 0  
    .TintAndShade = 0  
    .Weight = xlThin
```

End With

```
With MainCellM.Offset(i + 1, -1).Borders(xlEdgeRight)
```

```
    .LineStyle = xlContinuous  
    .ColorIndex = 0  
    .TintAndShade = 0  
    .Weight = xlThin
```

End With

```
With MainCellM.Offset(i + 1, -1).Borders(xlInsideVertical)
```

```
    .LineStyle = xlContinuous  
    .ColorIndex = 0  
    .TintAndShade = 0  
    .Weight = xlThin
```

End With

```
With MainCellM.Offset(i + 1, -1).Borders(xlInsideHorizontal)
```

```
    .LineStyle = xlContinuous  
    .ColorIndex = 0  
    .TintAndShade = 0  
    .Weight = xlThin
```

End With

'u prvi stupac u radnom listu matrica punimo drugi stupac iz radnog lista datoteke

```
MainCellM.Offset(0, k).Value = TNode.text
```

'formatiranje teksta

```
With MainCellM.Offset(0, k).Font
```

```
    .Name = "Calibri"  
    .Size = 12  
    .Underline = xlUnderlineStyleNone
```

```

.ThemeColor = xlThemeColorLight1
.TintAndShade = 0
.ThemeFont = xlThemeFontNone
End With
'Bojanje ćelije
With MainCellM.Offset(0, k).Interior
.Color = Worksheets("Postavke").Range("A7").Interior.Color
End With
'Poravnanje ćelija
With MainCellM.Offset(0, k)
.HorizontalAlignment = xlCenter
.VerticalAlignment = xlBottom
.WrapText = False
.Orientation = 90
.AddIndent = False
.IndentLevel = 0
.ShrinkToFit = False
.ReadingOrder = xlContext
.MergeCells = False
End With
'obrub
MainCellM.Offset(0, k).Borders(xlDiagonalDown).LineStyle = xlNone
MainCellM.Offset(0, k).Borders(xlDiagonalUp).LineStyle = xlNone
With MainCellM.Offset(0, k).Borders(xlEdgeLeft)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With MainCellM.Offset(0, k).Borders(xlEdgeTop)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With MainCellM.Offset(0, k).Borders(xlEdgeBottom)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With MainCellM.Offset(0, k).Borders(xlEdgeRight)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With MainCellM.Offset(0, k).Borders(xlInsideVertical)
.LineStyle = xlContinuous
.ColorIndex = 0

```

```

        .TintAndShade = 0
        .Weight = xlThin
    End With
    With MainCellM.Offset(0, k).Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    'broj ispunjenih kolona u matrici
    nCol = nCol + 1
    k = k + 1
    'broj ispunjenih redaka u matrici
    nRow = nRow + 1
    i = i + 1
Next
'uređivanje dijela matrice u kojem se definiraju relacije
Dim relationRange As Range
Set relationRange = Range(MainCellM.Offset(1, 0), MainCellM.Offset(nRow, nCol - 1))
If nCol > 1 Then
    'bojanje unutrašnjosti matrice
    relationRange.Interior.Color = Worksheets("Postavke").Range("A8:A8").Interior.Color
    'obrub
    relationRange.Borders(xlDiagonalDown).LineStyle = xlNone
    relationRange.Borders(xlDiagonalUp).LineStyle = xlNone
    With relationRange.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With relationRange.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With relationRange.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With relationRange.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With relationRange.Borders(xlInsideVertical)

```

```
.LineStyle = xlContinuous
```

```
.ColorIndex = 0
```

```
.TintAndShade = 0
```

```
.Weight = xlThin
```

```
End With
```

```
With relationRange.Borders(xlInsideHorizontal)
```

```
.LineStyle = xlContinuous
```

```
.ColorIndex = 0
```

```
.TintAndShade = 0
```

```
.Weight = xlThin
```

```
End With
```

```
End If
```

```
'poravnanje prema najdužem tekstu u retcima/kolonama
```

```
ThisWorkbook.Worksheets("Matrica2").UsedRange.Columns.AutoFit
```

```
ThisWorkbook.Worksheets("Matrica2").UsedRange.Rows.AutoFit
```

```
'formatiranje matrice dodati
```

```
ThisWorkbook.Worksheets("Matrica2").Activate
```

```
Application.ScreenUpdating = True
```

```
'izlaz iz aplikacije radi prikaza radnog lista s matricom (izbaci formu iz memorije)
```

```
Unload Me
```

```
End Sub
```

```
Private Sub cmdShowMatrix_Click()
```

```
' punimo matricu u kojoj se prikazuju veze između podataka iz TreeViewa i pripadnih datoteka
```

```
Dim MainCellID As Range
```

```
Dim MainCellIM As Range
```

```
Dim i As Long
```

```
Dim j As Long
```

```
Dim k As Long
```

```
Dim sPath As String
```

```
Dim sText As String
```

```
Dim nRow As Long
```

```
Dim nCol As Long
```

```
nRow = 0
```

```
nCol = 0
```

```
i = 0
```

```
j = 0
```

```
k = 0
```

```
' matrica će se prikazati tek kad se cijela ispuni i formatira
```

```
Application.ScreenUpdating = False
```

```
ThisWorkbook.Worksheets("Matrica").UsedRange.Columns.Delete
```

```
ThisWorkbook.Worksheets("Matrica").UsedRange.Rows.Delete
```

```
'dodavanje datoteka u prvi red u radni list matrica
```

```
'vanjska petlja je kretanje po redovima po radnom listu sFiles
```

```
'unutarnja petlja je kretanje po stupcima za pojedini red po radnom listu sFiles
```

