

Automatizacija rada u Excelu makro naredbama i korisničkim formama

Kuzmić, Damir

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:980399>

Rights / Prava: [Attribution-NonCommercial-NoDerivs 3.0 Unported / Imenovanje-Nekomercijalno-Bez prerada 3.0](#)

Download date / Datum preuzimanja: **2024-10-30**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Damir Kuzmić

**Automatizacija rada u Excelu makronaredbama i korisničkim
formama**

ZAVRŠNI RAD

Varaždin, 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Damir Kuzmić

Matični broj: 0066092928

Studij: Primjena informacijske tehnologije u poslovanju

**Automatizacija rada u Excelu makronaredbama i korisničkim
formama**
ZAVRŠNI RAD

Mentorica:

Dr. sc. Petra Žugec

Varaždin, 2023.

Sadržaj

Uvod	1
Metodologija rada	2
Automatizacija rada.....	2
Automatizacija rada pomoću dodataka	3
Preduvjeti za automatizaciju rada	5
Visual Basic for Applications.....	5
VBE	6
Objekti, svojstva i metode.....	7
Vrste podataka i varijable u VBA	9
Manipulacija objektima i kontrola izvršenja	11
Makronaredbe.....	11
Snimanje makronaredbi	12
Makronaredbe kao podprocedure	15
Primjeri podprocedura	16
Makronaredbe kao funkcije	20
Primjeri funkcija.....	21
Korisničke forme.....	24
Kreiranje i kontrole korisničkih formi	25
Primjeri korisničkih formi	26
Zaključak.....	31

Damir Kuzmić

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Ovaj rad se bavi automatizacijom rada u Microsoft Excel programskom alatu. Opisani su preduvjeti za automatizaciju rada te programski jezik Visual Basic for Applications (VBA). Unutar VBA obrađeni su objekti, svojstva i metode, vrste podataka te varijable kao i manipulacija objektima i kontrolama izvršenja ovog programskog jezika. U glavnom dijelu rada detaljnije su obrađene makronaredbe i korisničke forme. Opisani su načini snimanja i vrste makronaredbi te kreiranje korisničkih formi, a kroz jednostavnije primjere pokazane su mogućnosti i svrhe njihovog korištenja i primjene.

Ključne riječi: automatizacija rada, Microsoft Excel, Visual Basic for Applications, makronaredbe, korisničke forme, programiranje.

Uvod

Excel je već dugi niz godina jedan od najvažnijih programskih rješenja poslovne domene sa izuzetno širokom primjenom koji suvereno prati velik dio poslovanja kako malih i jednostavnih poslovnih entiteta sve do velikih i značajnih korporacijskih operacija. Kao integralni dio Office paketa Excel je jedan od njegovih najvažnijih dijelova te unatoč činjenici da je prva inačica izašla krajem osamdesetih godina prošlog stoljeća i dalje je jedan od najkorištenijih programskih rješenja za poslovne procese koji su vezani uz velike količine podataka, različite matematičke izračune kao i jednostavnije baze podataka. Poznavanje rada u Excelu nudi široku primjenu te mogućnosti obavljanje zadataka iz različitih domena poslovanja.

Obzirom na navedeno, kao bitna značajka Excela je njegova automatizacija radnih zadataka te će ovom radu biti objašnjeni osnovni postulati automatizacije, jednostavni konstrukti unutar VBA programskog jezika te kroz primjere makronaredbi i korisničkih formi pokazane neke mogućnosti i prilagodbe koje uvelike olakšavaju i ubrzavaju rad te generalno procese koji oduzimaju velik dio vremena čime je učinjena ušteda mnogostruka.

Metodologija rada

Budući da je ovaj rad temeljen na naprednijem korištenju aplikacije Excel, podrazumijevano je da su savladani osnovni postulati korištenja same aplikacije, nisu dodatno pojašnjavani osnovni principi rada u Excelu, već kako je u uvodu spomenuto, sav naglasak stavljen na razradu automatizacije rada, osnovnih postulata i pojmova te glavnih nositelja automatizacije u vidu makronaredbi i korisničkih formi. Obradeni su jednostavniji primjeri snimanja, izrade i korištenja makronaredbi te korisničkih formi.

U radu je korišten Microsoft Excel LTSC Professional Plus 2021 sa zadanim hrvatskim jezičnim paketom te Windows 10 Pro inačicom operativnog sustava.

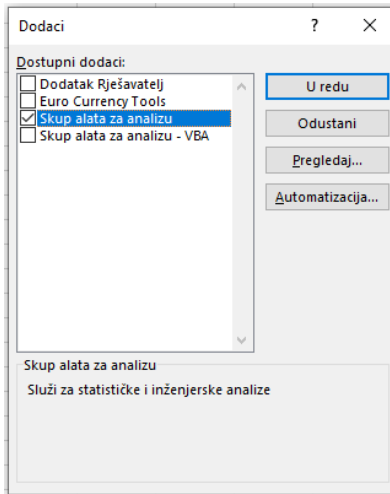
Automatizacija rada

Generalno gledano postoji više različitih načina za automatizaciju rada u Excelu. Svaki od njih ima svrhu i primjenu u različitim područjima te ne postoji jedinstveno rješenje za bilo koji problem odnosno poslovni zadatak koji je potrebno odraditi pomoću Excela. Bitno je istaknuti kako se korisničko iskustvo u primjeni Excela prema vlastitim znanjima i potrebama, ali ipak postoje određeni postulati koje je dobro slijediti kako bi maksimizirali produktivnost i smanjili utrošak vremena, a napose uz sve benefite koje današnje inačice Excela nude. Tržište danas nudi velik broj gotovih programskih rješenja u vidu besplatnih ali i komercijalnih tzv. dodataka odnosno eng. add-ins. Svaki takav dodatak nudi set gotovih mogućnosti te obuhvaća specifične radne zadatke koji se automatiziraju na različite načine. Navedeno se osobito odnosi na različite analitičke procese, vizualne prezentacije, prilagodbe, vrste pregleda, pivot tablice i još mnogo različitih specifičnih mogućnosti koje je moguće odraditi samostalno pomoću makronaredbi i korisničkih formi o kojima će više riječi biti u narednim poglavljima, a nadalje kako je naveo Paul Lomax nabrajajući mogućnosti VBA programskog jezika, „...automatizirati bilo što, a što je moguće učiniti uz pomoć tipkovnice, miša ili izbornika“.¹

¹ Vlastiti prijevod: Lomax, P., VB & VBA in a Nutshell: The Language, O'Reilly & Associates, Sebastopol CA, 1998., str. 7.

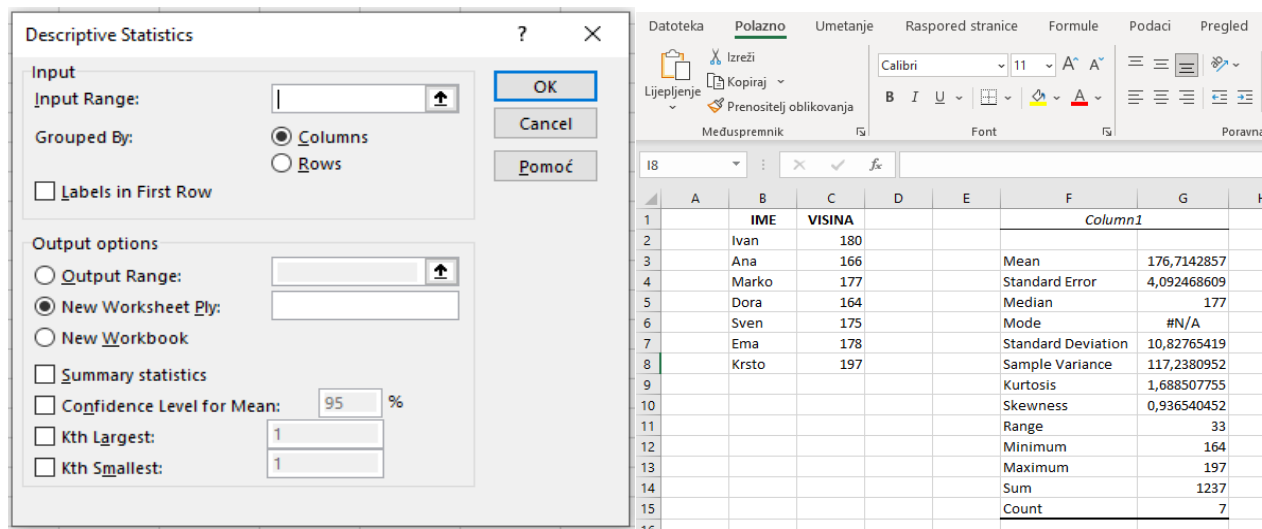
Automatizacija rada pomoću dodataka

Najjednostavniji način za automatizaciju određenog radnog procesa pomoću Excela je pomoću spomenutih dodataka. Određeni dodaci dolaze uz samu verziju Excela te ih je moguće pronaći u kartici programiranje pod odjeljkom dodaci. Pod opcijom dodaci za Excel možemo vidjeti koji dodaci su trenutno dodani, ali samo oni označeni su i instalirani.



