



Upošljavanje VBA

U OVOM DELU:

7	Programiranje Worda	111
8	Programiranje Excela	137
9	Programiranje PowerPointa	163
10	Programiranje Accessa	191
11	Programiranje Outlooka	219

Programiranje Worda

U ovom poglavlju

- *Zadaćete, otvarati, kreirati, čuvati i zatvarati dokumenta.*
- *Upotrebićete objekat Range za rad sa tekстом.*
- *Koristićete VBA za manipulisanje karakterima, rečima, rečenicama i pasusima.*

MICROSOFT WORD JE VELIKI, KOMPLEKSNI PROGRAM, PA TAKO NIJE NIKAKVO IZNENAĐENJE TO ŠTO JE NJEGOV SPISAK OBJEKATA JEDNAKO VELIKI I KOMPLIKOVAN. NA SREĆU, KAO ŠTO I VEĆINA LJUDI KORISTI SAMO MALI DEO WORDOVIH MOGUĆNOSTI, TAKO ĆETE VEROVATNO I VI KORISTITI SAMO NEKOLIKO WORDOVIH OBJEKATA U SVOM VBA PROGRAMIRANJU. U STVARI, U WORDU POSTOJE SAMO TRI NIVOJA OBJEKATA O KOJIMA UGLAVNOM TREBA DA BRINETE: SAM WORD, KOGA PREDSTAVLJA OBJEKAT `APPLICATION`; WORD DOKUMENTI, KOJE PREDSTAVLJA OBJEKAT `DOCUMENT`; I TEKST U TIM DOKUMENTIMA, KOJI JE PREDSTAVLJEN RAZNIM OBJEKTIMA, UKLJUČUJUĆI I `CHARACTER` (KARAKTER), `SENTENCE` (REČENICA) I `PARAGRAPH` (PASUS). U OVOM POGLAVLJU ĆEMO SE FOKUSIRATI NA PROGRAMIRANJE OVIH OSNOVNIH OBJEKATA.

Rad sa dokumentima

U Wordu se, u hijerarhiji objekat, objekat Document nalazi odmah ispod objekta `Application`. Kao što ćete videti u narednim odeljcima, VBA možete koristiti za kreiranje novih dokumenata, otvaranje ili brisanje postojećih dokumenata, čuvanje i zatvaranje otvorenih dokumenata i za još mnogo toga.

Specificiranje objekta Document

Ako treba da uradite nešto sa dokumentom ili ako treba da radite sa objektom koji se nalazi u određenom dokumentu (kao što je deo teksta), treba da kažete Wordu koji dokument da koristi. VBA vam daje tri načina da to uradite:

Pomoću objekta `Document` Objekat `Document` predstavlja skup svih otvorenih dokumenata. Da biste specificirali određeni dokument, upotrebite ili njegov indeksni broj (gde 1 predstavlja prvi otvoreni dokument) ili napišite ime dokumenta pod navodnicima. Na primer, ako je `Memo.doc` bio prvi otvoreni dokument, sledeće dve naredbe bile bi ekvivalentne:

```
Documents("Memo.doc")
```

```
Documents(1)
```

Pomoću objekta `ActiveDocument` Objekat `ActiveDocument` predstavlja dokument koji je trenutno u fokusu.

Pomoću objekta `ThisDocument` Objekat `ThisDocument` predstavlja dokument u kome se VBA kod izvršava. Ako vaš kod radi samo sa objektima koji se nalaze u dokumentu u kome je smešten i sam kod, možete da koristite objekat `ActiveDocument`. Međutim, ako vaš kod radi i sa drugim dokumentima, koristite `ThisDocument` kad god treba da obezbedite da kod deluje samo na dokument koji sadrži proceduru.

Otvaranje dokumenta

Da biste otvorili datoteku dokumenta, koristite metodu `Open` zbirke `Documents`. Metoda `Open` ima tuce argumenata koje možete da upotrebite za fino podešavanje otvaranja dokumenta; inače, samo jedan argument je obavezan. Evo pojednostavljene sintakse koja pokazuje jedan zahtevani argument (ostale argumente možete da vidite ako potražite metodu `Open` u sistemu VBA pomoći):

```
Documents.Open(ImeDatoteke)
```

ImeDatoteke Ime datoteke dokumenta, uključujući i drajv i direktorijum koji sadrže datoteku.

Na primer, da biste otvorili dokument Letter.doc iz direktorijuma C:\My Documents, koristili biste sledeću naredbu:

```
Documents.Open "C:\My  
Documents\Letter.doc"
```

Objekat RecentFiles

Drugi način otvaranja Word dokumenata je upotreba objekata RecentFiles koji predstavlja skup najskorije korišćenih datoteka, a koje se prikazuju pri dnu Wordovog menija File. Svaki element iz ove liste je RecentFile objekat.

Objekat RecentFile se specificira kao RecentFiles(*indeks*), gde je indeks ceo broj koji specificira datoteku sa kojom želite da radite. Najskorije korišćena datoteka se označava indeksom 1, druga najskorije korišćena datoteka označena je indeksom 2, i tako dalje.

Evo nekih osobina objekta RecentFile koje treba razmotriti:

`RecentFile.Name` Vraća ime datoteke specificiranog objekta RecentFile.

`RecentFile.Path` Vraća ime putanje specificiranog objekta RecentFile.

Evo koda koji govori Wordu da otvori najskorije korišćenu datoteku:

```
With RecentFiles(1)  
    Documents.Open.Path & "\" & .Name  
End With
```

Bilo bi zgodno kada bi Word otvarao najskorije korišćenu datoteku svaki put kada startujete program. Ako želite da Word izvrši neki kod svaki put kada se startuje, pratite sledeće korake:

1. Otvorite projekat Normal u Project Exploreru VBA Editora.
2. Kreirajte novi modul i preimenujte ga u **AutoExec**.
3. U tom modulu napravite Sub proceduru pod imenom Main.
4. Upišite svoj kod u tu proceduru.

Listing 7.1 pokazuje primer procedure Main koja otvara najskorije korišćenu datoteku na samom početku.

SAVET

Maksimalna vrednost indeksa zadaje se osobinom `RecentFiles.Maximum`. Primetite, takođe, da ovu osobinu možete podesiti. Na primer, sledeća naredba postavlja maksimalnu vrednost na 9 (to je najveća vrednost koju dozvoljava Word):

```
RecentFiles.Maximum = 9/
```

Listing 7.1: Procedura za otvaranje najskorije korišćenog dokumenta prilikom startovanja Worda

```

Sub Main()
    With RecentFiles(1)
        Documents.Open.Path & " \ " &
        .Name
    End With
End Sub

```

N A P O M E N A

Kod koji se koristi u listinzima za ovo poglavlje može se naći na mom web sajtu, na sledećoj adresi:

<http://www.mcfedries.com/ABGVBA/Chapter07.doc>

Kreiranje novog dokumenta

Ako treba da napravite novi dokument, koristite metodu Add zbirke Documents:

```
Documents.Add(Template, NewTemplate, DocumentType, Visible)
```

Template (šablon) Ovaj opcioni argument specificira datoteku šablona koju treba koristiti kao osnovu novog dokumenta. Upišite string koji sadrži putanju i ime .DOT datoteke. Ako izostavite ovaj argument, Word kreira novi dokument na osnovu šablona Normal.

newTemplate (noviŠablon) Ako postavite ovaj opcioni argument na True, Word će kreirati novi šablon.