```
'ćelija od koje preuzimamo nazive datoteka
```

```
Set MainCellID = ThisWorkbook.Worksheets(sFiles).Range("C1:C1")
```

```
' ćelija od koje kreće popunjavanje redaka i kolona
```

```
Set MainCellIM = ThisWorkbook.Worksheets("Matrica").Range("B1:B1")
```

```

'preuzimanje naziva datoteka iz radnog lista Datoteke
Do While MainCellD.Offset(i, 0).Value <> ""
    'i-ti redak, 1. stupac - uzimamo tekst podataka koji su upisani u Treeview strukturi
    MainCellM.Offset(i + 1, -1).Value = MainCellD.Offset(i, -1).Value
    'formatiranje teksta - font
    With MainCellM.Offset(i + 1, -1).Font
        .Name = "Calibri"
        .Size = 12
        .Underline = xlUnderlineStyleNone
        .ThemeColor = xlThemeColorLight1
        .TintAndShade = 0
        .ThemeFont = xlThemeFontNone
    End With
    'Bojanje ćelije
    With MainCellM.Offset(i + 1, -1).Interior
        .Color = Worksheets("Postavke").Range("A7").Interior.Color
    End With
    'obrub
    MainCellM.Offset(i + 1, -1).Borders(xlDiagonalDown).LineStyle = xlNone
    MainCellM.Offset(i + 1, -1).Borders(xlDiagonalUp).LineStyle = xlNone
    With MainCellM.Offset(i + 1, -1).Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With MainCellM.Offset(i + 1, -1).Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With MainCellM.Offset(i + 1, -1).Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With MainCellM.Offset(i + 1, -1).Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With MainCellM.Offset(i + 1, -1).Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With

```



```

With MainCellM.Offset(i + 1, -1).Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
'u prvi stupac u radnom listu matrica punimo drugi stupac iz radnog lista datoteke
j = 0
Do While MainCellD.Offset(i, j).Value <> ""
    MainCellM.Offset(0, k).Value = MainCellD.Offset(i, j).Value
    sPath = sFilePath & "\" & MainCellD.Offset(i, j).Value
    sText = MainCellD.Offset(i, j).Value
    'dodavanje hiperlinka
    ThisWorkbook.Worksheets("Matrica").Hyperlinks.Add Anchor:=MainCellM.Offset(0, k), Address:=sPath, _
    TextToDisplay:=sText
    'formatiranje teksta
    With MainCellM.Offset(0, k).Font
        .Name = "Calibri"
        .Size = 12
        .Underline = xlUnderlineStyleNone
        .ThemeColor = xlThemeColorLight1
        .TintAndShade = 0
        .ThemeFont = xlThemeFontNone
    End With
    'Bojanje ćelije
    With MainCellM.Offset(0, k).Interior
        .Color = Worksheets("Postavke").Range("A7").Interior.Color
    End With
    'Poravnanje ćelija
    With MainCellM.Offset(0, k)
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlBottom
        .WrapText = False
        .Orientation = 90
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
    End With
    'obrub
    MainCellM.Offset(0, k).Borders(xlDiagonalDown).LineStyle = xlNone
    MainCellM.Offset(0, k).Borders(xlDiagonalUp).LineStyle = xlNone
    With MainCellM.Offset(0, k).Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With MainCellM.Offset(0, k).Borders(xlEdgeTop)

```

```

.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With MainCellM.Offset(0, k).Borders(xlEdgeBottom)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With MainCellM.Offset(0, k).Borders(xlEdgeRight)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With MainCellM.Offset(0, k).Borders(xlInsideVertical)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With MainCellM.Offset(0, k).Borders(xlInsideHorizontal)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
'broj ispunjenih kolona u matrici
nCol = nCol + 1
k = k + 1
j = j + 1
Loop
'broj ispunjenih redaka u matrici
nRow = nRow + 1
i = i + 1
Loop
'uređivanje dijela matrice u kojem se definiraju relacije
Dim relationRange As Range
Set relationRange = Range(MainCellM.Offset(1, 0), MainCellM.Offset(nRow, nCol - 1))
If nCol > 1 Then
'bojanje unutrašnjosti matrice
relationRange.Interior.Color = Worksheets("Postavke").Range("A8:A8").Interior.Color
'obrub
relationRange.Borders(xlDiagonalDown).LineStyle = xlNone
relationRange.Borders(xlDiagonalUp).LineStyle = xlNone
With relationRange.Borders(xlEdgeLeft)
.LineStyle = xlContinuous
.ColorIndex = 0

```

.TintAndShade = 0

.Weight = xlThin

End With

With relationRange.Borders(xlEdgeTop)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With relationRange.Borders(xlEdgeBottom)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With relationRange.Borders(xlEdgeRight)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With relationRange.Borders(xlInsideVertical)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

With relationRange.Borders(xlInsideHorizontal)

.LineStyle = xlContinuous

.ColorIndex = 0

.TintAndShade = 0

.Weight = xlThin

End With

End If

'poravnanje

ThisWorkbook.Worksheets("Matrica").UsedRange.Columns.AutoFit

ThisWorkbook.Worksheets("Matrica").UsedRange.Rows.AutoFit

'formatiranje matrice dodati

ThisWorkbook.Worksheets("Matrica").Activate

Application.ScreenUpdating = True

'izlaz iz aplikacije radi prikaza radnog lista s matricom (izbaci formu iz memorije)

Unload Me

End Sub

Private Sub cmdSpremiDat\_Click()

'spremanje datoteka u radni list i kopiranje u odgovarajuću mapu

Dim lastrow As Long

Dim i As Long

Dim j As Long

```
Dim k As Integer
Dim MainCell As Range
Dim Repeat As Boolean
Dim RetType As Integer
Dim brojDatoteka As Integer
Dim SourceFile As String
Dim DestinationFile As String
Dim rng As Range
Repeat = True
```

```
Set MainCell = ThisWorkbook.Worksheets(sFiles).Range("A1:A1")
' najprije tražimo da li odabrani čvor već ima redak sa datotekama
' ako ima ažuriramo taj redak, a ako nema tražimo prvi slobodni red
'u 1. stupac zapisujemo ključ noda, u drugi stupac tekst
'od treće kolone nadalje zapisujemo datoteke
i = 0
Do While MainCell.Offset(i, 0).Value <> "" And Repeat
    If MainCell.Offset(i, 0).Value = TreeView2.SelectedItem.key Then
        ' dodaj ključ i text iz odabranog node-a
        MainCell.Offset(i, 0).Value = TreeView2.SelectedItem.key
        MainCell.Offset(i, 1).Value = TreeView2.SelectedItem.text
        ' najprije obriši red
        'ThisWorkbook.Worksheets("Datoteke").Rows(i).ClearContents
        j = 2
        Do While MainCell.Offset(i, j).Value <> ""
            MainCell.Offset(i, j).Delete
        Loop
        Repeat = False
        k = 2
        j = 0
        'kretanje po retku koji sadrži ključ odabranog Node-a
        ' dodavanje naziva datoteka u redak
        For j = 0 To IstDatoteke2.ListCount - 1
            MainCell.Offset(i, k + j).Value = IstDatoteke2.List(j)
        Next
        ' ako nema odabranih datoteka - brišemo i iz radnog lista
        'datoteke koje su prije bile u njemu
        If IstDatoteke2.ListCount = 0 Then
            j = 2
            Do While MainCell.Offset(i, j).Value <> ""
                MainCell.Offset(i, j).Delete
            Loop
        End If
    End If
    i = i + 1
Loop
' ako još nisu upisane datoteke za odabrani red
' upisujemo ih u prvi prazni red
If Repeat Then
    If IstDatoteke2.ListCount > 0 Then
```