Slika 1 - popis dostupnih dodataka

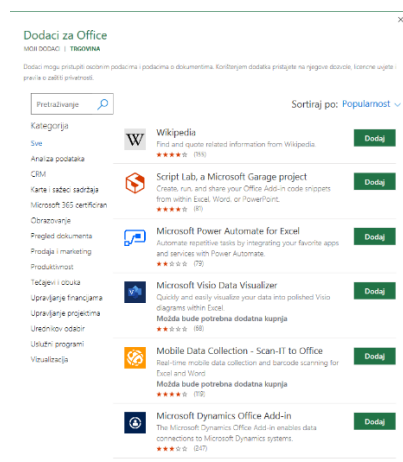
Instalacijom „Skup podataka za analizu“ u kartici Podaci dobivamo novi odjeljak u kojem je pod odjeljkom Analiza dodana nova ikona kojom možemo započeti detaljno analiziranje prema specifičnim zahtjevima. Za potrebe prikaza odabrana je najjednostavnija provjera nad nasumično unesenim podacima o visini osoba. Popunjavanjem zahtijevanih podataka o rasponu ćelija sa podacima, opcijama izvještaja te samoj vrsti analize dobivamo opće statističke podatke.



Slika 2 – prilaz podataka pomoću „Descriptive Statistic“ analize

Naravno, svrha ovakvog pregleda je pokazati kako i relativno jednostavniji alat može znatno ubrzati izračune te osigurati korisnika od potencijalnih grešaka u samim računanju, a uz precizno unesene podatke dati nove mogućnosti u automatizaciji te prezentaciji podataka, izradi izvještaja i ostalim oblicima poslovnih aktivnosti.

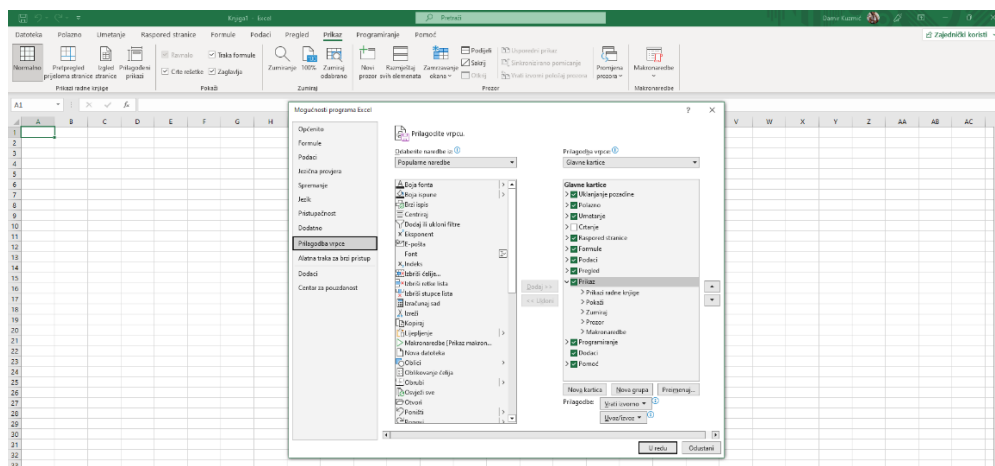
Novije inačice Excel-a u kartici programiranje također nude opciju za brz pristup velikom broju dodataka za Office paket među kojima se značajan dio odnosi na Excel što doprinosi imerzivnosti unutar cijelog Office paketa te sustavno pomaže u svekolikoj ukupnosti radnih procesa.



Slika 3 - Microsoft trgovina dodataka za Office paket

Preduvjeti za automatizaciju rada

Kako bi uopće mogli koristiti neku vrstu automatizacije potrebno je zadovoljiti nekoliko preduvjeta odnosno prilagoditi Excel za njihovo korištenje. Prije svega potrebno je u alatnoj traci aktivirati karticu programera odnosno eng. „Developer“ budući je ona inicijalno sakrivena. Kako bi aktivirali karticu programera potrebno je otvoriti datoteka te nadalje tzv. mogućnosti gdje se pod prilagodba vrpce nudi označavanje kvačicom programiranje čime se automatski ta kartica pojavljuje među ostalim glavnim karticama u vrpici. Ipak, moguće je napraviti, odnosno snimiti makronaredbu i bez aktivacije kartice programera, ali kako bi dobili punu funkcionalnost te dodatne opcije o kojima će više riječi bit u narednim poglavljima svakako je poželjno, a u slučaju ovog rada i nužna aktivacija iste.



Slika 4 - aktivacija kartice programera

Visual Basic for Applications

Visual Basic for Applications programski je jezik stvoren od strane Microsofta kako bi se omogućilo korištenje makronaredbi i korisničkih formi odnosno cjelokupna automatizacija procesa što je ujedno glavna tema ovoga rada. VBA tijekom svog razvoja prošao je put od jednostavnog interpreterskog jezika pa sve do vlastitog integriranog okruženja za razvoj i uklanjanje grešaka unutar aplikacija u kojima se koristi. Njegov razvojni put kreće sa implementacijom u Microsoft Excel 1993. godine, a svakoj sljedećom razvojnom iteracijom implementiran je u sve veći broj

proizvoda Office paketa da bi 1996. godine postao jezični odnosno sintaksni element Visual Basic programa i učinio značajan zaokret prema objektno-orientiranom programiranju. Inačicom 5.0 VBA pokriva cjelokupan Office 97 paket, a daljnje širenje potaknuto je Microsoftovim licenciranjem za druga softverska rješenja. Razvoj je nastavljen je kroz verziju 6.0 iz 1998. godine iako ne tako značajno kao u nekim ranijim inačicama. Napokon, u verzijama Office 2013, 2016, 2019 i 2021 paketa implementiran je VBA 7.1.

VBA je povezan i nerijetko se miješa sa VB odnosno Visual Basic što i ne čudi jer imaju istu sintaksu, ali znatno različitu primjenu. Kao glavne različitosti navodi se kako Visual Basic može kreirati samostalne izvršne programe dok se VBA koristi uglavnom unutar aplikacija Office paketa te povezano sa time VB aplikacije samostalno se kompajliraju dok je za izvršavanje koda VBA zahtjevan vanjski interpreter. U konačnici VB kreira cjelovite programe dok VBA djeluje kao poveznica između aplikacije unutar koje se koristi i korisnika kako bi pružio neku vrstu servisa i dodatnih mogućnosti za korištenu aplikaciju.

Više riječi o samom VBA kodu, načinu njegova korištenja te primjerima biti će riječi nadalje u ovome radu.

VBE

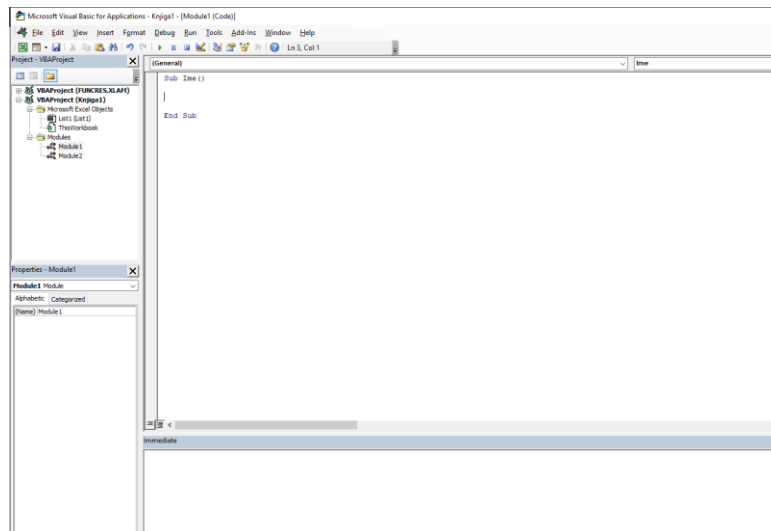
„Visual Basic Editor je zasebna aplikacija odnosno program koji omogućava pisanje VBA koda, procedura i modula.“² Ima zasebno sučelje odvojeno od Excel radne knjige, a najjednostavniji način za otvaranje istoga je pomoću tipkovnog prečaca ALT+F11. VBE ima nekoliko cjelina koje omogućavaju kreiranje makronaredbi te njihovu manipulaciju. Osim standardnih izbornika i alatne trake glavni dijelovi su preglednik projekta, prozor za kodiranje, prozor za svojstva i prozor za provjeru grešaka.

Ukoliko kreiramo makronaredbu snimačem u pregledniku projekta automatski ćemo u hijerarhijskoj strukturi vidjeti modul koji je kreiran samim snimanjem makronaredbe dok za funkcije sami kreiramo modul čime se kreiraju i ostali dijelovi VBE. Prema tipu koda moduli mogu sadržavati:

² Prema James, J., Excel 2021 for beginners, Independently published, 2021., str. 331.

- procedure
- funkcije
- deklaracije

U prozoru za kodiranje moguće je uređivati ili pisati sam kod ovisno o vrsti makronaredbe koju želimo kreirati ili izmijeniti dok prozor svojstava nudi prikaz svih svojstava objekta izabranog iz preglednika projekta te je moguće uređivanje i izmjena samih svojstava. Prozor za kontrolu grešaka služi za provjere odnosno kontrole grešaka.



Slika 5 - Prikaz VBE sučelja

Objekti, svojstva i metode

Zbog boljeg razumijevanja strukture Excela te mogućnosti koje pruža VBA pri kreiranju makronaredbi potrebno je detaljnije objasniti pojam objekata te njihova svojstva i metode korištenja.