DocumentType (tipDokumenta) Opcioni argument koji definiše tip dokumenta koji se kreira. Koristite neku od sledećih konstanti:

wdNewBlankDocument Kreira novi, prazan Word dokument (to je podrazumevana konstanta).

wdEmailMessage Kreira novu e-mejl poruku.

wdNewFrameset Kreira novi frejmsset web stranice.

wdNewWebPage Kreira novu web stranicu.

wdNewXMLDocument Kreira novi XML dokument.

Visible (vidljiv) Ovo je opciona Bulova vrednost koja određuje da li Word prikazuje novi dokument u vidljivom prozoru. Podrazumevana vrednost je True; koristite False kada želite da kreirate dokument koji se neće prikazivati u vidljivom prozoru.

Pošto su svi argumenti metode Add opcioni, osnovni Word dokument možete kreirati na osnovu šablona Normal pomoću sledeće jednostavne naredbe:

```
Documents.Add
```

Čuvanje dokumenta

Najgora noćna mora svakog korisnika Worda je nestanak struje ili prekid napajanja koji isključuje Word, pa i sam računar, kada vam je otvoren jedan ili više dokumenata sa nesačuvanim izmenama. Znam ljude koji su izgubili mnogo *radnih časova* kada im se to desilo. Sami sebi govorimo da treba češće da pamtimo izmene, ali se to lako zaboravlja dok se radi. Čak i Wordova mogućnost za automatski oporavak, AutoRecover, ne radi uvek onako kako se reklamira, tako da se na nju ne možete osloniti.

N A P O M E N A

Čim kreirate novi dokument, on automatski postaje aktivni dokument u prozoru Worda, tako da možete da koristite osobinu ActiveDocument da biste radili sa novim dokumentom (na primer, da ga sačuvate, o čemu će biti reči u narednom odeljku).

Korišćenje metode Save

Na sreću, VBA se pokazao vrlo korisnim u rešavanju ovog problema, zato što se lako piše procedura koja se brine o čuvanju. Međutim, pre nego što predemo na to, pogledajmo nekoliko osnovnih osobina i metoda objekta Document.

Prva metoda je metoda Save:

```
Document.Save
```

Document Ovo je referenca za dokument koji želite da sačuvate.

Na primer, sledeća naredba će sačuvati aktivan dokument:

```
ActiveDocument.Save
```

Ako radite sa velikim dokumentom, možda nećete hteti da nepotrebno izvršavate metodu Save zato što ona može i da potraje. To možete izbeći tako što ćete prvo proveriti osobinu dokumenta Saved. Ako se vrati vrednost False, to znači da dokument ima nesačuvanih izmena. Evo primera:

```
If ActiveDocument.Saved = False Then  
    ActiveDocument.Save  
End If
```

Metoda Save se neće uspešno izvršiti ako se radi o novom dokumentu koji nikada ranije nije sačuvan. Kako to znate? Postoje dva načina:

- Ako ste upravo kreirali dokument korišćenjem metode Add.

- Proverite osobinu dokumenta Path. Za dokument koji je već sačuvan, osobina Path vraća dražv i direktorijum u kome je dokument smešten. (Primetite da string koji vraća osobina Path nema poslednju kosu crtu; na primer, "C:\My Documents"). Međutim, ako dokument nikada nije sačuvan, osobina Path vraća prazan string (" ").

Evo dela koda koji proverava osobinu Path pre nego što pokuša da sačuva aktivni dokument:

```
If ActiveDocument.Path <> " " Then  
    ActiveDocument.Save  
End If
```

Listing 7.2 predstavlja proceduru BezbednoCuvanje koja kombinuje ove dve provere tako da se izbegava čuvanje novih ili neizmenjenih dokumenata.

Listing 7.: Procedura koja izbegava čuvanje novih ili neizmenjenih dokumenata

```
Sub BezbednoCuvanje()  
    With ActiveDocument  
        If .Path <> " " And .Saved = False Then  
            .Save  
        End If  
    End With  
End Sub
```

Korišćenje metode SaveAs

Ako je dokument nov, bolje je koristiti metodu SaveAs.

```
Document.SaveAs(ImeDatoteke, FormatDatoteke)
```

Dokument	Objekat Document koji želite da sačuvate u drugoj datoteci.
ImeDatoteke	(opcion) Puno ime nove datoteke dokumenta, uključujući i dražv i direktorijum u kome želite da se smesti. Ako ne zadate ovu vrednost, Word koristi tekući direktorijum i podrazumevani naziv (kao što je Doc1.doc).
FormatDatoteke	(opcion) Format datoteke u kome dokument treba sačuvati. Možete da koristite ili unapred definisanu wdSaveFormat konstantu ili broj koji definiše format (wdWordDocument je podrazumevana konstanta):

Format Datoteke	Konstanta	Celobrojna vrednost
Word dokument	wdFormatDocument	0
Šablon dokumenta	wdFormatTemplate	1
Samo tekst	wdFormatText	2
Samo tekst sa prelomom linija	wdFormatTextLineBreaks	3
MS-DOS tekst	wdFormatDOSText	4
MS-DOS tekst sa prelomom linija	wdFormatDOSTextLineBreaks	5
Obogaćeni tekstualni format	wdFormatRTF	6
Unikod tekst	wdFormatUnicodeText	7
Web stranica	wdFormatHTML	8
Web arhiva	wdFormatWebArchive	9
XML dokument	wdFormatXML	12

Obe sledeće naredbe su ekvivalentne (to jest, obe će sačuvati aktivan dokument kao web stranu):

```
ActiveDocument.Save "index.html",  
wdFormatHTML  
ActiveDocument.Save "index.html", 8
```

NAPOMENA

Ovo je pojednostavljena sintaksa metode `SaveAs`. Da biste videli svih 11 argumenata sa kompletnom sintaksom, potražite metodu `SaveAs` u sistemu VBA pomoći.