```

MainCell.Offset(i, 0).Value = TreeView2.SelectedItem.key
MainCell.Offset(i, 1).Value = TreeView2.SelectedItem.text
k = 2
j = 0
For j = 0 To lstDatoteke2.ListCount - 1
    MainCell.Offset(i, k + j).Value = lstDatoteke2.List(j)
Next
End If

```

```
End If
```

```
'kopiranje odabranih datoteka u mapu projekta
```

```
i = 0
```

```
brojDatoteka = 0
```

```
If bAddFile = False And bDelFile = False Then
```

```
    MsgBox "Niste napravili nikakvu promjenu u popisu datoteka!"
```

```
Elseif bAddFile = False And bDelFile = True Then
```

```
    MsgBox "Odabrane datoteke su uklonjene iz radnog lista!"
```

```
Elseif bAddFile = True Then
```

```
    For i = 0 To lstDatoteke2.ListCount - 1
```

```
        SourceFile = arrFiles(i + 1)
```

```
        DestinationFile = ThisWorkbook.path & "\" & sFilePath & "\" & lstDatoteke2.List(i)
```

```
        If SourceFile <> "" And SourceFile <> DestinationFile Then
```

```
            brojDatoteka = brojDatoteka + 1
```

```
            'ak su putanje različite, nemamo u odredištu odabranu sliku, stavljamo ju, Filecopy kopira tam
```

```
            'di je sad i stavlja u images
```

```
            FileCopy SourceFile, DestinationFile
```

```
            'informacija da je dodana datoteka
```

```
        End If
```

```
    Next
```

```
End If
```

```
If brojDatoteka > 0 And bDelFile = False Then
```

```
    RetType = MsgBox("Dodane su nove datoteke za podatak: " _
```

```
    & TreeView2.SelectedItem.text & vbCrLf & " u mapu " & ThisWorkbook.path & "\" _
```

```
    & sFilePath & vbCrLf & " Broj kopiranih datoteka: " & Cint(brojDatoteka), vbOKOnly, "Dodavanje datoteka")
```

```
End If
```

```
' poravnanje svih kolona
```

```
ThisWorkbook.Worksheets(sFiles).UsedRange.Columns.AutoFit
```

```
End Sub
```

```
Private Sub cmdSpremiZnacajke_Click()
```

```
'sprema značajke za podatak selektiran u TreeView u radni list Značajke
```

```
'značajke se unose u TextBox
```

```
    Dim i As Long
```

```
    Dim k As Long
```

```
    Dim MainCell As Range
```

```
    Dim Repeat As Boolean
```

```
    Repeat = True
```

```
Set MainCell = ThisWorkbook.Worksheets(sProperties).Range("A1:A1")
```

```
' najprije tražimo da li odabrani čvor već ima redak sa značajkama
```

```

' ako ima ažuriramo taj redak, a ako nema tražimo prvi slobodni red
'u 1. stupac zapisujemo ključ čvora, u drugi stupac tekst
'u treći stupac zapisujemo značajku
i = 0
' prikaz ključa odabranog čvora u Immediate prozoru VBE editora
Debug.Print TreeView1.SelectedItem.key
Do While MainCell.Offset(i, 0).Value <> "" And Repeat
    If MainCell.Offset(i, 0).Value = TreeView1.SelectedItem.key Then
        MainCell.Offset(i, 0).Value = TreeView1.SelectedItem.key
        MainCell.Offset(i, 1).Value = TreeView1.SelectedItem.text
        Repeat = False
        k = 2
        If txtZnacajke.text = "" Then
            ' ako nema odabranih datoteka - brišemo i iz radnog lista
            MainCell.Offset(i, k).Delete
        Else
            ' kretanje po retku koji sadrži ključ odabranog Node-a
            ' dodavanje značajke
            MainCell.Offset(i, k).Value = txtZnacajke.text
        End If
    End If
    i = i + 1
Loop
' ako još nisu upisane datoteke za odabrani red
' upisujemo ih u prvi prazni red
If Repeat Then
    MainCell.Offset(i, 0).Value = TreeView1.SelectedItem.key
    MainCell.Offset(i, 1).Value = TreeView1.SelectedItem.text
    k = 2
    If txtZnacajke.text = "" Then
        ' ako nema odabranih datoteka - brišemo i iz radnog lista
        MainCell.Offset(i, k).Delete
    Else
        ' kretanje po retku koji sadrži ključ odabranog Node-a
        ' dodavanje značajke
        MainCell.Offset(i, k).Value = txtZnacajke.text
    End If
End If
' poravnanje svih kolona
ThisWorkbook.Worksheets(sProperties).UsedRange.Columns.AutoFit
End Sub

```