Excel koristi objektni model u hijerarhijskoj strukturi:

- raspon
- radni list
- radna knjiga

➤ aplikacija³

Pošto se radi o hijerarhiji svaki sljedeći objekt sadržan je u prethodnom pa tako aplikacijski objekt kao najviše nadređen daje opće informacije o samoj aplikaciji, radna knjiga kao niže rangiran objekt sadrži radne listove i kao najniža hijerarhijska jedinica sadržana u svakom radnom listu čini raspon u kojem su sadržane pojedine ćelije, redovi, stupci sa mogućim grupiranjima u blokove odnosno selekcije istih.⁴ Općenito pojedine kategorije objekata prema svom tipu mogu biti grupirane u kolekcije koje tako usložene tvore objekte za sebe⁵

Generalna svrha hijerarhijske podjele ima daljnju podlogu u referenciranju odnosno kako bi točno određenom objektu pridijelili neku vrijednost ili ga opisali potrebno je navesti točnu putanju unutar aplikacije. Svaki objekt se referencira prema sintaksi u kojoj se navodi naziv radne knjige, radni list te raspon na koji se neka vrijednost odnosi.

Ipak, moguće je koristiti i skraćeni oblik reference budući Excel prepoznaje odnosno presumira da je otvorena aplikacija ona na koju se neka referenca odnosi, a isto se smatra i za otvorenu radnu knjigu te radni list.⁶ Skraćeni oblik se navodi kao;

```
Range("B4").Select
```

U prethodnoj izjavi vidljivo je kako je kao raspon odabrana ćelija B4 dok se automatski kao aktivna odabrana radna knjiga i radni list u kojem se nalazimo.

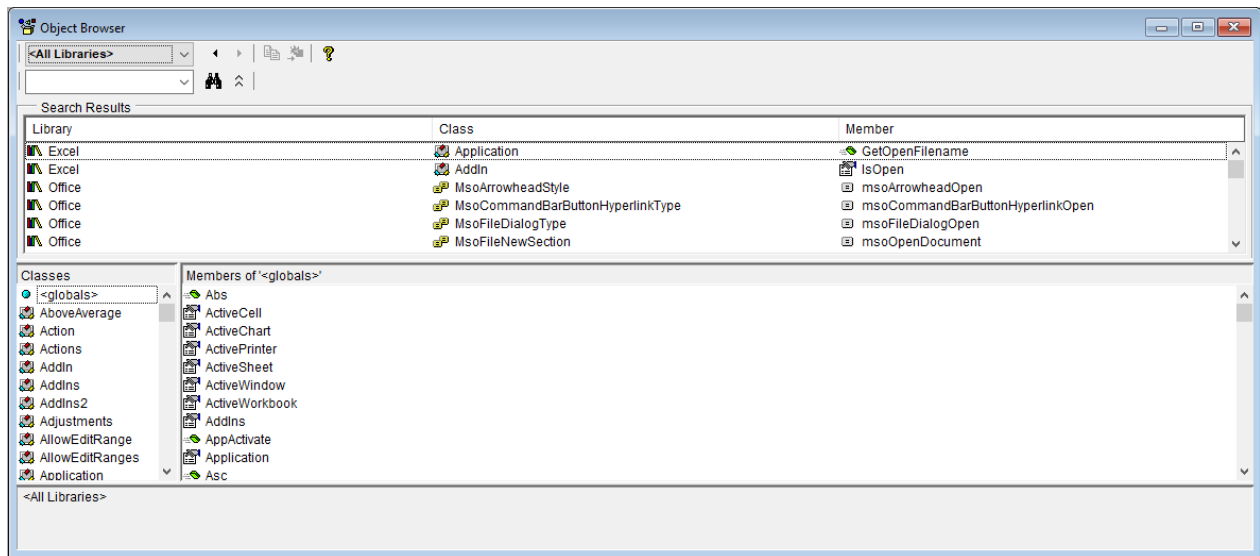
Popis svih objekata te njihovih svojstava možemo otvoriti u VBE sučelju pomoću F12 ili na alatnoj traci unutar odjeljka „View“.

³ Prema James, J., Excel 2021 for beginners, Independently published, 2021., str. 352

⁴ Prema James, J., Excel 2021 for beginners, Independently published, 2021., str. 353.

⁵ Prema Van Niekerk, M., VBA Automation for Excel 2019 Cookbook, Packt Publishing, Birmingham, 2020., str. 76.

⁶ Ibid. str. 76



Slika 6 - Object Browser u VBE sučelju

Objasnivši objekte dolazimo do svojstava koje objekti mogu poprimiti. Svojstva se mogu objasniti kao atributi kojima se opisuju objekti. Ta svojstva su različita za pojedine objekte, a mogu biti reprezentirani u vidu visine, širine, boje, fonta, vrijednosti koje pojedini objekt može poprimiti i slično. Za svaki objekt je moguća provjera njegovih svojstava te promjena istih.

Metode su još jedan konstrukt koji je potrebno pobliže objasniti te se navodi kako su to akcije koje je moguće izvesti objektima, a pomoću njih je moguća i promjena samih svojstva objekata. Metode omogućavaju interakciju Excela i ostalih programa računala kao što su printeri ili čitanje podataka iz nekih dokumenata ili modificiranje više svojstava objekata u isto vrijeme.⁷

Vrste podataka i varijable u VBA

Zbog lakšeg spremanja podataka u memoriju moguće je odrediti vrstu podataka sa kojima se manipulira pa iako VBA ne traži nužno deklariranje tipa podataka te je sposoban to samostalno odrađivati, zbog ekonomičnosti i brzine izvođenja korisno unaprijed definirati tipove podataka koji

⁷ Prema Alexander, M., Kusleika, D., previously by Walkenbach, J., Excel 2019 Bible, Wiley, Indianapolis IN, 2019., str 937.

će se koristiti. Ukoliko VBA samostalno određuje tip podataka može doći do usporavanja rada programa dok je prednost definiranja tipa mogućnost dodatne provjere grešaka koje je inače teško otkriti.⁸ Tipove podatka koje je moguće definirati dijelimo na numeričke i ne numeričke;

NUMERIČKI	NENUMERIČKI
bajt eng. byte	niz znakova (fiksna duljina) eng. string (fixed length)
cjelobrojni eng. integer	niz znakova (varijabilna duljina) eng. string (variable length)
dugi eng. long,	logički eng. boolean
jednostruki eng. single,	datum eng. date
dvostruki eng. double,	objekt eng. object
decimalni eng. decimal	varijantni (sa brojevima) eng. variant (with numbers)
valuta eng. currency	varijantni (sa znakovima) eng. variant (with characters) ⁹

U slučaju da tip podataka ipak nije definiran, automatski se postavlja kao varijantni tip, eng. variant što predstavlja univerzalni tip koji može pohranjivati bilo koji od gore navedenih tipova podataka već prema određenoj potrebi.¹⁰

Vezano za same podatke i njihovu pohranu potrebno je spomenuti i varijable. Varijable se definiraju kao imenovani prostor za pohranu podataka koje mogu poprimiti različite tipove podataka. Budući da VBA ne zahtjeva nužno deklariranje tipova podataka i ukoliko ih se ne definira u određeni tip podataka inicijalno su postavljene na promjenjivi tip i definiraju se samostalno prema potrebi. Ipak, kako je rečeno, zbog brzine izvođenja kao i zbog eventualne provjere grešaka ili njihovog lakšeg uočavanja poželjno je njihovo definiranje.

Varijable mogu biti vezane samo za jednu proceduru, modul ili pak za sve module neke radne knjige ovisno o načinu njihovog deklariranja. Lokalne varijable vezane su samo za proceduru u kojoj su deklarirane, a ukoliko želimo istu varijablu koristiti u više procedura nekog modula

⁸ Prema Alexander, M., Kusleika, R., Excel 2019 Power Programing, John Wiley & Sons, Indianapolis IN, 2019., str. 66.

⁹ Vidi James, J., Excel 2021 for beginners, Independently published, 2021., str. 380.

¹⁰ Prema Alexander, M., Kusleika, R., Excel 2019 Power Programing, John Wiley & Sons, Indianapolis IN, 2019., str. 67.

potrebno ju je deklarirati prije prve procedure. Ukoliko je varijabla potrebna za sve module nekog projekta potrebno je njezino deklariranje kao javno.¹¹

Uz varijable vežemo i konstante koje su slične po svrsi, ali vrijednosti koje su u njima pohranjene nije moguće mijenjati. Njihovo korištenje je osobito korisno zbog preglednosti i lakšeg razumijevanja koda kao i boljeg kontroliranja samih vrijednosti na jednom mjestu.¹²

Manipulacija objektima i kontrola izvršenja

Kako bi uspješnije koristili VBA postoje konstrukti koji olakšavaju kako pisanje tako i preglednost samog koda te olakšavaju manipulaciju objektima te njihovim kolekcijama. Prvi takav je „With-End With“ konstrukt pomoću kojega možemo znatno brže i jednostavnije obaviti neke operacije odnosno pridijeliti neka svojstva i metode na određene objekte u kodu koji pišemo. Sličan učinak ima i konstrukt „For each-Next“ jer olakšava rad s kolekcijama objekata osobito u slučajevima kada nije poznato koliko elemenata neka kolekcija ima, a potrebno je nad svima odraditi neku akciju.