Zatvaranje dokumenta

Kada vam dokument više nije potreban, možete da smanjite gomilu na ekranu tako što ćete, što se Worda tiče, zatvoriti odgovarajući objekat `Document` pomoću metode `Close`:

```
Document.Close (SacuvajIzmene, OriginalniFormat, UsmeriDokument)
```

Dokument	Objekat <code>Document</code> koji želite da zatvorite.						
SacuvajIzmene	(opciono) Ako je dokument modifikovan, ovaj argument određuje da li će Word sačuvati te izmene: <table> <tr> <td><code>wdSaveChanges</code></td><td>Sačuvaće izmene pre zatvaranja.</td></tr> <tr> <td><code>wdDoNotSaveChanges</code></td><td>Ne čuva izmene.</td></tr> <tr> <td><code>wdPromptToSaveChanges</code></td><td>Pita korisnika da li želi da sačuva izmene (ovo je podrazumevana vrednost).</td></tr> </table>	<code>wdSaveChanges</code>	Sačuvaće izmene pre zatvaranja.	<code>wdDoNotSaveChanges</code>	Ne čuva izmene.	<code>wdPromptToSaveChanges</code>	Pita korisnika da li želi da sačuva izmene (ovo je podrazumevana vrednost).
<code>wdSaveChanges</code>	Sačuvaće izmene pre zatvaranja.						
<code>wdDoNotSaveChanges</code>	Ne čuva izmene.						
<code>wdPromptToSaveChanges</code>	Pita korisnika da li želi da sačuva izmene (ovo je podrazumevana vrednost).						

OriginalniFormat	Specificira format koji treba koristiti kada se dokument pamti:
wdOriginalFormat	Sačuvaće dokument u njegovom originalnom formatu (ovo se podrazumeva).
wdWordDocument	Sačuvaće dokument u Word formatu.
wdPromptUser	Pita korisnika da li želi da sačuva dokument u njegovom originalnom formatu.
UsmeritiDokument	Ako je njegova vrednost True, ovaj argument govori Wordu da usmeri dokument do sledećeg primaoca.

Na primer, sledeća naredba zatvara aktivni dokument i čuva sve izmene:

```
ActiveDocument.Close wdSaveChanges
```

Primer: Pravljenje rezervnih kopija dokumenata

Hajde da iskoristimo osobine i metode objekta Document koje smo do sada naučili i da kreiramo proceduru koja ne samo da čuva dokument već i pravi rezervnu kopiju na disketi. Listing 7.3 ilustruje proceduru NapraviRezervnuKopiju koja sve to radi pomoću metode SaveAs, kao i nekoliko drugih metoda i osobina objekta Document.

Listing 7.3: Procedura koja pravi rezervnu kopiju aktivnog dokumenta na floppy disku.

```
Sub NapraviRezervnuKopiju()  
    Dim rezervnaDatoteka As String  
    Dim tekucaDatoteka As String  
    With ActiveDocument  
        '  
        ' Nemojte nista da radite ako je dokument neizmenjen ili novi  
        '  
        If .Saved Or .Path = "" Then Exit Sub  
        '  
        'Oznacite trenutnu poziciju u dokumentu  
        '  
        .Bookmarks.Add Name:="PoslednjaPozicija"  
        '  
        ' Iskljucite osvezavanje ekrana  
        '  
        Application.ScreenUpdating = False  
        '  
        ' Sacuvajte datoteku  
        '  
        .Save  
        '  
    End With  
End Sub
```

```
' Smestite putanju tekuce datoteke, napravite putanju za datoteku
rezervne kopije,
'
' a zatim je sacuvajte na drajvu A
'
tekucaDatoteka = .FullName
rezervnaDatoteka = "A:\" + .Name
.SaveAs FileName:=rezervnaDatoteka
End With
'
' Zatvorite rezervnu kopiju (koja je sada aktivna)
'
'
'ActiveDocument.Close
'
' Ponovo otvorite tekucu datoteku
'
Documents.Open FileName:=tekucaDatoteka
'
' Vratite se na poziciju pre pravljenja rezervne kopije
'
Selection.GoTo What:=wdGoToBookmark, Name:="PoslednjaPozicija"
'
' Vratite azuriranje ekrana
'
Application.ScreenUpdating = True
End Sub
```

Nakon deklarisanja par promenljivih, ova procedura proverava da vidi da li je operacija pravljenja rezervne kopije neophodna. Drugim rečima, ako dokument nema nesačuvanih izmena (osobina `.Saved` vraća vrednost `True`) ili ako se radi o novom, nesačuvanom dokumentu (osobina `Path` vraća `" "`), procedura se napušta (izvršavanjem komande `Exit Sub`).

U suprotnom se kreira objekat `Bookmark` da bi se sačuvala tekuća pozicija u dokumentu. (Ovim se obezbeđuje da se kasnije, kada ponovo otvorimo dokument nakon izvršavanja metode `SaveAs`, korisnik vraća na svoju poziciju u dokumentu). `Bookmarks` je skup koji čuva sve definisane oznake na strani u datom objektu `Document`. Svaki element ovog skupa je `Bookmark` objekat. Da bismo dodali oznaku pozicije u dokumentu koristimo metodu `Add` na sledeći način:

```
Document.Bookmarks.Add Name:=ImeOznake
```

Document Objekat Document sa kojim želite da radite.

ImeOznake String koji specificira ime oznake pozicije u dokumentu (bookmark).

Sledeća naredba isključuje ažuriranje ekrana, što znači da korisnik neće videti otvaranje i zatvaranje datoteka koje se dešava malo kasnije u kodu:

```
Application.ScreenUpdating = False
```

Zatim će se datoteka sačuvati korišćenjem metode `Save`. Sada ste spremni za pravljenje rezervne kopije. Prvo, promenljiva `tekucaDatoteka` koristi se za držanje imena pune putanje dokumenta (to jest, drajva, direktorijuma i imena datoteke dokumenta), što se daje osobinom `FullName`. Zatim se pravi ime putanje za datoteku rezervne kopije pomoću sledeće naredbe:

```
rezervnaDatoteka = "A:\" + .Name
```

Ovo će se koristiti za čuvanje datoteke na drajvu A. Primetite da se ova naredba može lako prilagoditi ako želite da sačuvate datoteku na drugom hard disku ili mrežnom drajvu.

Pravljenje rezervne kopije se odvija preko metode `SaveAs`, koja će sačuvati dokument na putanji koju daje promenljiva `rezervnaDatoteka`. Nakon toga, procedura zatvara datoteku rezervne kopije, ponovo otvara originalnu datoteku i koristi metodu `GoTo` za vraćanje na originalnu poziciju u dokumentu.