```

Private Sub cmdUkloniDat_Click()
' Briše odabrane datoteke iz ListBoxa
bDelFile = False
Dim i As Integer
For i = IstDatoteke2.ListCount - 1 To 0 Step -1
    If IstDatoteke2.Selected(i) = True Then
        bDelFile = True
        IstDatoteke2.RemoveItem (i)
    End If
Next i

```

End If

Next

End Sub

Private Sub cmdUkloniPodatak\_Click()

'uklanja čvor iz TreeView kontrole

Dim nNode As Node

Set nNode = TreeView2.SelectedItem

If nNode.key <> "" Then

TreeView2.Nodes.Remove nNode.key

End If

End Sub

Private Sub cmdUkloniSve\_Click()

'briše sve datoteke iz listboxa

bDelFile = True

IstDatoteke2.Clear

End Sub

Private Sub MultiPage1\_Change()

'inicijalizacija ListBoxa s popisom datoteka

IstDatoteke1.Clear

IstDatoteke2.Clear

End Sub

Private Sub TreeView1\_NodeClick(ByVal Node As MSComctlLib.Node)

'na klik na podatak iz TreeViewa inicijaliziraju se dodatne kontrole u kojima

' prikazujemo značajke, inicijalizacija labela i ListBoxa prije nego u

'TreeView klikne na neki podatak

Dim MainCell As Range

Dim i As Integer

Dim j As Integer

Dim Repeat As Boolean

Repeat = True

i = 0

j = 2

IstDatoteke1.Clear

lblNodeZnacajke.Caption = ""

lblNodeDatoteke.Caption = ""

txtZnacajke.text = ""

cmdSpremiZnacajke.Enabled = True

'pamti odabrani čvor

lblNodeZnacajke.Caption = TreeView1.SelectedItem.text

lblNodeDatoteke.Caption = TreeView1.SelectedItem.text

Set MainCell = ThisWorkbook.Worksheets(sFiles).Range("A1:A1")

'traži redak na radnom listu datoteke koji odgovara ključu odabranog čvora

Do While MainCell.Offset(i, 0).Value <> "" And Repeat

If MainCell.Offset(i, 0).Value = Node.key Then

```

Repeat = False
j = 2
'kretanje po retku koji sadrži ključ odabranog Node-a
'1. stupac je Node.Key, drugi sadrži Node.text, od 3. nadalje su stupci
's nazivima datoteka
Do Until MainCell.Offset(i, j).Value = ""
    IstDatoteke1.AddItem (MainCell.Offset(i, j).Value)
    j = j + 1
Loop
End If
i = i + 1
Loop
'traži redak na radnom listu značajke koji odgovara ključu odabranog čvora
i = 0
Repeat = True
Set MainCell = ThisWorkbook.Worksheets(sProperties).Range("A1:A1")
Do While MainCell.Offset(i, 0).Value <> "" And Repeat
    If MainCell.Offset(i, 0).Value = Node.key Then
        Repeat = False
        'kretanje po retku koji sadrži ključ odabranog Node-a
        '1. stupac je Node.Key, drugi sadrži Node.text, 3. stupac = značajka
        txtZnacajke.text = MainCell.Offset(i, 2).Value
    End If
    i = i + 1
Loop
End Sub

```

```

Private Sub TreeView2_NodeClick(ByVal Node As MSCComctlLib.Node)
'na klik na podatak iz TreeViewa inicijaliziraju se dodatne kontrole u kojima
' prikazujemo značajke, inicijalizacija labela i ListBoxa prije nego u
'TreeView klikne na neki podatak
'na TreeView2 imamo i dodatne akcije na gumbе
    Dim MainCell As Range
    Dim i As Integer
    Dim j As Integer
    Dim Repeat As Boolean
    Repeat = True
    i = 0
    j = 2
    IstDatoteke2.Clear
    lblNodeName.Caption = ""
    cmdSpremiPodatke.Enabled = True
    Set MainCell = ThisWorkbook.Worksheets(sFiles).Range("A1:A1")
    'traži redak koji odgovara ključu odabranog Node-a
    Debug.Print Node.key
    Do While MainCell.Offset(i, 0).Value <> "" And Repeat
        If MainCell.Offset(i, 0).Value = Node.key Then
            Repeat = False
            j = 2

```



```

'kretanje po retku koji sadrži ključ odabranog Node-a
Do Until MainCell.Offset(i, j).Value = ""
    IstDatoteke2.AddItem (MainCell.Offset(i, j).Value)
    j = j + 1
Loop
End If
i = i + 1
Loop
'nakon odabira čvora omogućuje se klik na gumbе za manipulaciju
'podacima iz TreeView2
cmdDodajDat.Enabled = True
cmdSpremiDat.Enabled = True
cmdUkloniDat.Enabled = True
cmdUkloniSve.Enabled = True
'prikaz teksta odabranog čvora u labeli uz ListBox za dodavanje
'datoteka
lblNodeName.Caption = TreeView2.SelectedItem.text
End Sub

```

```

Private Sub UserForm_Initialize()
'Deklaracija varijabli
'objektne varijable - za pretraživanje svih listova koji sadrže podatke za Treeview
'trenutno aktivni
Dim wks As Worksheet
' prethodni
Dim wks0 As Worksheet
' slijedeći
Dim wks1 As Worksheet
' puni polje na razini modula
Call PuniArrWorkSheets
'polje s nazivima radnih listova
'Veličina polja s nazivima radnih listova se dinamički mijenja ovisno o popunjenosti
'radnih listova preko funkcija za određivanje broja popunjenih stupaca i redaka
Dim arrWorksheet() As String
Dim rg As Range
Dim nNode As Node
' pomoćne varijable - brojači
Dim i As Long
Dim counter As Long
Dim k As Long
Dim lastrow As Long
Dim NumberOfColumns As Long
' ako se dogodi greška
On Error Resume Next
' inicijalizacija varijabli
' private varijable na razini forme UserForm1
'naziv radnih listova s podacima koji pripadaju projektu
If IsEmpty(ThisWorkbook.Worksheets("Postavke").Range("A1")) Then
    sLevel = "RAZINA_"

```