Kontrola izvršenja koda ima veći broj konstrukata koji su u većini slični drugim programskim jezicima te se dijele na uvijete i petlje. Tako postoji „IF-Then“ i „Select case“ uvjet za donošenje odluka te petlje „For-Next“, „Do While“ i „Do Until“. Kod kontrole koda postoji i specifična naredba „Go-To“ koja prenosi izvođenje programa na novu instrukciju.¹³

Makronaredbe

Makronaredbe su dio koda izrađenog pomoću VBA programskog jezika koje se koriste za automatizaciju radnih procesa u Excelu.¹⁴ Često se postavlja pitanje kada je poželjno ili potrebno

¹¹ Prema Alexander, M., Kusleika, R., Excel 2019 Power Programing, John Wiley & Sons, Indianapolis IN, 2019., str. 70-72.

¹² Vidi James, J., Excel 2021 for beginners, Independently published, 2021., str. 383.

¹³ Vidi Alexander, M., Kusleika, R., Excel 2019 Power Programing, John Wiley & Sons, Indianapolis IN, 2019., str. 88.

¹⁴ Vidi Jordan, J., Excel 2021 for beginners, Independently published, 2021., str. 309.

koristiti makronaredbe, a kao odgovor na to pitanje navedene su situacije kada je svakako poželjno njihovo korištenje kao što su slučajevi kada su radni zadaci ponavljajući, česti, zahtjevni, strukturirani, vremenski ograničeni, u zadacima podložnim ljudskim greškama ili zadacima koji traju više od 15 do 20 minuta.¹⁵

Makronaredbe mogu biti snimljene ili pisane u VBE odnosno Visual Basic Editoru, a dijele se u dvije kategorije, podprocedure i funkcije. Kako je VBA izuzetno bogat programski jezik, zapravo ne postoje klasična ograničenja u mogućnostima programiranja, a prema autorima knjige Excel 2019 Bible, samo neki od mogućih područja kreiranja i primjene makronaredbi su;

- umetanje predefiniranog teksta
- automatizacija procedura koje se često koriste
- automatizacija ponavljajuće radnje
- kreiranje vlastitih naredbi
- kreiranje pojednostavljenih sučelja za korisnike koji nisu detaljnije upoznati sa Excelom
- razvijanje novih funkcija radnih listova
- kreiranje aplikacija potpuno pokretanih pomoću makronaredbi
- kreiranje vlastitih dodataka za Excel¹⁶

Naravno, sve navedeno samo djelomično obuhvaća brojnost mogućnosti koje korisnik može usustaviti, kreirati ili poboljšati, a sve u svrhu brzine, jednostavnosti, bolje produktivnosti ili uopće poboljšanja korištenja Excela kao programskog alata.

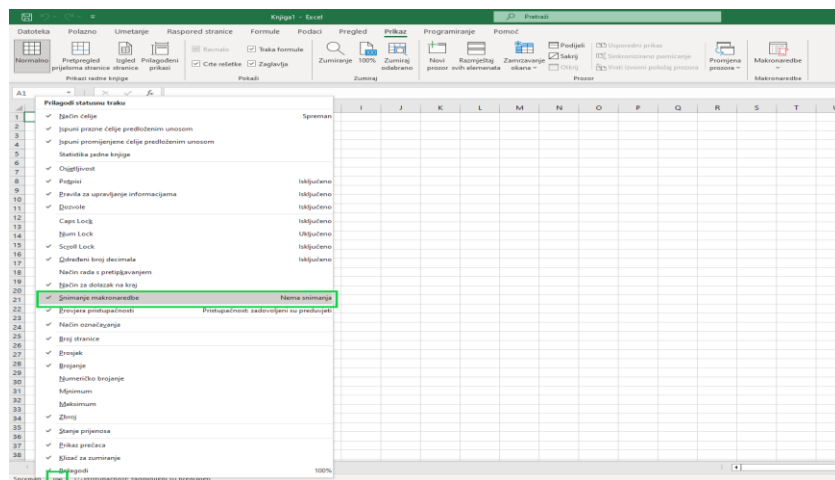
Snimanje makronaredbi

Proces snimanja makronaredbi svodi se na nekoliko sistemskih zadataka koji predstoje samom procesu bilježenja radnji koje želimo memorirati kako bi se automatizirao set nekih radnji koje je potrebno rekreirati u novim radnim zadacima bez obveze da ih ponavljamo ručno.

¹⁵ Prema Blackwood, N., Make the most of macros, In the Black, July 2011., str. 50.

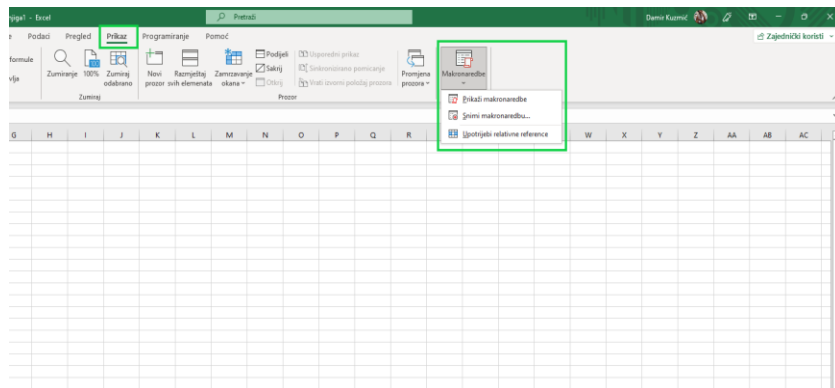
¹⁶ Vidi Alexander, M., Kusleika, D., previously by Walkenbach, J., Excel 2019 Bible, Wiley, Indianapolis IN, 2019., str 919.

Snimanje je moguće provesti na nekoliko različitih načina od kojih svaki dovodi do realizacije istog zadatka te je ostavljeno na volju korisniku koji je će upotrijebiti. Najbrži i najjednostavniji je korištenje statusne trake u podnožju Excel programa gdje u lijevom kutu postoji poruka „Spreman“ čijim otvaranjem desnom tipkom miša otkrivamo izbornik u kojem je moguće uključiti naredbu „Snimanje makronaredbe“ nakon kojeg se pojavljuje ikona čijim pritiskom započinjemo samo snimanje makronaredbe kako je prikazano na slici 7.



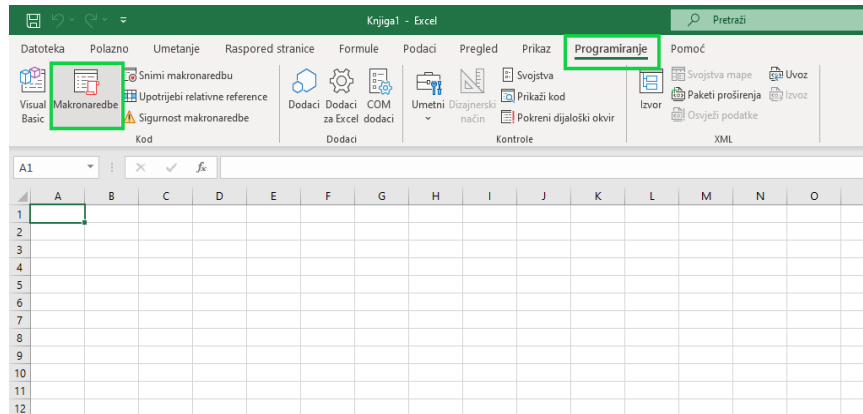
Slika 7 - snimanje makronaredbe pomoću statusne trake

Sljedeći način podrazumijeva otvaranje kartice „Pregled“ u čijoj sekciji „Makronaredbe“ otvaramo padajući izbornik unutar kojeg se nudi opcija „Snimi makronaredbu...“ kojom također pokrećemo snimanje makronaredbe.



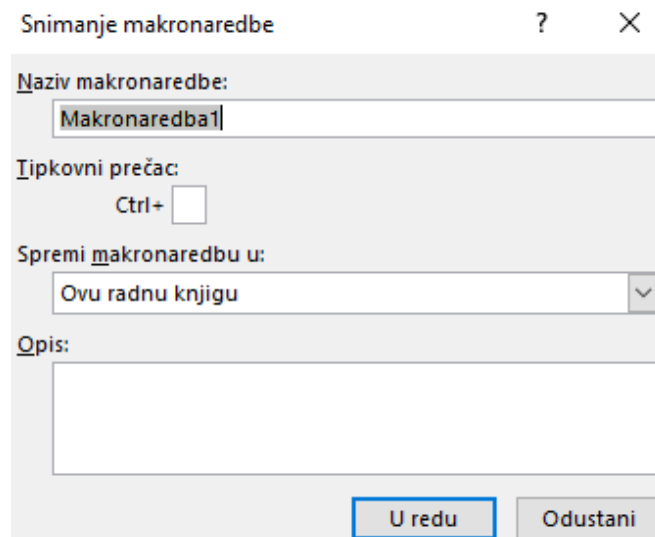
Slika 8 - snimanje makro naredbe pomoću kartice View

Napokon, kao najnapredniji način za koji prethodno moramo uključiti karticu programera, kako je već poblíže opisano u prethodnom poglavlju gdje se nalazi ikona koja omogućava pokretanje snimanja makronaredbe.



