



Programiranje kroz aplikacije

Aktivni objekti

For Each i With

Programiranje događaja

Aktivni objekti

- U Excel-u, samo jedna radna sveska može biti trenutno aktivna. Slično, kod aktivne sveske, samo jedan radni list može biti trenutno aktivan i samo jedna ćelija je aktivna ćelija, čak i kada se radi sa opsegom koji obuhvata više ćelija. Uzimajući ovo u obzir, VBA nam dopušta da jednostavno pristupamo aktivnim objektima.
- Aktivnim objektima pristupamo preko osobina objekta `Application`. Na primer, objekat `Application` ima osobinu `ActiveCell` koja vraća referencu na aktivnu ćeliju. Tako, na primer, ako želimo da u aktivnu ćeliju upišemo broj 32.1, dovoljno je napisati
`ActiveCell.Value = 32.1`
- Objekat `Application` je izostavljen jer se podrazumeva.
- Ukoliko je selektovan opseg unutar radnog lista, aktivna ćelija će biti jedna ćelija tog opsega.

Aktivni objekti (nastavak)

- Drugi aktivni objekti su:
 - **Selection** (selektovani objekat, bila to jedna ćelija, opseg ćelija, dijagram, tekst boks ili oblik),
 - **ActiveChart** (aktivni dijagram ili dijagram na radnom listu),
 - **ActiveWindow** (aktivni prozor),
 - **ActiveWorkbook** (aktivna radna sveska),
- **ThisWorkbook** predstavlja radnu svesku koja sadrži proceduru koja se trenutno izvršava – može biti aktivni objekat, ali i ne mora.
- Pogodnost rada sa ovim osobinama je da nema potrebe znati aktivnu ćeliju, radni list ili svesku, što omogućava pisanje VBA kôda koji nije vezan ni za jednu posebnu radnu svesku, list ili opseg.

Konstrukcija With

- Posmatrajmo sledeći niz operacija na jednom objektu (opseg **A1:D5**):

```
Range("A1:D5").Interior.Color = RGB(59, 123, 19)
```

```
Range("A1:D5").WrapText = True
```

```
Range("A1:D5").Font.Name = "Courier New"
```

```
Range("A1:D5").Font.Bold = True
```

```
Range("A1:D5").Font.Size = 13
```

```
Range("A1:D5").Font.Underline = xlUnderlineStyleSingle
```

```
Range("A1:D5").Font.ColorIndex = 34
```

- Ove naredbe se mogu grupisati na sledeći način:

```
With Range("A1:D5")
```

```
    .Interior.Color = RGB(59, 123, 19)
```

```
    .WrapText = True
```

```
    .Font.Name = "Courier New"
```

```
    .Font.Bold = True
```

```
    .Font.Size = 13
```

```
    .Font.Underline = xlUnderlineStyleSingle
```

```
    .Font.ColorIndex = 34
```

```
End With
```

Konstrukcija With (nastavak)

- Prvi set naredbi je jasniji, ali se drugi set naredbi izvršava brže, jer se objekat ne referencira eksplicitno u svakoj naredbi.
- Pošto se u drugom setu naredbi ponavlja objekat **Font**, može se ugnezditi jedna **With** naredba radi grupisanja osobine ovog objekta:

```
With Range("A1:D5")
```

```
    .Interior.Color = RGB(59, 123, 19)
```

```
    .WrapText = True
```

```
    With .Font
```

```
        .Name = "Courier New"
```

```
        .Bold = True
```

```
        .Size = 13
```

```
        .Underline = xlUnderlineStyleSingle
```

```
        .ColorIndex = 34
```

```
    End With
```

```
End With
```

Definisanje boje

- U prethodnom primeru smo boju fonta odredili pomoću funkcije **RGB**, koja se poziva na sledeći način:

RGB(crvena, zelena, plava)

- Obavezni argumenti, **crvena**, **zelena** i **plava**, su celi brojevi od 0 do 255. Na ovaj način se može definisati 256^3 boja.
- Drugi način da se definiše boja u VBA je pomoću osobine **ColorIndex**, koja može uzeti vrednosti od 0 (bez boje) do 56.
- Jedini način da se ukloni boja objekta je pomoću osobine **ColorIndex** upisivanjem vrednosti **0** ili **xlColorIndexNone**.