```

Else
    sLevel = Trim(ThisWorkbook.Worksheets("Postavke").Range("A1") + "_")
End If
'maksimalni broj razina
If IsEmpty(ThisWorkbook.Worksheets("Postavke").Range("A2")) Then
    iLevels = 10
Else
    iLevels = CInt(ThisWorkbook.Worksheets("Postavke").Range("A2")) - 1
End If
'maksimalni broj čvorova
If IsEmpty(ThisWorkbook.Worksheets("Postavke").Range("A3")) Then
    iNodes = 500
Else
    iNodes = CInt(ThisWorkbook.Worksheets("Postavke").Range("A3")) - 1
End If
'mapa za spremanje datoteka
If IsEmpty(ThisWorkbook.Worksheets("Postavke").Range("A4")) Then
    sFilePath = "Datoteke projekta"
Else
    sFilePath = Trim(ThisWorkbook.Worksheets("Postavke").Range("A4"))
End If
'naziv radnog lista u koji spremamo nazive datoteka
If IsEmpty(ThisWorkbook.Worksheets("Postavke").Range("A5")) Then
    sFiles = "Datoteke"
Else
    sFiles = Trim(ThisWorkbook.Worksheets("Postavke").Range("A5"))
End If
'radni list za unos značajki
If IsEmpty(ThisWorkbook.Worksheets("Postavke").Range("A6")) Then
    sProperties = "Značajke"
Else
    sProperties = Trim(ThisWorkbook.Worksheets("Postavke").Range("A6"))
End If
' lokalno deklarirano polje radnih listova s podacima za treeview kontrolu
ReDim arrWorksheet(0 To iLevels)
' polja za manipulaciju svojstvima pojedinih node-ova
ReDim arrName(0 To iNodes)
ReDim arrKeyRoot(0 To iNodes)
ReDim arrKeyParent(0 To iNodes)
ReDim arrKeyChild(0 To iNodes)
' zastavica - indikator za spremanje datoteka - spremaju se samo nove
bAddFile = False
bDelFile = False
' postavi se na 1. tab tj. prikaz podataka u TreeView1
UserForm1.MultiPage1.Value = 0
lblNodeName.Caption = ""
' onemogućiti gumbe, omogućuje se pokretanje tek na klik na neki node u Treeview2
cmdDodajDat.Enabled = False
cmdSpremiDat.Enabled = False
cmdUkloniDat.Enabled = False

```

```

cmdUkloniSve.Enabled = False
cmdSpremiZnacajke.Enabled = False
cmdSpremiPodatke.Enabled = False
'počisti sve prikazne kontrole
IstDatoteke1.Clear
IstDatoteke2.Clear
IstSheets.Clear
txtZnacajke.text = ""
lblNodeZnacajke.Caption = ""
lblNodeDatoteke.Caption = ""
lblNodeName.Caption = ""
'inicijalizacija TreeViewa za prikaz podataka (TreeView1)
'i za manipulaciju s podacima (TreeView2)
TreeView1.Nodes.Clear
TreeView2.Nodes.Clear
'prikaz matrice na radnom liste tek kad se cijeli list ispuni s podacima i formatira
Application.ScreenUpdating = False
i = 0
Call PuniArrWorkSheets
For Each wks In ThisWorkbook.Worksheets
    ' iz kolekcije radnih listova izdvajamo samo one koje prema nazivu pripadaju strukturi za vizualni prikaz putem Treeview kontrole
    ' strcomp je naredba za usporedbu stringova
    If StrComp(wks.Name, sLevel + CStr(i), vbTextCompare) = 0 Then
        'RAZINE spremamo u polje
        arrWorksheet(i) = wks.Name
        i = i + 1 ' pomoćna varijabla za numeraciju radnih listova - podazumijevani dio naziva
    End If
Next
ThisWorkbook.Worksheets(arrWorksheet(0)).Activate
'u for petlji obrađuje jedan po jedan radni list tj. razinu za Treeview
For i = LBound(arrWorksheet) To UBound(arrWorksheet)
    'uzimaju se samo popunjena mjesta u polju , još treba prilagoditi da gornja granica se
    'očita iz veličine arraya, funkcija Lbound donja granica, Ubound gornja granica
    If arrWorksheet(i) <> "" Then
        If i = 0 Then
            'Root
            Set wks0 = ThisWorkbook.Worksheets(arrWorksheet(i))
            Set wks1 = ThisWorkbook.Worksheets(arrWorksheet(i))
            Set rg = wks0.Range(Cells(1, 1), wks0.Cells(1, 1))
            'broj zadnjeg popunjenog retka za zadani rradni list i raspon
            lastrow = GetLastUsedRow(rg, wks0)
            For k = 1 To lastrow
                arrKeyRoot(k) = wks1.Name & "_" & wks0.Cells(k, 1)
                ' punimo u treeview2 samo ako je ćelija popunjena
                If wks0.Cells(k, 1) <> "" Then
                    TreeView1.Nodes.Add key:=arrKeyRoot(k), text:=wks0.Cells(k, 1).Value
                    TreeView2.Nodes.Add key:=arrKeyRoot(k), text:=wks0.Cells(k, 1).Value
                    TreeView2.Nodes(arrKeyRoot(k)).Tag = wks0.Name
                End If
            Next
        Next
    End If
Next

```

```

Else
'Child
Set wks0 = ThisWorkbook.Worksheets(arrWorksheet(i - 1))
Set wks1 = ThisWorkbook.Worksheets(arrWorksheet(i))
NumberOfColumns = wks1.Cells(1, wks1.Columns.Count).End(xlToLeft).Column
counter = 1
'za svaku kolonu mora pozvati childove
For counter = 1 To NumberOfColumns
If wks1.Cells(2, counter).Value <> "" Then
Set rg = wks1.Cells(1, counter)
arrKeyParent(i) = wks0.Name & "_" & rg.Value
' poziv procedure u kojoj se dodaju childovi
'wks1 je trenutni radni list, counter broji kolone
Call FillChildNodes(wks1, counter, arrKeyParent(i))
End If
Next counter
End If
End If
Next i
'nakon punjenja TreeViewa s podacima postavlja se svojstvo
'da se prikažu svi podaci
For Each nNode In TreeView1.Nodes
nNode.Expanded = True
Next
For Each nNode In TreeView2.Nodes
nNode.Expanded = True
Next
Application.ScreenUpdating = True
If TreeView1.Nodes.Count = 0 Then
' postavi se na 2. tab - unos podataka u radne listove tj. dodavanje / brisanje podataka
UserForm1.MultiPage1.Value = 1
Else
' postavi se na 1. tab tj. prikaz podataka u TreeView1
UserForm1.MultiPage1.Value = 0
End If
'ListBox se puni nazivima radnih listova koji predstavljaju
'bazu podataka za prikaz u TreeView
lstSheets.Clear
For i = LBound(arrWorksheet) To UBound(arrWorksheet)
lstSheets.AddItem (arrWorksheet(i))
Next
' poništavanje objektnih varijabli koje više ne koristim
Set wks0 = Nothing
Set wks1 = Nothing
Set wks = Nothing
'aktiviramo radni list na kojem je gumb za
'pokretanje aplikacije
ThisWorkbook.Worksheets("Vizualizacija").Activate
Range("A1:A1").Select
End Sub

```