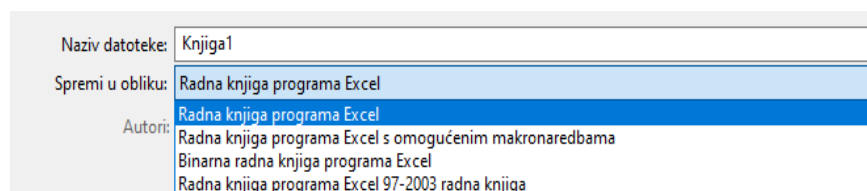
Slika 9 - snimanje makronaredbe pomoću kartice programera

Unatoč vlastitim preferencijama oko izbora za pokretanje snimanja makronaredbi, svima su zajednički daljnji postupci oko imenovanja, spremanja te manipulacije i daljnje obrade kreiranih makronaredbi.



Slika 10 - spremanje makronaredbe

Pokretanjem snimača makronaredbi otvara se dijaloški okvir u koji je potrebno upisati sam naziv makronaredbe koja se snima, vodeći računa o svrsishodnosti samog naziva iako nema zapreka u imenovanju. Nadalje, u dijaloškom okviru ponuđeno je pridjeljivanje tipkovnog prečaca pomoću kojega se brzo poziva spremljenu makronaredbu. Opisni odjeljak služi za detaljnije pojedinosti oko makronaredbe gdje je poželjno navesti neke specifičnosti, osobito ukoliko postoji više sličnih kako bi ih jednostavnije razlikovali prilikom ponovnog pozivanja. Kao mogućnost spremanja inicijalno je ponuđeno „ova radna knjiga“ koja je trenutno aktivna, „nova radna knjiga“ ili „osobna radna knjiga makronaredbi“. Za slučaj da želimo koristiti stvorenu makronaredbu u više projekata kao mjesto spremanja koristimo „osobna radna knjiga makronaredbi“ koje se kreira prilikom prvog spremanja. Bitno je naglasiti kako prilikom spremanja same radne knjige koja sadrži makronaredbu potrebno spremite u format koji to omogućava. Zadani format spremanja Excel radnih knjiga je .xlsx koji ne podržava makronaredbe već da bi spremili makronaredbu unutar Excela koristimo tip datoteke .xlsm dok se još kao osnovni načini spremanja nude starija ekstenzija .xls za Excel 97-2003 te novija varijanta .xlsb koja označava Excel binarnu radnu knjigu.¹⁷ U slučaju Excela 2021 tekstualno su opisane vrste spremanja te se također podudaraju se sa više opisanim.



Slika 11 - mogućnosti spremanja radne knjige Excel-a

Makronaredbe kao podprocedure

Budući su u prethodnim poglavljima detaljnije objašnjeni osnovni konstrukti VBA programskog jezika kao i snimanja makronaredbi te njihova osnovna manipulacija unutar Excela i VBE sučelja u ovom poglavlju više riječi će biti o samim proceduralnim makronaredbama.

¹⁷ Prema James, J., Excel 2021 for beginners, Independently published, 2021., str. 310.

Svaka podprocedura započinje sa „Sub“ i završava sa „End Sub“. U prikazanom kodu automatski su dodijeljeni komentari koji se navode sa jednostrukim znakom navodnika, ali nisu nužni i ukoliko ih se obriše nakon uređivanja samog koda u VBE editoru, neće imati nikakav utjecaj na njegovo izvršenje. Selektiranim ćelijama u rasponu je pomoću formule FormulaR1C1 koja se predefinirano koristi pri radu sa svojstvima objekata od strane snimača makronaredbi dodano svojstvo „Primjer 1“ te „Primjer 2“ a deklarira se i svojstvo fonta "bold" kao istinito.

	A	B	C	D	E
1					
2		Primjer 1			
3		Primjer 2			
4					
5					
6					

Slika 12 – prikaz snimljene makronaredbe Primjer_1

Često je potrebno u radni list unijeti unaprijed zadane nazive kolona te ih formatirati na određeni način da budu uočljiviji i spremni za daljnje obradu i unos podataka pa je u narednom primjeru pokazano kako je takav zadatak jednostavno odraditi pomoću kreirane makronaredbe;

Sub Formatiranje_stranice()

With Range("A1:G1")

.Font.Size = 12

.Font.Bold = True

.Font.Name = "Arial"

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

.Value = Array("Signatura", "Naslov", "Autor", "Godina izdanja", "Mjesto izdanja",

"Inventarni broj", "Vrijednost")

```

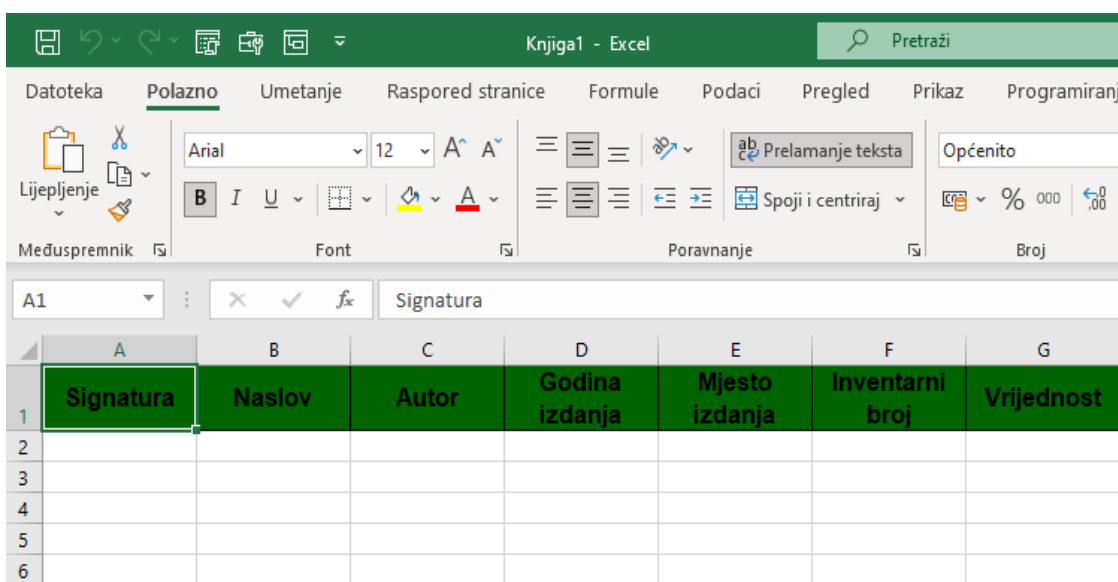
.WrapText = True
.Columns.ColumnWidth = 14
.Interior.Color = RGB(0, 100, 0)
.Borders.LineStyle = xlContinuous
.Borders.Weight = xlThin

```

End With

End Sub

Ovakvom makronaredbom označen je raspon ćelija u koje je automatski unesen željeni tekst sa specificiranim veličinom, podebljanem, vrstom fonta, a također je određena veličina ćelija, boja, prelamanje testa i slično. Pomoću konstrukta with - end with obrađene su sve željene osobine i pridijeljene na zadani raspon, u ovom slučaju između A1 i G1.



Slika 13 - prikaz makronaredbe Formatiranje_stranice

Za sljedeći primjer makronaredbe odabran je konkretan zahtjev nad gotovim html izvještajem kopiranim u excel datoteku. Radi se o izvještaju knjižnice Pravnog fakulteta u Zagrebu gdje je bilo potrebno pridijeliti unaprijed definirane vrijednosti pojedinim naslovima u izvještaju prema godini

njihovog izlaska.²⁰ Obzirom na veliku količinu podataka sa više od devet tisuća redova koje bi ručno trebalo provjeriti, izrada makro naredbe odličan je primjer kako se sav posao odrađen u svega nekoliko minuta.

```
Sub izracun_cijene()  
  Dim celija As Range  
  Dim vrijednost As Variant  
  For Each celija In Range("E1:E" & Cells(Rows.Count, "E").End(xlUp).Row)  
    vrijednost = celija.Value  
    If IsNumeric(vrijednost) Then  
      If vrijednost <= 1800 Then  
        celija.Offset(0, 3).Value = 333  
      ElseIf vrijednost <= 1900 Then  
        celija.Offset(0, 3).Value = 200  
      ElseIf vrijednost <= 1990 Then  
        celija.Offset(0, 3).Value = 10  
      Else  
        celija.Offset(0, 3).Value = 25  
      End If  
    Else  
      celija.Offset(0, 3).Value = "Provjeriti godinu"  
    End If  
  Next celija  
End Sub
```

Pomoću For each petlje prolazi se kroz sve ćelije u stupcu „Godina izdanja. Vrijednost svake ćelije se dodjeljuje varijabli vrijednost te se provjerava da li je ta vrijednost numerička. Pomoću uvjeta provjeravaju se numerički slučajevi vrijednosti u zadanim rasponima te na temelju njihovih vrijednosti u stupcu „Vrijednost“ upisuje odgovarajuća vrijednost. Ukoliko podatak u svakoj

²⁰ Javno dostupan katalog knjižnice pomoću kojeg je kreiran izvještaj za navedeni primjer, https://www.pravo.unizg.hr/knjiznica/online_katalog

pojedinoj ćeliji stupca „Godina izdanja“ nije numerički stupcu „Vrijednost“ upisuje se „Provjeriti godinu“, a u protivnom upisuje se odgovarajuća vrijednost zadana u uvjetu.