For Each petlja

- **For Each** petlja predstavlja varijaciju **For** petlje i namenjena je radu sa kolekcijama, tj. kada je potrebno izvršiti određenu operaciju nad svim objektima u datoj kolekciji. Sintaksa **For Each** petlje je:

For Each Element in Grupa

Naredbe

Next Element

Grupa predstavlja ime kolekcije i to može biti i niz.

- Iz **For Each** petlje se takođe izlazi koristeći **Exit For**. Umesto **Next Element** možemo koristiti samo **Next**.
- Štampanje elemenata niza **Niz** se može izvršiti na sledeći način:

For Each n In Niz

Debug.Print n

Next n

- Na ovaj način ne moramo znati broj elemenata niza.

For Each petlja: Primeri

- Prikazati imena svih radnih listova aktivne radne sveske.

```
Dim List As Worksheet  
For Each List In ActiveWorkbook.Worksheets  
    MsgBox List.Name  
Next List
```

Osobina **Name** vraća ime radnog lista.

- Svakoj ćeliji selekcije dodeliti slučajan ceo broj između 0 i 100.

```
Dim Cell As Range  
For Each Cell In Selection  
    Cell.Value = Fix(Rnd * 101)  
Next
```

- Zatvoriti sve radne sveske osim aktivne.

```
Dim Sveska as Workbook  
For Each Sveska In Workbooks  
    If Sveska.Name <> ActiveWorkbook.Name Then Sveska.Close  
Next Sveska
```

Metoda **Close** zatvara radnu svesku.

Objekti Workbook / Workbooks

Osobina	Opis
Objekat Workbook	
FullName	Vraća ime radne sveske, uključujući i put do nje.
Name	Vraća ime radne sveske.
Path	Vraća put do radne sveske. Kod novih, nesnimljenih, radnih sveski Path je prazan string.
Kolekcija Workbooks	
Count	Vraća broj trenutno otvorenih radnih sveski.
Metoda	Opis
Objekat Workbook	
Activate	Aktiviranje radne sveske. Ta sveska postaje ActiveWorkbook .
Close	Zatvaranje radne sveske.
PrintOut	Štampanje specificirane radne sveske.
Save	Snimanje radne sveske. U slučaju nove radne sveske, koristiti SaveAs .
SaveAs	Snimanje specificirane radne sveske pod drugim imenom.
Kolekcija Workbooks	
Add	Kreiranje nove radne sveske.
Open	Otvaranje postojeće radne sveske.

Objekti Worksheet / Worksheets

Osobina	Opis
Objekat Worksheet	
Name	Vraća ili menja ime radnog lista.
Visible	Određuje da li se dati radni list može videti (True) ili ne (False).
Rows	Vraća kolekciju vrsta datog radnog lista (objekat Range).
Columns	Vraća kolekciju kolona datog radnog lista (objekat Range).
Kolekcija Worksheets	
Count	Vraća broj radnih listova sveske.

Metoda	Opis
Objekat Worksheet	
Activate	Aktiviranje specificiranog radnog lista. Taj list postaje ActiveSheet .
Select	Selektovanje specificiranog radnog lista.
Calculate	Izvršavanje specificiranog radnog lista.
Copy	Kopiranje specificiranog radnog lista.
Move	Pomeranje specificiranog radnog lista.
Delete	Brisanje specificiranog radnog lista.
Kolekcija Worksheets	
Add	Dodavanje novih radnih listova u datu svesku.

Range objekat

Osobina	Opis
Address	Vraća adresu specificiranog opsega (u vidu teksta).
Rows	Vraća kolekciju vrsta datog opsega (objekat Range).
Columns	Vraća kolekciju kolona datog opsega (objekat Range).
Count	Vraća broj ćelija specificiranog opsega.
CurrentRegion	Vraća Range objekat koji predstavlja region u kom se nalazi predmetni opseg.
Resize	Menja veličinu datog opsega i vraća referencu na dobijeni opseg.
Worksheet	Vraća Worksheet objekat koji sadrži predmetni opseg.