```
Public Sub FillChildNodes(ByVal w As Worksheet, ByVal col As Integer, ByVal KeyParent)
```

```
'puni Child čvorove u strukturu
```

```
' treba utvrditi do kojeg retka treba uzimati vrijednosti ćelija u pojedinom stupcu
```

```
Dim lastrow As Long
```

```
Dim dataCell As Range
```

```
Dim rg0 As Range
```

```
Dim keyChild As String
```

```
Dim nRow As Long
```

```
'broji za svaki stupac koliko ima redaka
```

```
Set rg0 = w.Range(w.Cells(1, col), w.Cells(1, col))
```

```
' raspon je tu kolona
```

```
lastrow = GetLastUsedRow(rg0, w)
```

```
For Each dataCell In w.Range(w.Cells(2, col), w.Cells(lastrow, col))
```

```
'uzeti ponavljanje unutar istog lista u obzir, probati sa ključem generiranim kao random
```

```
keyChild = w.Name & "_" & dataCell.Value
```

```
If dataCell.Value <> "" Then
```

```
TreeView1.Nodes.Add KeyParent, tvwChild, keyChild, dataCell.Value
```

```
TreeView2.Nodes.Add KeyParent, tvwChild, keyChild, dataCell.Value
```

```
TreeView2.Nodes(keyChild).Tag = w.Name
```

```
End If
```

```
Next dataCell
```

```
End Sub
```

```
Function GetLastUsedRow(rg As Range, w As Worksheet) As Long
```

```
' vraća broj zadnjeg popunjenog retka za zadani raspon
```

```
Dim IMaxRows As Long
```

```
IMaxRows = w.Rows.Count
```

```
If IsEmpty(rg.Parent.Cells(IMaxRows, rg.Column)) Then
```

```
GetLastUsedRow = _
```

```
rg.Parent.Cells(IMaxRows, rg.Column).End(xlUp).Row
```

```
Else
```

```
GetLastUsedRow = rg.Parent.Cells(IMaxRows, rg.Column).Row
```

```
End If
```

```
End Function
```

```
Public Sub AddSheetAfterLast()
```

```
'dodaje radni list iza zadnjeg lista koji
```

```
'predstavlja bazu podataka
```

```
Dim shtNew As Worksheet
```

```
Dim shtLast As Worksheet
```

```
Dim ws As Worksheet
```

```
Dim i As Long
```

```
i = 0
```

```
Set shtLast = ThisWorkbook.Worksheets("Relacije2")
```

```
With ThisWorkbook
```

```

'Set shtLast = .Worksheets(.Worksheets.Count)
If WorksheetExists(ThisWorkbook, sLevel + "0") Then
    For Each ws In ThisWorkbook.Worksheets
        'strcomp je naredba za usporedbu stringova
        If StrComp(ws.Name, sLevel + CStr(i), vbTextCompare) = 0 Then
            i = i + 1
        End If
    Next
    ' Dodajemo radni list nakon zadnjeg koji se zove RAZINA - parametrizirano s sLevel
    Set shtLast = Worksheets(sLevel + CStr(i - 1))
    Set shtNew = Worksheets.Add(After:=shtLast)
    shtNew.Name = sLevel & CStr(i)
    arrDataWorkSheets(i) = sLevel & CStr(i)
Else
    Set shtNew = Worksheets.Add(After:=shtLast)
    shtNew.Name = sLevel & "0"
    arrDataWorkSheets(0) = sLevel & "0"
End If
End With
End Sub

```

```

Function WorksheetExists(wb As Workbook, sName As String) As Boolean

```

```

' provjera da li određeni radni list postoji u radnoj knjizi,
' parametri su radna knjiga i naziv radnog lista
    Dim s As String
    Dim ws As Worksheet
    On Error GoTo ErrHandler
    ' ako pokušamo otvoriti radni list koji ne postoji, javlja se greška
    ' u tom slučaju hvatamo pojavu greške i funkcija vraća False tj. da radni list ne postoji
    ' u toj radnoj knjizi
    s = wb.Worksheets(sName).Name
    WorksheetExists = True
Exit Function
ErrHandler:
    WorksheetExists = False
End Function

```

```

Function DeleteSheet(ws As Worksheet, bQuiet As Boolean) As Boolean

```

```

'google
' brisanje radnog lista koji je prosljeđen kao parametar
' drugi parametar uvjetuje da li se javlja VB poruka koja upozorava na brisanje
    Dim bDeleted As Boolean
    On Error GoTo ErrHandler
    bDeleted = False
    If CountVisibleSheets(ws.Parent) > 1 Then
        If bQuiet Then Application.DisplayAlerts = False
        bDeleted = ws.Parent.Worksheets(ws.Name).Delete
    Else

```

End If

ExitPoint:

Application.DisplayAlerts = True

DeleteSheet = bDeleted

Exit Function

ErrorHandler:

bDeleted = False

Resume ExitPoint

End Function

Private Function NavigateFromWorkBookPath() As String

'putanja koja se koristi prilikom zapisa datoteka

'mapa s podacima je podmapa mape u kojoj se

'nalazi aplikacija, koriste se za hyperlinkove

Dim WorkbookFolderPath As String

Dim FileFolderPath As String

Dim SlashPos As Integer

WorkbookFolderPath = ThisWorkbook.path

'VBA ugrađena funkcija InStrRev - nađe prvu poziciju pojavljivanja znaka,

'počevši od zadnjeg znaka stringa - vraća broj

'uzme poziciju znaka (broj) \ , posljednjeg u WorkbookFolderPath stringu

'u kojem je cijela putanja

SlashPos = InStrRev(WorkbookFolderPath, "\")

'VBA ugrađena funkcija Left uzima određeni broj znakova iz stringa

'počevši od 1. znaka u stringu - vraća string

FileFolderPath = Left(WorkbookFolderPath, SlashPos)

NavigateFromWorkBookPath = FileFolderPath

End Function

Public Sub PuniArrWorkSheets()

'nazive radnih listova koji su razine u hijerarhijskoj strukturi podataka

'spremamo u polje koje je deklarirano kao Private u deklaraciji na razini Userfom1

Dim i As Long

Dim ws As Worksheet

ReDim arrDataWorkSheets(0 To iLevels)

i = 0

For Each ws In ThisWorkbook.Worksheets

'strcomp je ugrađena VBA funkcija za usporedbu stringova

If StrComp(ws.Name, sLevel + CStr(i), vbTextCompare) = 0 Then

arrDataWorkSheets(i) = ws.Name

i = i + 1

End If

Next

End Sub

Private Sub cmdSpremiPodatke\_Click()

'nakon dodavanja i uklanjanja čvorova TreeViewa za manipulaciju

'podaci iz hijerarhijske strukture se zapisuju u radne listove