	A	B	C	D	E	F	G	H
	SIGNATURA	UNIN	NASLOV	AUTOR	GODINA IZDANJA	MJESTO IZDANJA	INV. BROJ	VRIJEDNOST
1	V - 924	100218088	Grundlegung zur Metaphysik der Sitten	Kant	<s. a. >	Leipzig	2558/56	
2	V - 966	100219072	Liberalism	Hobhouse	<s. a. >	London		Provjeriti godinu
3	V - 980	100220011	Nacrt socijalne filozofije	Gumplowicz	<s. a. >	Osijek		Provjeriti godinu
4	V - 1.911	100214048	Philosophie der Rechtsgeschichte	Dulceit	<s. a. >	Heidelberg	1283/52	Provjeriti godinu
5	V - 2.575	100107009	Masters of political thought		<s. a. >	Boston		Provjeriti godinu
6	V - 2.703	100107019	Locke berkeley Hume	Morris	1059	London	655/62	
7	V - 2.928	100109061	Essays in the liberal idea of freedom	Spitz	1064	Tuchson	513/65	333
8	V - 717 add.1	100225072	Iurisconsulti cheriani	Gribaldi	1552	Venetiis	480/1935	333
9	V - 717 add.2	100225073	Petri cordati adolescentis bellunensis praeludia		1553	Florentiae	872/1960	333
10	V - 1.968	100216085	De methodo ac ratione studendi, libri tres	Gribaldus	1558	Venetiis	1585/53	333
11	V - 287	100228047	Del governo de regni et delle republiche antiche e t moderne	Sansovino	1567	Venetia	392/1927	333
12	V - 1.882	100215050	Dello stato delle republiche secondo la mente di A ristotele con essempli moderni giornate otto	Gozzi	1591	Venetia	393/1927	333
13								

Slika 14 - prikaz tablice izvještaja nakon provjere godina i pridjeljivanja vrijednost

Makronaredbe kao funkcije

Funkcije su VBA izjave koje su stvorene da bi se pomoću njih izvela ili izvršila bilo koja vrsta operacija koje će se često koristiti u računalnom kodu.²¹ Stvorene funkcije kao i predefinirane vraćaju samo jednu vrijednost. Za razliku od podprocedura, funkcije se ne mogu izvršavati izravno. Postoje samo dva načina na koje je moguće izvršiti funkcije;

- od strane druge procedure
- putem formule radnog lista²²

Zapravo, možemo reći da se vlastite funkcije mogu pozivati kao i one ugrađene te će primjeri za to biti prikazani dalje u radu.

S obzirom na više od 450 ugrađenih funkcija u Excelu moguće je zapitati se da li je uopće potrebno stvaranje novih funkcija odnosno koji su razlozi za to. Glavni razlog je što funkcije mogu vrlo uspješno zamijeniti pisanje dugih formula koje mogu biti nepregledne i komplicirane za rad te kao

²¹ Prema James, J., Excel 2021 for beginners, Independently published, 2021., str. 360.

²² Van Niekerk, M., VBA Automation for Excel 2019 Cookbook, Packt Publishing, Birmingham, 2020., str. 102.

drugo, pomoću funkcija je moguće izvesti operacije koje bi inače bilo nemoguće za izvesti.²³ Također je moguće da se u nekim procedurama često ponavlja isti izračuni pa je za takve slučajeve korisno izraditi funkciju koja će osim ubrzavanja posla isključiti mogućnosti pogrešaka pri računanju.

Primjeri funkcija

Kako je u prethodnom poglavlju kao jedan od razloga za kreiranje vlastitih funkcija spomenut i taj da je zbog čestog ponavljanja određenih izračuna oportuno stvaranje vlastite funkcije koja će ubrzati proces računanja i osigurati nas od možebitnih grešaka.

Obzirom da je prošlo tek nešto više od dva mjeseca od uvođenja Eura u Republici Hrvatskoj još uvijek često postoji potreba za preračunavanjem Kune u Euro pa će kao primjer sa nekoliko razrada biti će prikazana funkcija koja preračunava uneseni iznos u Euro.

Function EuroRacun(kune As Double) As Double

EuroRacun = kune / 7.53450

End Function

Svaka funkcija započinje sa „Function“ i završava sa „End Function“. U konkretnom slučaju definirane su varijable kune i sama funkcija sa nazivom EuroRacun kao numerički tip podataka double. Kao rezultat funkcija daje kvocijent zadane vrijednosti i fiksnog borja 7.53450. Sama po sebi funkcija nema neku svrhu jer je takvo računanje vrlo jednostavno i bez njezine pomoći, ali zbog mogućnosti njezine daljnje razrade i korištenja u složenijim subprocedurama ima smisla.

Kako bi dobili prikladnije sučelje za unos vrijednosti za računanje korišten je InputBox te MsgBox za prikaz rezultata u kodu kako slijedi;

²³ Alexander, M., Kusleika, D., previously by Walkenbach, J., Excel 2019 Bible, John Wiley & Sons, Indianapolis IN, 2019., str. 947.

Function EuroRacun() As Double

Dim kune As Double

kune = InputBox("Unesite iznos u kunama:")

Dim euro As Double

euro = kune / 7.53450

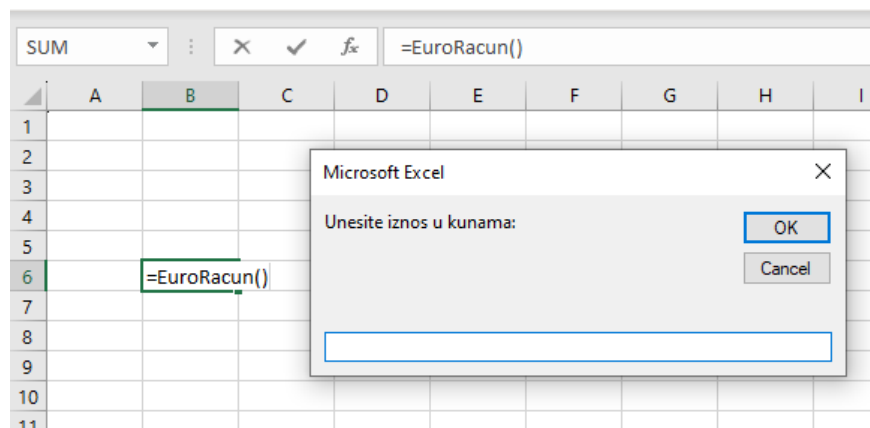
euro = Round(euro, 2)

MsgBox "Iznos u eurima: " & euro

EuroRacun = euro

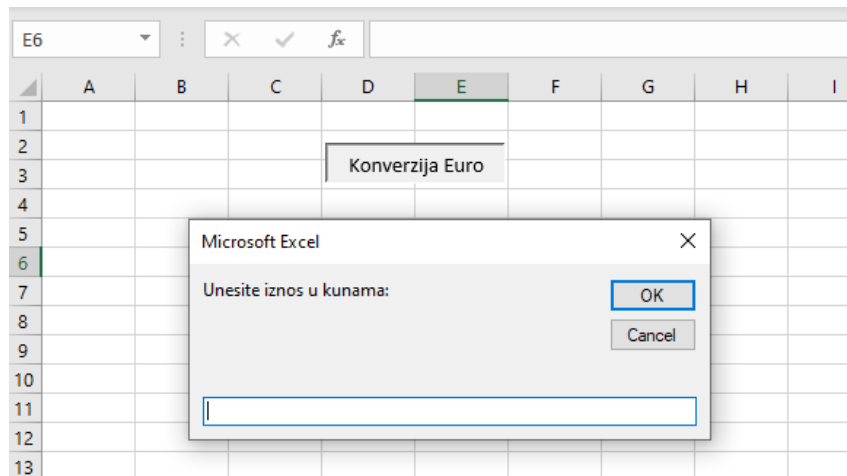
End Function

Ipak ovakvu funkciju je moguće pozivati samo iz radnog lista kao i bilo koju drugu ugrađenu funkciju ili iz VBE editora što je nepraktično u slučaju čestog korištenja.



Slika 15 - pozivanje funkcije EuroRacun

Kako bi doskočili rješavanju takvog problema moguće je koristiti gumb koji će biti na radnom listu stalno dostupan. Klikom na gumb dobivamo identičan InputBox kao i u primjeru pozivanja funkcije iz radnog lista



Slika 16 - pozivanje funkcije EuroRacun preko gumba

Za ovakav slučaj kod je malo nadopunjen i izmijenjen jer u prvoj verziji nisu bile uključene mogućnosti da korisnik neće unijeti broj ili će ostaviti unos prazan što je rezultiralo greškama u izvođenju koda.

Function EuroRacun() As Double

Dim kune As Variant

kune = InputBox("Unesite iznos u kunama:")

If Trim(kune) = "" Then

MsgBox "Molimo unesite iznos."

ElseIf IsNumeric(kune) Then

Dim euro As Double

euro = kune / 7.5345

euro = Round(euro, 2)

MsgBox "Iznos u eurima: " & euro

EuroRacun = euro

Else

MsgBox "Molimo unesite broj."

End If

End Function

Varijabla kune je sada definirana sa Variant što omogućuje da primi i druge tipove podataka jer u protivnom dolazi do prekida u radu funkcije. Nadalje, funkcijom Trim uklanjamo sve razmake moguće slučajno upisane te ukoliko je unos prazan, funkcija javlja poruku o potrebi za unosom iznosa, a ukoliko unos postoji, funkcijom IsNumeric se provjerava da li je unesen broj. U slučaju da nije unesen broj dobivamo poruku o potrebi unosa broja, a u protivnom funkcija se izvršava do kraja.