Automatizacija procedure za pravljenje rezervne kopije

Umesto da se procedura `NapraviRezervnuKopiju` izvršava "ručno", bilo bi bolje da se zakaže redovno pravljenje rezervnih kopija u određeno vreme ili periodično. To možete uraditi korišćenjem metode `OnTime` objekta `Application`, koja izvršava proceduru u određeno vreme, koristeći sledeću sintaksu:

```
Application.OnTime(When, Name, Tolerance)
```

When Vreme (i datum, ako je potrebno) kada želite da se procedura izvrši. Upišite serijski broj za datum/vreme.

Name Ime (upisano kao tekst) procedure koju treba izvršiti kada dođe vreme zadato argumentom `When`.

Tolerance Ako Word nije spreman za izvršavanje procedure kada dođe vreme zadato argumentom `When`, pokušavaće se ponovo za onoliko sekundi koliko je zadato argumentom `Tolerance`. Ako izostavite ovaj argument, VBA će čekati sve dok Word ne bude spreman.

Najlakši način za upisivanje serijskog broja za vreme argumenta When je korišćenjem funkcije TimeValue:

```
TimeValue(Vreme)

Vreme    String koji predstavlja vreme koje želite da koristite (kao što je "5:00PM"
          ili 17:00").
```

Na primer, sledeća formula izvršava proceduru NapraviRezervnuKopiju u 5:00PM:

```
Application.OnTime _
    When:=TimeValue("5:00PM"), _
    Name:="NapraviRezervnuKopiju"
```

To je u redu, ali ono što zaista želimo jeste da se OnTime metoda izvršava u zadatim intervalima (na primer, za posla sata od ovog trenutka). Da biste to postigli, iskoristite Now + TimeValue(Vreme) za argument When (gde je Vreme interval koji želite da koristite). Na primer, sledeća naredba zakazuje izvršavanje procedure NapraviRezervnuKopiju na pet minuta:

```
Application.OnTime _
    When:=Now + TimeValue("00:05"), _
    Name:="NapraviRezervnuKopiju"
```

Korisnije metode objekta Document

Pored Open, Save, SaveAs i Close, objekti Document imaju i desetine drugih metoda koje vam omogućavaju da uradite sve, od aktiviranja dokumenta da njegovog štampanja. Evo nekih metoda koje ćete češće da koristite:

Document.Activate Aktivira specificirani dokument. Na primer, sledeća naredba aktivira dokument Tirade.doc:

```
Documents("Tirade.doc").Activate
```

Document.PrintOut Štampa specificirani dokument. Kompletne sintaksa ove metode ima ni manje ni više nego 18 argumenata (svi su opcioni).

Evo nekih:

```
Document.PrintOut (Background, Append, Range, OutputFileName, From,
To, Item, Copies, Pages, PageType, PrintToFile, Collate)
```

Document (Dokument) Objekat Document koji želite da štampate.

Background Ako je True, procedura se nastavlja dok Word štampa
(Pozadina) dokument u pozadini. Ako je False, procedura čeka dok se dokument spuluje.

Append Ako je True, izlaz se otvara na kraju datoteke koja je zadata argumentom OutputFileName. Ako je False, izlaz se prepisuje preko sadržaja datoteke.

Range	Definiše opseg teksta za štampu, na sledeći način:
wdPrintAllDocument	Štampa ceo dokument.
wdPrintCurrentPage	Štampa samo tekuću stranu.
wdPrintFromTo	Štampa opseg stranica od početne strane (vidi argument From) do poslednje strane (vidi argument To).
wdPrintRangeOfPages	Štampa opseg stranica (videti argument Pages).
wdPrintPrintSelection	Štampa samo trenutno selektovani tekst.
OutPutFileName	Putanja i ime datoteke u koju želite da stampate dokument. (Primetite da argument PrintToFile mora da se postavi na vrednost True).
From	Ako argument Range ima vrednost wdPrintFromTo, ovaj argument definiše broj stranice od koje treba početi štampanje.
To	Ako argument Range ima vrednost wdPrintFromTo, ovaj argument definiše broj poslednje stranice za štampanje.
Item	Konstanta koja predstavlja ono što želite da stampate. Koristite neku od sledećih vrednosti - wdPrintAutoTextEntries, wdPrintComments, wdPrintDocumentContent, wdPrintKeyAssignments, wdPrintProperties ili wdPrintStyles.
Copies	Broj kopija koje treba odštampati. Podrazumevana vrednost je 1.
Pages	Ako argument Range ima vrednost wdPrintRangesOfPages, ovaj argument definiše opseg stranica (na primer, 4-8, 10).
PageType	Specificira koje stranice treba štampati korišćenjem neke od sledećih konstanti - wdPrintAllPages, wdPrintEvenPagesOnly, wdPrintOddPagesOnly.
PrintToFile	Ako ima vrednost True, Word štampa dokument u datoteci i traži od korisnika da zada ime datoteke.
Collate	Ako ima vrednost True i ako je argument Copies već od 1, štampa ceo dokument, a zatim štampa sledeću kopiju. Ako ima vrednost False, a vrednost argumenta Copies je veća od 1, štampa prvo kopije prve strane, zatim druge i tako dalje.

`Document.PrintPreview` Prikazuje specificirani Document u prozoru Print Preview.

`Document.Redo` Ponovo izvršava akciju (ili akcije) koja je poslednja otkazana.

`Document.Redo(Times)`

`Document` Objekat Document sa kojim želite da radite.

`Times` (opciono) Broj akcija koje želite da ponovo izvršite (podrazumevana vrednost je 1).

`Document.Select` Selektuje sav tekst u specificiranom dokumentu.

`Document.Undo` Poništava akciju (ili akcije) koja je poslednja izvršena.

`Document.Undo(Times)`

`Document` Objekat Document sa kojim želite da radite.

`Times` (opciono) Broj akcija koje želite da otkazete (podrazumevana vrednost je 1).

Rad sa tekstom

Iako u dokumente možete da ubacujete linije, grafike i ostale objekte, tekst je ono čime se bavi Word. Tako da vas neće iznenaditi to što Word ima čitav tovar objekata koji obezbeđuju brojne i razne načine za rad sa tekstom. Sledećih nekoliko odeljaka će vas provesti kroz neke od tih objekata.

Objekat Range

Ako ste koristili VBA sa Excelom, verovatno znate da Excel nema poseban objekat za predstavljanje ćelije. Umesto toga, ćelija se smatra samo instancom generičke klase Range.

Slično, Word nema posebne objekte za njegove najosnovnije tekstualne jedinice: karakter i reč. Kao i Excel, Word smatra ove elemente instancama generičke klase, takođe, objekta Range. Objekat Range se definiše kao kontinualni deo teksta u dokumentu: nekoliko uzastopnih karaktera, nekoliko uzastopnih reči, nekoliko uzastopnih paragrafa ili nešto slično. Opseg (range) može da predstavlja sve - od jednog karaktera do celog dokumenta - dokle god je tekst takvog elementa kontinualan.