```
Dim TNode As Node
Dim TnodePrev As Node
Dim TnodeNext As Node
Dim i As Long
Dim j As Long
Dim k As Long
Dim lastrow As Long
Dim LastCol As Long
Dim MainCell As Range
Dim MainCellNext As Range
Dim wks As Worksheet
Dim wksNext As Worksheet
Dim bDodaj As Boolean
```

```
On Error GoTo ErrHandler
```

```
Application.ScreenUpdating = False
```

```
i = 0
```

```
j = 0
```

```
k = 0
```

```
bDodaj = True
```

```
' sve radne listove baze podataka najprije obrišemo
```

```
' zatim punimo prema izmijenjenoj Treeview strukturi
```

```
' polje arrDataWorkSheets saadrži nazive radnih listova s podacima
```

```
Do While arrDataWorkSheets(i) <> ""
```

```
    ThisWorkbook.Worksheets(arrDataWorkSheets(i)).UsedRange.Columns.Delete
```

```
    ThisWorkbook.Worksheets(arrDataWorkSheets(i)).UsedRange.Rows.Delete
```

```
    i = i + 1
```

```
Loop
```

```
' ROOT
```

```
Set MainCell = ThisWorkbook.Worksheets(arrDataWorkSheets(0)).Range("A1:A1")
```

```
Set wks = ThisWorkbook.Worksheets(arrDataWorkSheets(0))
```

```
wks.Activate
```

```
' 1. prolaz - zapisati ROOT podatke
```

```
i = 0
```

```
For Each TNode In TreeView2.Nodes
```

```
    Set wks = ThisWorkbook.Worksheets(arrDataWorkSheets(0))
```

```
    wks.Activate
```

```
    Set MainCell = ActiveSheet.Range("A1:A1")
```

```
    If TNode.Parent Is Nothing Then
```

```
        'Root - punimo podatke u prvi stupac
```

```
        MainCell.Offset(j, 0).Value = TNode.text
```

```
        j = j + 1 ' broj Root podataka
```

```
    End If
```

```
Next
```

```
' 2. prolaz - u radne listove zapisati Parent podatke
```

```
i = 0
```

```
For Each TNode In TreeView2.Nodes
```

```
    'sljedeća razina u odnosu na onu na kojoj se nalazi trenutni node
```

```
    i = Int(Mid(TNode.Tag, InStr(TNode.Tag, "_") + 1)) + 1
```



```

Set wksNext = ThisWorkbook.Worksheets(arrDataWorkSheets(i))
wksNext.Activate
Set MainCellNext = wksNext.Range("A1:A1")
'parenti za sljedeću razinu
If TNode.Children > 0 Then
    bDodaj = True
    k = 0
    ' tražimo 1. slobodni stupac
    Do Until MainCellNext.Offset(0, k) = ""
        'provjeravamo da li već postoji taj podatak
        If MainCellNext.Offset(0, k) = TNode.text Then
            bDodaj = False
        End If
        'child podatke punimo u k-ti stupac
        k = k + 1
    Loop
    If bDodaj Then
        'u 1.slobodni stupac u 1.retku zapisujemo parent za sljedeću razinu
        wksNext.Range("A1:A1").Offset(0, k).Value = TNode.text
    End If
End If
Next
' 3.prolaz - zapiši Child podatke u radne listove
Set MainCell = ThisWorkbook.Worksheets(arrDataWorkSheets(0)).Range("A1:A1")
Set wks = ThisWorkbook.Worksheets(arrDataWorkSheets(0))
wks.Activate
i = 0
For Each TNode In TreeView2.Nodes
    Set wks = ThisWorkbook.Worksheets(TNode.Tag)
    wks.Activate
    Set MainCell = ActiveSheet.Range("A1:A1")
    If TNode.Tag <> sLevel & "0" Then
        k = 0
        ' tražimo stupac koji je parent za trenutni node
        Do Until MainCell.Offset(0, k) = TNode.Parent.text Or MainCell.Offset(0, k) = ""
            'child podatke punimo u k-ti stupac
            k = k + 1
        Loop
        With ActiveSheet
            lastrow = GetLastUsedRow(MainCell.Offset(0, k), wks)
        End With
        If TNode.Parent.text = MainCell.Offset(0, k).Value Then
            MainCell.Offset(lastrow, k).Value = TNode.text
        End If
    End If
Next
Application.ScreenUpdating = True
Exit Sub
ErrorHandler:
    MsgBox Err.Description & vbNewLine & _

```

"Procedure cmdSpremiPodatke.", vbCritical, "Greška"

End Sub

---

' Sheet22 - macro vezan za radni list matrica

Private Sub Worksheet\_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)

'POSTAVLJANJE RELACIJA

Dim MainCell As Range

Dim relationRange As Range

Dim nrR As Integer

Dim nrC As Integer

'Definiramo glavnu ćeliju

Set MainCell = ThisWorkbook.Worksheets("Matrica").Range("A1:A1")

'Broj redaka

nrR = 1

Do Until MainCell.Offset(nrR, 0).Value = ""

nrR = nrR + 1

Loop

'Broj stupaca

nrC = 1

Do Until MainCell.Offset(0, nrC).Value = ""

nrC = nrC + 1

Loop

'Dio s relacijama

If nrC > 1 And nrR > 1 Then

Set relationRange = Range(MainCell.Offset(0, 0), MainCell.Offset(nrR - 1, nrC - 1))

'Provjera ćelije (je li odabrana ćelija u matrici?)

Set myIntersect = Application.Intersect(relationRange, ActiveCell)

If Not myIntersect Is Nothing Then

'popunjavanje veza

If ActiveCell.Value = "" Then

'Pokaži menu

Call ShowMenu

Else

'Obriši vezu

ActiveCell.Value = ""

End If

'poravnanje veza u svakom slučaju

With ActiveCell

.Font.Size = 18

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