U konačnici, na jednom primjeru vidljivo je kako naoko jednostavna funkcija sa vrlo bazičnim značajkama uz male promjene može dobiti proširenu funkcionalnost te primjenjivost na rješavanje konkretnog zadatka.

Korisničke forme

Već je u prethodnom poglavlju sa primjerima prikazan prednost korištenja funkcija InputBox te MsgBox pomoću kojih je omogućen unos i prikaz podataka u zasebnim dijaloškim okvirima koji imaju određene mogućnosti prilagodbe. Ipak, kada govorimo o svrsishodnosti i prilagođavanju korisničkim potrebama nemaju ni približnih mogućnosti koje nude korisničke forme.²⁴

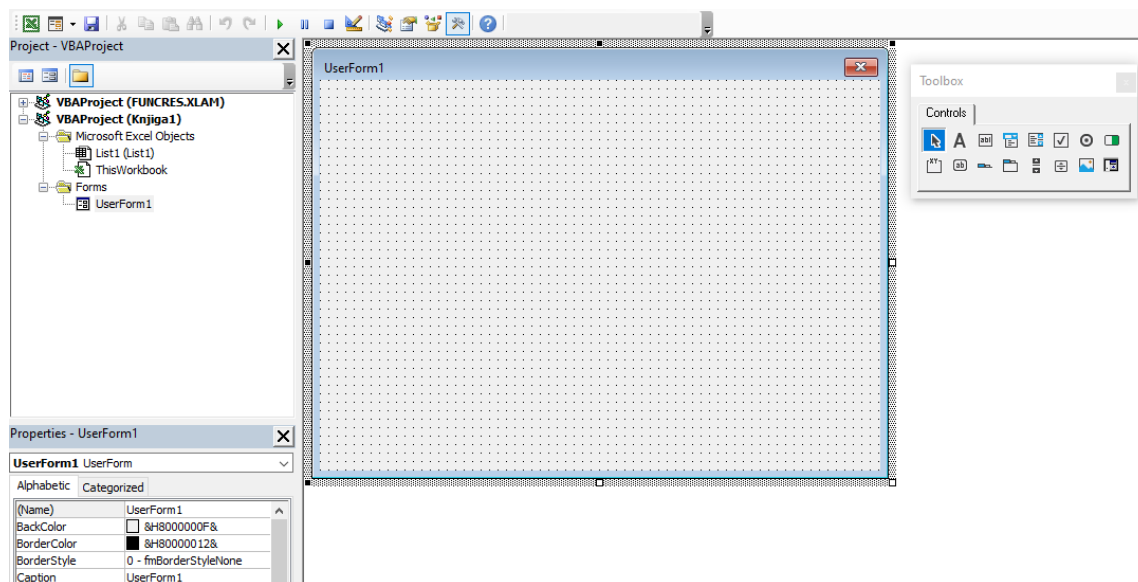
Sama definicija korisničkih formi bi bila „prilagođeni dijaloški okvir koji omogućuje unos podataka koji je kontroliran i lakši za korištenje od strane korisnika.“²⁵ Zapravo korisničke forme su interaktivni obrasci koji nude mogućnost upisivanja i uređivanja podataka te eventualne prikaze bez potrebe za unosom izravno u same ćelije. Korištenje korisničkih formi nudi razne funkcionalnosti za rad sa samim podacima te samim time štedi vrijeme i poboljšava produktivnost.

²⁴ Prema Alexander, M., Kusleika, D., previously by Walkenbach, J., Excel 2019 Bible, John Wiley & Sons, Indianapolis IN, 2019., str. 962.

²⁵ Prijevod: James, J., Excel 2021 for beginners, Independently published, 2021., str. 446.

Prema autoru Jordanu postoji nekoliko razloga odnosno slučajeva u kojima bi bilo poželjno koristiti korisničke forme;

- slučajevi kada bi korisnici trebali moći donositi odluke koje nije moguće zabilježiti u MsgBox ili InputBox obrascima
- u slučaju da makronaredbe zahtijevaju unos nekoliko različitih unosa podataka
- slučaj kada je korisnički unos prekompleksan za MsgBox ili InputBox obrasce
- ograničavanje korisničkog izbora ili unosa uspostavljanjem kompleksnih provjera podataka i kontrola
- za osiguravanje jednostavnog premještanja unosa podataka²⁶



Slika 17 - kreiranje korisničke forme

Kreiranje i kontrole korisničkih formi

Kao i kod makronaredbi za kreiranje korisničkih formi potrebno je otvoriti sučelje VBE editora gdje u radnu knjigu umećemo novu korisničku formu. Sama forma ima veći broj svojstava koje je moguće mijenjati prema potrebama i preferencijama, a moguće je vizualno prilagoditi i oblikovati

²⁶ Prijevod: James, J., Excel 2021 for beginners, Independently published, 2021., str. 446

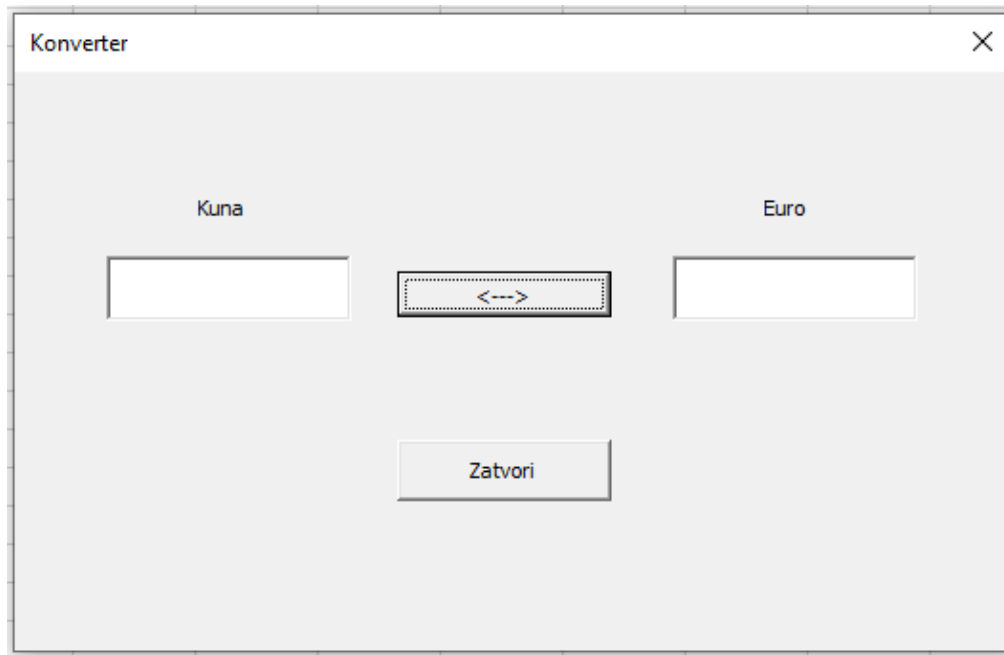
sam prikaz korisničke forme od njezine veličine do pozicioniranja unutar otvorenog radnog lista te znatan broj drugih svojstava koje je moguće mijenjati ili unutar već ponuđenih zadanih opcija ili samostalno pridjeljivanjem određenih vrijednosti ili svojstava.

Otvaranjem nove korisničke forme automatski se pojavljuje alatna traka sa ponuđenim kontrolama koje je moguće dodati na formu po principu povuci i ispusti. Kao i kod same korisničke forme, svaka kontrola ima veći broj promjenjivih svojstava od kojih je dio zajednički za sve kontrole dok je dio zaseban za svaku pojedinu kontrolu ovisno o njezinoj namjeni.

Kako bi imali koristi od napravljene forme odnosno kreiranih kontrola potreban je VBA kod koji će uz svaku kontrolu povezati neki događaj te govorimo o tzv. pokretačima događaja, eng. event handler. Pokretači događaja su zapravo podprocedure koje se izvršavaju prilikom nekog konkretnog događaja kao što je klik na određeni gumb, otvaranje nekog obrasca i slično. Aktivacijom samog događaja VBA pronalazi programski kod koji je povezan uz određenu kontrolu te pokreće podproceduru koja izvršava zadanu akciju. U primjerima korisničkih formi vidljivo je kako se svaki pokretač događaja povezuje uz određenu kontrolu pomoću naziva kontrole te imena samog događaja među kojima je znak donja crtica, eng. underscore, npr. *preracunaj_Click()*. Bez takvih pokretača događaja sama korisnička forma nema smisla jer bez obzira na njezin vizualni prikaz, klikom na bilo koji od umetnutih kontrola nećemo dobiti nikakav odaziv budući da oni djeluju jedino uz već detaljno opisane potprocedure koje izvršavaju određene naredbe ili akcije.

Primjeri korisničkih formi

U jednom od prijašnjih primjera prikazan je konverter valuta pomoću InputBox te MsgBox funkcija koji sasvim dobro odrađuje svoj posao, ali kako bi sučelje bilo preglednije te imalo još neke mogućnosti u sljedećem primjeru pomoću korisničke forme biti će prikazan konverter koji ima mogućnost računanja eura u kune te obrnuto.