Postoje dve osnovne metode za vraćanje objekta Range: metoda Range objekta Document i osobina Range.

Metoda Range

Objekat Document ima metodu Range koja vam omogućava da zadate početne i krajnje tačke opsega. Evo sintakse:

```
Document.Range(Start, End)
```

Document	Objekat Document sa kojim želite da radite.
Start	Pozicija početnog karaktera. Primetite da prvi karakter u dokumentu ima poziciju 0.
End	Pozicija poslednjeg karaktera. Primetite da se ovaj karakter ne uključuje u opseg.

Na primer, sledeće naredbe koriste objektnu promenljivu `mojOpseg` za čuvanje 100 karaktera (od 0 do 99) u aktivnom dokumentu:

```
Dim mojOpseg As Range  
mojOpseg = ActiveDocument.Range(0, 100)
```

Osobina Range

Mnogi Wordovi objekti imaju osobinu `Range` koja vraća objekat `Range`, uključujući i objekte `Paragraph` i `Selection` (o kojima će biti reči kasnije). Ovo je važno zato što ti objekti nemaju izvesne osobine i metode koje su zgodne za manipulaciju sa tekстом. Na primer, objekat `Paragraph` nema osobinu `Italic`. Međutim, tu osobinu ima objekat `Range`, tako da se font paragrafa programski formatira preko njegove osobine `Range`, na sledeći način:

```
ActiveDocument.Paragraphs(1).Range.Italic = True
```

Ova naredba formatira prvi paragraf u aktivnom dokumentu tako da tekst bude iskošen. (Uskoro ću se baviti zbirkom `Paragraphs`.)

Čitanje i izmena opsega teksta

Objekat `Range` ima osobinu `Text` koja vraća tekst iz specificiranog opsega. Osobinu `Text` možete koristiti i za upisivanje teksta u specificirani opseg. Na primer, sledeći deo koda proverava tekst u dokumentu `letter.doc` da bi se videlo da li je prvih pet karaktera jednako stringu `"Dragi"`; ako jesu, tekst se zamenjuje drugim stringom - `"Pozdravi"`:

```
With Documents("letter.doc").Range(0, 5)  
    If .Text = "Dragi" Then  
        .Text = "Pozdravi"  
    End If  
End With
```


Formatiranje teksta

Osobine objekta `Range` takođe obuhvataju i mnoge komande za standardno formatiranje teksta. Evo kratkog pregleda samo nekih od ovih osobina:

`Range.Bold` Vraća vrednost `True` ako je specificirani opseg kompletno formatiran kao zacrtnjen (bold), odnosno `False` ako nijedan deo opsega nije zacrtnjen i vraća vrednost `wdUndefined` ako je samo deo opsega zacrtnjen. Ovoj osobini možete i zadavati vrednosti: `True` (za zacrtnjivanje), `False` (za uklanjanje zacrtnjenja) ili `wdToggle` (za promenu tekućeg podešenja i prebacivanje između `True` i `False`).

`Range.Case` Vraća ili postavlja tip veličine slova specificiranog opsega. Ova osobina koristi razne `wdCharacterCase` konstante, uključujući i `wdLowerCase` (za mala slova), `wdTitleCase` (za naslove), `wdTitleWord`, `wdToggleCase` i `wdUpperCase` (za velika slova).

`Range.Font` Vraća ili postavlja objekat `Font` koji specificira korišćeno formatiranje karaktera u opsegu.

`Range.Italic` Vraća vrednost `True` ako je specificirani opseg kompletno formatiran kao italik (iskošen), vraća vrednost `False` ako nijedan deo opsega nije iskošen i vraća vrednost `wdUndefined` ako je samo deo opsega formatiran kao iskošen. Ovoj osobini možete i da zadajete vrednosti korišćenjem `True` (za iskošena slova), `False` (za uklanjanje iskošenih slova) ili `wdToggle` (za promenu tekućeg podešenja i prebacivanje između `True` i `False`).

Na primer, sledeći deo koda uzima objekat `Range` prvog paragrafa aktivnog dokumenta, pa zatim postavlja `Bold` na `True`, `Italic` na `True` i `Case` na `wdTitleWord`:

```
With ActiveDocument.Paragraphs(1).Range
    .Bold = True
    .Italic = True
    .Case = wdTitleWord
End With
```

Ako želite maksimalnu kontrolu nad formatiranjem karaktera u opsegu, koristite osobinu `Font`, koja vraća objekat `Font`. Pomoću njega možete da manipulišete ne samo osobinama `Bold` i `Italic`, već i veličinom (osobina `Size`), bojom (`Color`), precrtavanjem (`StrikeThrough` i `DoubleStrikeThrough`), malim slovima (`SmallCaps`) i tako dalje. Evo primera:

```
With ActiveDocument.Range.Font
    .Color = RGB(0, 0, 255)
    .Size = 12
    .SmallCaps = True
End With
```

Neke korisne metode objekta Range

Pošto je fundamentalni tekstualni objekat, nije ništa iznenađujuće to što objekat Range ima veliki broj metoda koje možete da koristite za manipulisanje tekstem. Evo nekih od njih koje ćete najčešće koristiti:

Range.Copy Kopira opseg u klipbord.

Range.Cut Iseca opseg iz dokumenta i stavlja ga u klipbord.

Range.Delete Ako se koristi bez argumenata, ova metoda briše ceo opseg. Međutim, svoja brisanja možete fino podešavati korišćenjem sledeće sintakse:

Range.Delete(Unit, Count)

- | | |
|--------------|---|
| Range | Objekat Range koji sadrži tekst koji želite da brišete. |
| Unit | (opciono) Konstanta koja specificira da li ćete brisati karaktere (koristite <code>wdCharacter</code>) ili cele reči (koristite <code>wdWord</code>). Ako izostavite ovaj argument, VBA podrazumeva da brišete karaktere. |
| Count | (opciono) Broj jedinica koje treba izbrisati. Koristite pozitivan broj za prisanje nadesno; koristite negativan broj za brisanje ulevo. (Podrazumevana vrednost je 1.) |

Range.InsertAfter Ubacuje tekst iza specificiranog opsega:

Range.InsertAfter(Text)

- | | |
|--------------|---|
| Range | Objekat Range iza koga želite da ubacite tekst. |
| Text | Tekst koji se ubacuje. |

Range.Insert Before Ubacuje tekst ispred specificiranog opsega:

Range.InsertBefore(Tekst)

- | | |
|--------------|--|
| Range | Objekat Range ispred koga želite da ubacite tekst. |
| Text | Tekst koji se ubacuje. |

Range.InsertParagraph Ubacuje paragraf koji zamenjuje specificirani opseg.

Range.InsertParagraphAfter Ubacuje paragraf iza specificiranog opsega.

Range.InsertParagraphBefore Ubacuje paragraf ispred specificiranog opsega.