Metoda	Opis
Cut	Premeštanje opsega na Clipboard ili novu destinaciju.
Copy	Kopiranje opsega na Clipboard ili novu destinaciju.
Clear	Brisanje kompletnog sadržaja opsega, uključujući format i komentare.
ClearContents	Brisanje sadržaja iz opsega, uz očuvanje formata.
Delete	Brisanje specificiranog opsega.
Select	Selektovanje specificiranog opsega.
AddComment	Dodavanje komentara u specificiranu ćeliju.

Primer

- Napisati proceduru ABC koja kreira novu radnu svesku, daje imena listovima List1, List2,..., upisuje string unesen pomoću input box-a u ćeliju A1 svakog lista, snima svesku pod imenom **Sveska.xlsx** i zatvara je.

```
Sub ABC()  
    Dim Sveska As Workbook, List As Worksheet, S As String, I As Integer  
    Set Sveska = Workbooks.Add  
    S = InputBox("Uneti string:")  
    For Each List In Sveska.Worksheets  
        I = I + 1  
        List.Name = "List" & I  
        List.Range("A1").Value = S  
    Next  
    Sveska.SaveAs ThisWorkbook.Path & "\Sveska.xlsx"  
    Sveska.Close  
End Sub
```

Programiranje događaja

- Događaj predstavlja akciju koju inicira bilo korisnik bilo VBA kôd.
- Programiranje događaja je moćan alat pomoću kojeg pratimo korisničke akcije ili promene same aplikacije i preduzimamo odgovarajuće korake.
- Procedure za upravljanje događajima (eng. *event-handler procedures*) su subprocedure koje programiramo, u skladu sa samim događajem, i koje Excel automatski izvršava kad se događaj pojavi.
- Događaji i procedure za upravljanje događajima postoje od Excel-a 97. Ranije verzije nisu podržavale rad sa događajima. Od verzije 97 do 2013, celokupna struktura događaja je ostala nepromenjena, uz dodatak manjeg broja događaja zavisno od same verzije.

Događaji u Excel-u

- Događaje u Excel-u možemo klasifikovati na:
 - događaje vezane za radne sveske,
 - događaje vezane za radne listove,
 - događaje vezane za dijagrame,
 - događaje vezane za aplikaciju,
 - događaje vezane za korisničke forme,
 - događaje vezane za klase i
 - događaje koji nisu vezani za objekte.
- Procedure za upravljanje događajima moraju biti smeštene u okviru odgovarajućih modula. Na primer, procedure koje upravljaju događajima vezanim za radne sveske se smeštaju u okviru [ThisWorkbook](#) modula, procedure koje upravljaju događajima vezanim za radne listove se smeštaju u okviru modula radnih listova (npr. [Sheet1](#), [Sheet2](#)).

Događaji na nivou radne sveske

Događaj	Akcija koja okida događaj
Activate	Radna sveska je aktivirana.
BeforeClose	Radna sveska treba da se zatvori.
BeforePrint	Radna sveska treba da se štampa ili pregleda sa Print Preview.
BeforeSave	Radna sveska treba da se snimi.
Deactivate	Radna sveska je deaktivirana.
NewSheet	Novi list je dodat u radnu svesku.
Open	Radna sveska je otvorena.
SheetActivate	Bilo koji radni list je aktiviran.
SheetBeforeDoubleClick	Dvoklik na bilo kojem radnom listu.
SheetBeforeRightClick	Desni klik na bilo kojem radnom listu.
SheetCalculate	Bilo koja radna sveska je proračunata.
SheetChange	Bilo koja radna sveska je promenjena.
SheetDeactivate	Bilo koja radna sveska je deaktivirana.
SheetSelectionChange	Selekcija na bilo kojoj radnoj svesci je promenjena.
WindowActivate	Bilo koji prozor radne sveske je aktiviran.
WindowDeactivate	Bilo koji prozor radne sveske je deaktiviran.
WindowResize	Veličina bilo kojeg prozora radne sveske je promenjena.