Slika 18 - korisnička forma Konverter

U korisničkoj formi prikazanoj na slici 18. vidljivo je kako postoje dvije TextBox kontrole te ovisno u koju od njih je upisana vrijednost klikom na središnji gumb dobivamo rezultat konverzije u suprotnom TextBox-u. Također, dodan je gumb za zatvaranje, a sama forma realizirana je pomoću sljedećeg VBA koda;

```
Private Sub preracunaj_Click()  
    Dim kunavrijednost As Double  
    Dim eurovrijednost As Double  
    If IsNumeric(txtkuna.Value) Then  
        kunavrijednost = CDb(txtkuna.Value)  
        eurovrijednost = Round(kunavrijednost / 7.5345, 2)  
        txtEuro.Value = eurovrijednost  
    ElseIf IsNumeric(txtEuro.Value) Then
```

```
eurovrijednost = CDbI(txtEuro.Value)
kunavrijednost = Round(eurovrijednost * 7.5345, 2)
txtkuna.Value = kunavrijednost
```

Else

```
MsgBox "Molimo unesite vrijednost u jedno od polja."
```

End If

End Sub

```
Private Sub zatvori_Click()
```

```
Unload Me
```

```
End Sub
```

Kako je vidljivo iz samog koda računanje je provedeno identično kao i u slučaju već spomenute funkcije, gdje su definirane dvije varijable kojima se pomoću funkcije CDbI pridjeljuje vrijednost unesene u TextBox kontrolu. Ovisno o tome u koju TextBox kontrolu je unesena vrijednost, u drugoj se prikazuje rezultat konverzije. U slučaju da nije unesena niti jedna vrijednost korišten je MsgBox kako bi upozorio korisnika na isto. Također, ovakva forma je vrlo jednostavna, ali konkretno prikazuje mogućnosti zbog kojih je samo korištenje korisničkih formi poželjno i potrebno.

U narednom prikazu korisničke forme korišten je već obrađen primjer stvorene podprocedure koja automatizira unošenje naziva kolona. Svaka kolona predstavlja određeni tip podataka koje je potrebno unijeti, a kako bi ubrzali i pojednostavili takav proces korištena je korisnička forma sa sljedećim VBA kodom;

```
Private Sub CommandButton1_Click()
```

```
Dim redak As Long
```

```
redak = Range("A" & Rows.Count).End(xlUp).Row + 1
```

Range("A" & redak).Value = signatura.Value

Range("B" & redak).Value = naslov.Value

Range("C" & redak).Value = autor.Value

Range("D" & redak).Value = god_izd.Value

Range("E" & redak).Value = mj_izd.Value

Range("F" & redak).Value = inv_br.Value

Range("G" & redak).Value = vrijednost.Value

signatura.Value = ""

naslov.Value = ""

autor.Value = ""

god_izd.Value = ""

mj_izd.Value = ""

inv_br.Value = ""

vrijednost.Value = ""

End Sub

Private Sub CommandButton2_Click()

signatura.Value = ""

naslov.Value = ""

autor.Value = ""

god_izd.Value = ""

```
mj_izd.Value = ""
```

```
inv_br.Value = ""
```

```
vrijednost.Value = ""
```

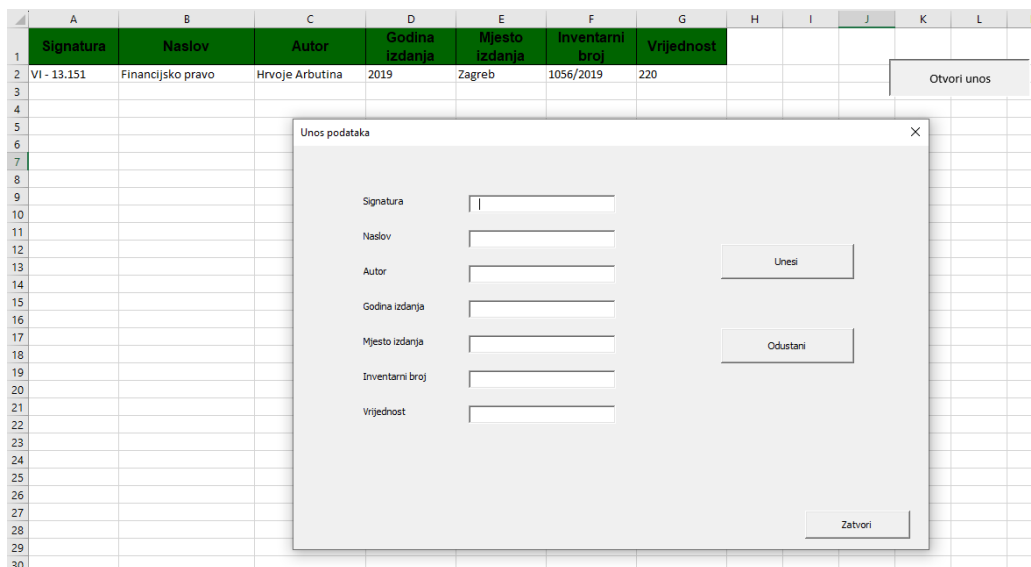
```
End Sub
```

```
Private Sub CommandButton3_Click()
```

```
    Unload Me
```

```
End Sub
```

Na slici 19. je vizualni prikaz forme za koju je kreiran VBA kod koji u prvi slobodan red svake kolone unosi podatke prethodno upisane u svaku TextBox kontrolu. Definirani su gumbi za unos podataka u redove prema nazivu kolone, brisanje podataka iz svih TextBox kontrola te zatvaranje same forme, a kako bi brže pristupali ovoj korisničkoj formi na radnom listu je stvoren prečac „Otvori unos“



Slika 19 - prikaz korisničke forme za unos podataka

Zaključak

Budući sam već u uvodu istaknuo kako je Excel jedan od najvažnijih programskih rješenja u poslovnoj domeni sa primjenom u vrlo širokom dijapazonu poslovnih entiteta, pišući ovaj rad taj dojam se još intenzivirao. Izuzetna prilagodljivost, brzina, jednostavnost te još čitav niz značajki čine ga savršenim rješenjem za različite systemske zadatke koje prate poslovanje i poslovne procese.

Pošto je naglasak ovog rada bio na makronaredbama i korisničkim formama važno je istaknuti kako s jedne strane makronaredbe značajno ubrzavaju i olakšavaju radne procese dok s druge strane korisničke forme vizualno definiraju i konkretiziraju manipulaciju podacima čineći savršenu sinergiju za različite vremenske i financijske uštede, povećanja učinkovitosti te još mnogostruke prednosti.

Ukoliko su kvalitetno izrađene, makronaredbe mogu stvoriti značajne uštede te samim time intenzivirati produktivnost i kvalitetu te smanjiti mogućnosti za pogreške dok korisničke forme rade bitnu razliku u unosu i obradi podataka čime poboljšavaju korisničko iskustvo i olakšavaju rad. Bilo da se oslanjamo na gotova programska proširenja ili sami izrađujemo neko programsko rješenje važno je imati na umu kako je lakše postizanje cilja ono što čini razliku i zbog čega bi se uopće trebalo upuštati u potragu za novim načinima i mogućnostima u automatizaciji.

U konačnici, premda je ovaj rad tek površinski objedinio te djelomice pokrio neke aspekte teme, vjerujem da automatizacija rada ima svjetlu budućnost što će doprinijeti da Excel vrlo vjerojatno ostane jedan od najkorištenijih programskih alata poslovnog svijeta i samim time imati zagantirano mjesto u poslovnim aktivnostima vremena koje je pred nama.

Popis slika:

Slika 1 - popis dostupnih dodataka	3
Slika 2 – prilaz podataka pomoću „Descriptive Statistic“ analize	4
Slika 3 - Microsoft trgovina dodataka za Office paket	4
Slika 4 - aktivacija kartice programera	5
Slika 5 - Prikaz VBE sučelja.....	7
Slika 6 - Object Browser u VBE sučelju	9
<i>Slika 7 - snimanje makronaredbe pomoću statusne trake</i>	<i>13</i>
<i>Slika 8 - snimanje makro naredbe pomoću kartice View.....</i>	<i>13</i>
<i>Slika 9 - snimanje makronaredbe pomoću kartice programera</i>	<i>14</i>
<i>Slika 10 - spremanje makronaredbe.....</i>	<i>14</i>
<i>Slika 11 - mogućnosti spremanja radne knjige Excel-a</i>	<i>15</i>
Slika 12 – prikaz snimljene makronaredbe Primjer_1.....	17
Slika 13 - prikaz makronaredbe Formatiranje_stranice	18
Slika 14 - prikaz tablice izvještaja nakon provjere godina i pridjeljivanja vrijednost	20
Slika 15 - pozivanje funkcije EuroRacun	22
Slika 16 - pozivanje funkcije EuroRacun preko gumba.....	23
Slika 17 - kreiranje korisničke forme	25
Slika 18 - korisnička forma Konverter.....	27
Slika 19 - prikaz korisničke forme za unos podataka.....	30

Literatura:

Alexander, M., Kusleika, R., Excel 2019 Power Programming, John Wiley & Sons, Indianapolis IN, 2019.

Alexander, M., Kusleika, D., previously by Walkenbach, J., Excel 2019 Bible, John Wiley & Sons, Indianapolis IN, 2019.

Blackwood, N., Make the most of macros, In the Black, July 2011., str. 48-58.

Jordan, J., Excel 2021 for beginners, Independently published, 2021.

Lomax, P., VB & VBA in a Nutshell: The Language, O'Reilly & Associates, Sebastopol CA, 1998.

Van Niekerk, M., VBA Automation for Excel 2019 Cookbook, Packt Publishing, Birmingham, 2020.