Range.Paste "Lepi" sadržaj klipborda na poziciju tekućeg objekta. Da biste izbegli prepisivanje preko trenutno selektovanog opsega, pre lepljenja koristite metodu `Collapse`.

Range.Select Selektuje specificirani opseg. Vidite sledeći odeljak u vezi objekta `Selection`.

Objekat Selection

Objekat Selection uvek se odnosi na dve stvari:

- Selektovani tekst
- Pozicija kursora

Pošto u Wordu uglavnom radite nešto od ove dve stvari (formatiranje teksta, ubacivanje teksta na mestu kursora i tako dalje), objekat Selection je jedan od najbitnijih objekata u Wordu. (Ovde malo pojednostavljujem stvari zato što objekat Selection može da se odnosi i na selektovani oblik ili blok. U ovom odeljku baviću se samo selektovanjem teksta.)

Trenutno selektovanom tekstu ili poziciji kursora obraćate se pomoću osobine Selection, bez kvalifikatora objekta. Na primer, sledeća naredba formatira selektovani tekst kao zacrtnjen:

```
Selection.Range.Bold = True
```

Da biste specifikovali objekat Selection, koristite metodu Select, koja postoji u mnogim Wordovim objektima, uključujući i Document, Range, Bookmark i Table. Na primer, sledećom naredbom selektuje se prvi paragraf u aktivnom dokumentu:

```
ActiveDocument.Paragraphs(1).Range.Select
```

Provera tipa selekcije

Objekat Selection ima veliki broj osobina, uključujući i neke koje ste već imali priliku da vidite kod objekata Document i Range. Nekoliko osobina koje su jedinstvene za objekat Selection nisu naročito korisne, tako da ih ovde nećemo diskutovati. Izuzetak je osobina Type:

Type - Vraća tip selekcije:

wdNoSelection	Ništa nije selektovano.
wdSelectionColumn	Selektovana je kolona u tabeli.
wdSelectionIP	Selektovana je pozicija kursora.
wdSelectionNormal	Selektovan je neki tekst.
wdSelectionRow	Selektovan je red u tabeli.

Metode objekta Selection

Ponovo, objekat Selection nudi mnoge metode koje ima i objekat Range, uključujući Copy, Cut, Delete, InsertAfter, InsertBefore, InsertParagraphAfter, InsertParagraphBefore i Paste. Pogledajmo neke metode koje su ili jedinstvene za objekat Selection ili se najčešće koriste sa ovim objektom:

Collapse Vrší deselektovanje i pozicionira kursor u skladu sa sledećom sintaksom:

Selection.Collapse.Direction

Direction (opcionij) Specificira mesto na kome želite da se premesti kursor. Koristite **wdCollapseStart** da biste pozicionirali kursor na početku selekcije (to se podrazumeva). Koristite **wdCollapseEnd** da biste postavili kursor na kraju selekcije.

EndKey Proširuje selekciju ili prebacuje kursor na kraj zadate jedinice (kao što je linija). Drugim rečima, ova metoda je ekvivalentna pritiskanju tastera **End**. Primetite, takođe, da ova metoda vraća broj karaktera za koji je selekcija pomerena. Evo njene sintakse:

Selection.EndKey(Unit, Extend)

Unit (opcionij) Specificira jedinicu na čiji kraj se pomera selekcija. Za regularni tekst koristite **wdLine** (podrazumevana vrednost) ili **wdStory**. U tabeli koristite **wdColumn** ili **wdRow**.

Extend (opcionij) Specificira šta se dešava selekciji. Da biste proširili selekciju do kraja specificirane jedinice (**Unit**) koristite **wdExtend**; za skupljanje selekcije na sam kursor, koristite **wdMove** (to je podrazumevana vrednost).

EndOf Proširuje selekciju ili pomera kursor na kraj specificirane jedinice (kao što je paragraf). Ova metoda vraća broj karaktera za koji je selekcija pomerena. Ova metoda nudi malo veću fleksibilnost od metode **EndKey**. Evo njene sintakse:

Selection.EndOf(Unit, Extend)

Unit (opcionij) Specificira jedinicu za koju se selekcija pomera. Za regularni tekst koristite **wdCharacter**, **wdWord** (ovo je podrazumevana vrednost), **wdLine**, **wdSentence**, **wdParagraph**, **wdSection** ili **wdStory**. U tabeli koristite **wdCell**, **wdColumn**, **wdRow** ili **wdTable**.

Extend (opcionij) Specificira šta se dešava selekciji. Da biste proširili selekciju do kraja specificirane jedinice (**Unit**) koristite **wdExtend**; za skupljanje selekcije na sam kursor, koristite **wdMove** (to je podrazumevana vrednost).

Expand Proširuje selekciju korišćenjem sledeće sintakse:

Selection.Expand Unit

Unit (opciono) Specificira način na koji se selekcija proširuje. Za regularni tekst koristite `wdCharacter`, `wdWord` (ovo je podrazumevana vrednost), `wdLine`, `wdSentence`, `wdParagraph`, `wdSection` ili `wdStory`. U tabeli koristite `wdCell`, `wdColumn`, `wdRow` ili `wdTable`.

Na primer, sledeći kod selektuje prvu reč u prvom paragrafu. Pošto je objekat `Selection` definisan, kod zatim pretvara slova prve reči u velika slova, proširuje selekciju na ceo paragraf, formatira novu selekciju tako što je zacrnuje, a zatim vrši deselektovanje tako da se kursor nađe iza datog paragrafa:

```
ActiveDocument.Paragraphs(1).Range.Words(1).Select
With Selection
    .Range.Case = wdUpperCase
    .Expand wdParagraph
    .Range.Bold = True
    .Collapse wdCollapseEnd
End With
```

HomeKey Proširuje selekciju ili pomera kursor na početak zadate jedinice (kao što je linija). Ova metoda je ekvivalentna pritiskanju tastera Home. Kao i kod metode **EndKey**, **HomeKey** vraća broj karaktera za koji je selekcija pomeren. Koristite sledeću sintaksu:

```
Selection.Home(Unit, Extend)
```

Unit (opciono) Specificira jedinicu na čiji početak se pomera selekcija. Za regularni tekst koristite `wdLine` (podrazumevana vrednost) ili `wdStory`. U tabeli koristite `wdColumn` ili `wdRow`.

Extend (opciono) Specificira šta se dešava selekciji. Da biste proširili selekciju do početka specificirane jedinice (**Unit**) koristite `wdExtend`; za skupljanje selekcije na sam kursor, koristite `wdMove` (to je podrazumevana vrednost).