```

    End With
End If
End If
'Ipak ne želimo dupli klik
Cancel = True
End Sub
Sub ShowMenu()
'Padajući izbornik (google)
    Dim relationRange As Range
    Dim i As Integer
'Definicija izbornika
    Dim myBar As CommandBar
'Definicija gumba u izborniku
    Dim myItem As CommandBarControl
On Error Resume Next
'Brisanje postojećeg izbornika
CommandBars("MyShortcut").Delete
On Error GoTo ErrorHandler
'Kreiranje izbornika
Set myBar = CommandBars.Add _
    (Name:="MyShortcut", Position:=msoBarPopup, Temporary:=True)
'Kreiranje svakog reda u izborniku za svaku ćeliju u radnom listu Relacije
'počinje se od ćelije A2 u radnom listu Relacije
Set relationRange = ThisWorkbook.Worksheets("Relacije").Range("A2:A2")
'tako dugo dok ima relacija
i = 0
Do While Not relationRange.Offset(i, 0).Value = ""
    'kreiraj gumb
    Set myItem = myBar.Controls.Add(Type:=msoControlButton)
    With myItem
        'Naslov
        .Caption = relationRange.Offset(i, 0).Value & " " & relationRange.Offset(i, 1).Value
        'Što se dešava kad se klikne (poziva se funkcija "SetRelation" i šalje parametar koji je u prvoj ćeliji)
        .OnAction = "Sheet22.SetRelation """" & relationRange.Offset(i, 0).Value & """"""
    End With
    i = i + 1
Loop
'Pokaži izbornik
CommandBars("MyShortcut").ShowPopup
Exit Sub
ErrorHandler:
    MsgBox Err.Description & vbNewLine & _
        "Procedure CreateShortcut.", vbCritical, "Error"
End Sub

Sub SetRelation(text As String)
'Postavljanje veze
'U aktivnu ćeliju se postavlja tekst (value) koji je zadan za relaciju
ActiveCell.Value = text

```

End Sub

' Sheet30 - macro vezan za radni list Matrica2

Private Sub Worksheet\_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)

'POSTAVLJANJE RELACIJA

Dim MainCell As Range

Dim relationRange As Range

Dim nrR As Integer

Dim nrC As Integer

'Definiramo glavnu ćeliju

Set MainCell = ThisWorkbook.Worksheets("Matrica2").Range("A1:A1")

'Broj redaka

nrR = 1

Do Until MainCell.Offset(nrR, 0).Value = ""

nrR = nrR + 1

Loop

'Broj stupaca

nrC = 1

Do Until MainCell.Offset(0, nrC).Value = ""

nrC = nrC + 1

Loop

'Dio s relacijama

If nrC > 1 And nrR > 1 Then

Set relationRange = Range(MainCell.Offset(0, 0), MainCell.Offset(nrR - 1, nrC - 1))

'Provjera ćelije (je li odabrana ćelija u matrici?)

Set myIntersect = Application.Intersect(relationRange, ActiveCell)

If Not myIntersect Is Nothing Then

'popunjavanje veza

If ActiveCell.Value = "" Then

'Pokaži menu

Call ShowMenu

Else

'Obriši vezu

ActiveCell.Value = ""

End If

'poravnanje veza u svakom slučaju

With ActiveCell

.Font.Size = 18

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

.WrapText = False

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

End If

```

End If
'Ipak ne želimo dupli klik
Cancel = True
End Sub

```

```

Sub ShowMenu()

```

```

'Padajući izbornik (google)

```

```

Dim relationRange As Range

```

```

Dim i As Integer

```

```

'Definicija izbornika

```

```

Dim myBar As CommandBar

```

```

'Definicija gumba u izborniku

```

```

Dim myItem As CommandBarControl

```

```

On Error Resume Next

```

```

'Brisanje postojećeg izbornika

```

```

CommandBars("MyShortcut").Delete

```

```

On Error GoTo ErrorHandler

```

```

'Kreiranje izbornika

```

```

Set myBar = CommandBars.Add _

```

```

(Name:="MyShortcut", Position:=msoBarPopup, Temporary:=True)

```

```

'Kreiranje svakog reda u izborniku za svaku ćeliju u radnom listu Relacije

```

```

'počinje se od ćelije A2 u radnom listu Relacije

```

```

Set relationRange = ThisWorkbook.Worksheets("Relacije2").Range("A2:A2")

```

```

'tako dugo dok ima relacija

```

```

i = 0

```

```

Do While Not relationRange.Offset(i, 0).Value = ""

```

```

'kreiraj gumb

```

```

Set myItem = myBar.Controls.Add(Type:=msoControlButton)

```

```

With myItem

```

```

'Naslov

```

```

.Caption = relationRange.Offset(i, 0).Value & " " & relationRange.Offset(i, 1).Value

```

```

'Što se dešava kad se klikne (poziva se funkcija "SetRelation" i šalje parametar koji je u prvoj ćeliji)

```

```

.OnAction = "Sheet30.SetRelation """" & relationRange.Offset(i, 0).Value & """"""

```

```

End With

```

```

i = i + 1

```

```

Loop

```

```

'Pokaži izbornik

```

```

CommandBars("MyShortcut").ShowPopup

```

```

Exit Sub

```

```

ErrorHandler:

```

```

MsgBox Err.Description & vbNewLine & _

```

```

"Procedure CreateShortcut.", vbCritical, "Error"

```

```

End Sub

```

```

Sub SetRelation(text As String)

```

```

'Postavljanje veze

```

```

'U aktivnu ćeliju se postavlja tekst (value) koji je zadan za relaciju

```

```

ActiveCell.Value = text

```

End Sub