Move Sažima tekuću selekciju i pomera kursor za zadati broj jedinica. Ova metoda vraća broj jedinica za koji je pomeren kursor. Evo sintakse:

```
Selection.Move(Unit, Count)
```

Unit (opciono) Specificira jedinicu za koju se selekcija pomera. Za regularni tekst koristite `wdCharacter` (ovo je podrazumevana vrednost), `wdWord`, `wdLine`, `wdSentence`, `wdParagraph`, `wdSection` ili `wdStory`. U tabeli koristite `wdCell`, `wdColumn`, `wdRow` ili `wdTable`.

Count (opcion) Broj jedinica za koji se kursor pomera (podrazumevana vrednost je 1).

StartOf Proširuje selekciju ili pomera kursor na početak zadate jedinice (kao što je paragraf). Ova metoda vraća broj karaktera za koji je selekcija pomeren. Ova metoda nudi nešto veću fleksibilnost od metode `HomeKey`. Evo njene sintakse:

```
Selection.StartOf(Unit, Extend).
```

Unit (opcion) Specificira jedinicu za koju se selekcija pomera. Za regularni tekst koristite `wdCharacter`, `wdWord` (ovo je podrazumevana vrednost), `wdLine`, `wdSentence`, `wdParagraph`, `wdSection` ili `wdStory`. U tabeli koristite `wdCell`, `wdColumn`, `wdRow` ili `wdTable`.

Extend (opcion) Specificira šta se dešava selekciji. Da biste proširili selekciju do početka specificirane jedinice (`Unit`) koristite `wdExtend`; za skupljanje selekcije na sam kursor koji se postavlja na početak jedinice, koristite `wdMove` (to je podrazumevana vrednost).

TypeBackspace Briše karakter sa leve strane kursora. Ako trenutna selekcija nije skupljena, ova metoda briše selektovani tekst. (Ova metoda je ekvivalentna pritiskanju tastera `Backspace`.)

TypeParagraph Ubacuje novi paragraf na poziciji kursora. Ako trenutna selekcija nije skupljena, ova metoda briše selektovani tekst. (Ova metoda je ekvivalentna pritiskanju tastera `Enter`.)

TypeText Ubacuje tekst na poziciji kursora. Ako trenutna selekcija nije skupljena i ako je osobina `Options.ReplaceSelection` postavljena na vrednost `True`, ova metoda zamenjuje selekciju specificiranim tekstom. Evo sintakse:

```
Selection.TypeText Text
```

Text String koji se ubacuje.

N A P O M E N A

Ako ne želite da se trenutna selekcija briše kada ubacujete novi paragraf, koristite metode `InsertParagraphAfter` ili `InsertParagraphBefore`. Pogledajte odeljak "Neke korisne metode objekta `Range`" iz ovog poglavlja.

Listing 7.4 pokazuje proceduru koja ubacuje tekst. Ona prvo proverava da li je trenutna selekcija kursor ili da li osobina `Options.ReplaceSelection` ima vrednost `False`. Ako nešto od toga važi, tekst se ubacuje. U suprotnom, korisniku se postavlja pitanje da li želi da zameni selektovni tekst. (Vidite Poglavlje 12, "Interakcija sa korisnikom", da biste naučili više o funkciji `MsgBox`.)

Listing 7.4: Procedura koja pita korisnika da li selektovani tekst treba da se zameni

```
SubPitajZaZamenuTeksta()  
    Dim rezultat As Integer  
    '  
    'Da li smo na poziciji kursora ili da li je mogucnost zamene selekcije iskljucena?  
    '  
    If Selection.Type = wdSelectionIP Or _  
        Not Options.ReplaceSelection Then  
        '  
        'Ako jeste, samo ubaciti tekst  
        '  
        Selection.TypeText "Zdravo"  
    Else  
        '  
        'U suprotnom, pitati korisnika da li treba zameniti tekst  
        '  
        rezultat = MsgBox("Da li zameniti selektovani tekst?" vbYesNo + vbQuestion)  
        If rezultat = vbNo Then  
            '  
            'Iskljuciti ReplaceSelection da bi se ubacivanje izvrsilo bez zamene  
            Options.ReplaceSelection = False  
            Selection.TypeText "Zdravo"  
            Options.ReplaceSelection = True  
        Else  
            '  
            'Zameniti selektovani tekst  
            '  
            Selection.TypeText "Zdravo"  
        End If  
    End If  
End Sub
```

Objekat Characters

Objekat Characters je zbirka koja predstavlja sve karaktere u nekom zadatom objektu. Na primer, `ActiveDocument.Paragraphs(1).Range.Characters` je zbirka svih karaktera u objektu Range koji je specificiran sa `ActiveDocument.Paragraphs(1).Range` (prvi paragraf u aktivnom dokumentu). Ostali objekti koji imaju osobinu Characters su Document i Selection

Pošto objekat Characters predstavlja zbirku, pojedinačnim karakterima se obraćate tako što ćete uključiti indeksni broj - na primer, Characters(50). Sledeća naredba formatira prvi karakter u aktivnom dokumentu na veličinu od 20 tačaka:

```
ActiveDocument.Characters(1).Font.Size = 20
```

Za brojanje karaktera u zadatom objektu, koristite osobinu Count:

```
ukupanBrojKaraktera = Documents("Poglavlje07.doc").Characters.Count
```

U ovom primeru se definiše da je promenljiva ukupanBrojKaraktera jednaka broju karaktera u datoteci Poglavlje07.doc.

Listing 7.5 pokazuje drugi primer koji koristi objekat Characters. U ovom slučaju, procedura funkcije BrojanjeKaraktera kao argument uzima argument Object, koji je nazvan brojObjekt, i argument String, nazvan pismo. Procedura određuje koliko puta se niz karaktera pismo pojavljuje u objektu brojObjekt.

Listing 7.5: Funkcija koja broji koliko puta se zadati karakteri pojavljuju u objektu

```
Function BrojanjeKaraktera (brojObjekt As Object, pismo As String) As Long
    Dim i As Long, char As Range
    i = 0
    For Each char in brojObjekt.Characters
        If char = pismo Then i = i + 1
    Next 'char
    BrojanjeKaraktera = i
End Function

Sub ProveraBrojaKaraktera()
    MsgBox BrojanjeKaraktera(Active.Document, "e")
End Sub
```

Objekat Words

Objekat Words je zbirka koja predstavlja sve reči u nekom zadatom objektu. Na primer, ActiveDocument.Words je zbirka svih reči u aktivnom dokumentu. Ostali objekti koji imaju osobinu Words su Paragraph, Range i Selection.

Pojedinim rečima se obraćamo pomoću indeksnog broja u zbirci Words. Međutim, kao što sam i ranije pomenuo, ne vraća se "Word" objekat; tako nešto ne postoji u svetu Microsoft Wordovog VBA. Umesto toga, pojedinačne reči se klasifikuju kao Range objekti (vidi naslov "Objekat Range" iz ovog poglavlja).

Sledeća naredba formatira prvu reč u aktivnom dokumentu tako da bude zacrnjena:

```
ActiveDocument.Words(1).Font.Bold = True
```

Za brojanje reči u zadatom objektu, koristite osobinu Count:

```
ukupanBrojReci = Documents("Clanak.doc").Words.Count
```

Međutim, primetite da objekat Word obuhvata i znakove interpunkcije i oznake paragrafa u okviru objekta, što je sigurno bizarno ponašanje i služi za manje ili više beskorisno renderovanje osobine Word.Count. Ako želite da znate broj pravih reči u objektu, koristite funkciju BrojanjeReci iz listinga 7.6.

Listing 7.6: Funkcija koja broji "prave" reči u objektu, ignorišući znakove interpunkcije i oznake paragrafa

```
Function BrojanjeReci(brojObjekat As Object) As Long
    Dim i As Long, word As Range
    i = 0
    For Each word in brojObjekat.Words
        Select Case Asc(Left(word, 1))
            Case 48 To 57, 65 To 90, 97 To 122
                i = i + 1
        End Select
    Next 'word
    BrojanjeReci = i
End Function

Sub ProveraBrojaReci()
    With ActiveDocument
        MsgBox "Words.Count izvestava " & .Words.Count & Chr(13) & _
            "BrojanjeReci izvestava " & BrojanjeReci(.Range)
    End With
End Sub
```

Ova funkcija uzima generički objekat kao argument (zato što može da radi sa objektom Document, Range ili Selection). Zatim koristi petlju For Each da bi prošla kroz svaku reč u objektu. Pri svakom prolasku kroz petlju se ASCII vrednost poslednjeg (levog) karaktera ispituje u okviru naredbe Select Case. Ako je ta vrednost između 48 i 57, 65 i 90 ili između 97 i 122, to znači da je karakter ili broj ili slovo. Ako je to slučaj, funkcija broji reč kao pravu reč i brojač i uvećava za 1.

Objekat Sentences

Sledeći rang na lestvici Wordovih tekstualnih objekata je objekat **Sentences**. Ovo je zbirka svih rečenica u bilo kom objektu koji zadate, bez obzira na to da li se radi o objektu **Document**, **Range** ili **Selection**.

Kao i kod zbirke **Words**, i ovde se pojedinačnim članovima zbirke **Sentences** obraćate pomoću indeksnog broja, a rezultujući objekat je **Range**. Na primer, sledeća naredba vraća prvu rečenicu aktivnog dokumenta preko promenljive **prvaRecenica**:

```
prvaRecenica = ActiveDocument.Sentences(1)
```

Ponovo, osobina **Count** se može koristiti za prebrojavanje ukupnog broja rečenica u objektu. U narednom delu koda, osobina **Count** se koristi za određivanje poslednje rečenice u dokumentu:

```
With Documents("Napomene.doc")  
    poslednjaRecenica = .Sentences(.totalSentences)  
End With
```

Objekat Paragraph

Od karaktera, reči i rečenica pravi se sledeća logična tekstualna jedinica: paragraf. Objekat **Paragraphs** je član zbirke **Paragraphs**, koja predstavlja sve paragrafe u zadatom objektu **Document**, **Range** ili **Selection**. Kao i kod drugih tekstualnih objekata, indeksni broj se koristi sa objektom **Paragraphs** da bi se specificirao pojedinačni paragraf.

Neke korisne osobine objekta Paragraph

Razne Wordove opcije formatiranja paragrafa su dobro predstavljene velikim skupom osobina koje ima objekat **Paragraph**. Evo nekih korisnih:

- | | |
|------------------------------|--|
| Paragraph.LeftIndent | Vraća ili postavlja levo uvlačenje teksta (u tačkama) specificiranog paragrafa. |
| Paragraph.LineSpacing | Vraća ili postavlja razmak između linija (u tačkama) specificiranog paragrafa. |
| Paragraph.RightIndent | Vraća ili postavlja desno uvlačenje teksta (u tačkama) specificiranog paragrafa. |

`Paragraph.SpaceAfter` Vraća ili postavlja razmak (u tačkama) iza specificiranog paragrafa.

`Paragraph.SpaceBefore` Vraća ili postavlja razmak (u tačkama) izpred specificiranog paragrafa.

`Paragraph.Style` Vraća ili postavlja stil specificiranog paragrafa. Word ima ogroman broj konstanti koje predstavljaju njegove unapred definisane stilove. Na primer, da biste podesili stil `Heading1` (naslov 1), koristili biste konstantu `wdStyleHeading1`. Ostale konstante možete videti ako u Object Browseru potražite `wdBuiltInStyle`.

Sledeći deo koda primenjuje nekoliko osobina na aktivan paragraf (setite se da funkcija `InchesToPoints` konvertuje vrednosti izražene u inčima u ekvivalentne vrednosti izražene u tačkama):

```
With Selection.Range
    .LeftIndent = InchesToPoints(1)
    .LineSpacing = 12
    .SpaceAfter = 6
    .Style = wdStyleNormal
End With
```

Neke korisne metode objekta Paragraph

Za kraj pregleda objekta `Paragraph`, evo nekoliko metoda koje možete da ugradite u svoj kod:

`Paragraph.Indent` Uvlači zadati paragraf do sledećeg markera tabulatora.

`Paragraph.Outdent` Izvlači zadati paragraf do prethodnog markera tabulatora.

`Paragraph.Space1` Postavlja jednolinijski razmak kod zadatog paragrafa.

`Paragraph.Space15` Postavlja razmak od 1.5 linije kod zadatog paragrafa.

`Paragraph.Space2` Postavlja razmak od 2 linije kod zadatog paragrafa.

APSOLUTNI MINIMUM

Ovo poglavlje vas je provelo kroz VBA u Wordu. Naučili ste više o Wordovom objektu `Application`, uključujući i nekoliko korisnih osobina i metoda. Odatle smo prošli kroz nekoliko objekata specifičnih za Word, kao što su `Document`, `Range`, `Selection`, `Characters`, `Words`, `Sentences` i `Paragraph`.

Evo spiska poglavlja u kojima ćete naći srodne informacije:

- Opšta diskusija VBA objekata nalazi se u Poglavlju 5, "Rad sa objektima".
 - Pokazaću vam kako da koristite `MsgBox` i ostale interaktivne funkcije u Poglavlju 12, "Interakcija sa korisnikom".
 - Procedura `NapraviRezervnuKopiju` bi verovatno trebalo da proverí da li postoji disketa u disketnoj jedinici (drajv A) da bi se izvršila metoda `SaveAs`. Pokazaću vam kako da uračunate i takvu vrstu greške u Poglavlju 15, "Testiranje VBA procedura".
-