Primer – Evidencija otvaranja sveske

- Uradimo primer sa vođenjem evidencije o otvaranju radne sveske. U radni list, nazvan **Evidencija**, ćemo upisivati vreme i datum svakog otvaranja sveske.
- Upisivanje vršimo u ćelijama A1, A2 itd. Ukoliko radni list Evidencija ne postoji, kreiraćemo ga pri prvom otvaranju sveske.
- Ukoliko sakrijemo list Evidencija (desni klik na ime lista u donjem levom uglu ili dugme **Hide** na grupi komandi **Window** kartice **View**), možemo neprimetno voditi evidenciju otvaranja radne sveske.


```
Private Sub Workbook_Open()
```

```
Dim Sveska As Workbook, List As Worksheet
```

```
Dim Postoji As Boolean, BrList As Integer, I As Integer
```

```
Set Sveska = ThisWorkbook
```

```
Postoji = False
```

```
For Each List In Sveska.Worksheets
```

```
    If List.Name = "Evidencija" Then Postoji = True
```

For petlja utvrđuje da li postoji radni list sa imenom Evidencija

```
Next
```

```
If Postoji = False Then
```

Ako ne postoji list Evidencija, kreiraj ga nakon svih listova

```
    BrList = Sveska.Worksheets.Count
```

```
    Sveska.Worksheets.Add After:=Sveska.Worksheets(BrList)
```

```
    Sveska.Worksheets(BrList + 1).Name = "Evidencija"
```

```
End If
```

```
Set List = Sveska.Worksheets("Evidencija")
```

```
I = 1
```

```
Do While List.Cells(I, 1).Value <> ""
```

```
    I = I + 1
```

Pronađi prvu praznu ćeliju u prvoj koloni gde se vrši upis

```
Loop
```

```
List.Cells(I, 1).Value = "Otvorena " & Date & " u " & Time
```

```
Sveska.Worksheets(1).Select
```

```
Sveska.Save
```

```
End Sub
```

Događaj NewSheet (radna sveska)

- Ovaj događaj se okida dodavanjem novog lista u predmetnu radnu svesku. List može biti radni list ili dijagram (eng. *chart*). Mi ćemo raditi samo sa radnim listovima.
- Procedura koja odgovara ovom događaju je `Workbook_NewSheet`.
- Programirati događaj `NewSheet` tako da se pri kreiranju novog lista prikazuje input box u koji korisnik unosi ime radnog lista. Uneseni radni list se smešta nakon svih postojećih listova.

Kreirani list je prosleđen kao argument!!!

```
Private Sub Workbook_NewSheet(ByVal Sh As Object)
    Dim ime As String, L As Worksheet, brojListova as Integer
    ime = InputBox("Uneti ime: ", "Ime radnog lista")
    brojListova = ThisWorkbook.Worksheets.Count
    Sh.Name = ime
    Sh.Move After:=ThisWorkbook.Worksheets(brojListova)
End Sub
```

Događaj BeforePrint (radna sveska)

- Dešava se neposredno pred štampu ili pregled pred štampu (Print Preview). Ne postoji način da se odredi o kom zahtevu se radi.
- Događaj koristi argument **Cancel**, kojim se može poništiti štampa radne sveske. To se postiže postavljanjem vrednosti **Cancel** na True.
- Ovaj događaj postoji samo na nivou radne sveske, tj. ne postoji na nivou radnog lista, pa se ne može odrediti šta da se štampa.
- Programirati događaj **BeforePrint** tako da se javlja poruka (potrošen toner) korisniku pri svakom zahtevu za štampu date radne sveske.

```
Private Sub Workbook_BeforePrint(Cancel As Boolean)
    Dim Poruka As String, Odgovor As Integer
    Poruka = "Ponestalo je tonera!" & vbCrLf & _
        "Želite li da nastavite sa štampom?"
    Odgovor = MsgBox(Poruka, vbYesNo, "Podsećanje na toner")
    If Odgovor = vbNo Then Cancel = True
End Sub
```

Događaji na nivou radnog lista

Događaj	Akcija koja okida događaj
<code>Activate</code>	Radni list je aktiviran.
<code>BeforeDoubleClick</code>	Dvoklik na radnom listu.
<code>BeforeRightClick</code>	Desni klik na radnom listu.
<code>Calculate</code>	Radni list je proračunat ili nanovo proračunat.
<code>Change</code>	Ćelija radnog lista je promenjena.
<code>Deactivate</code>	Radni list je deaktiviran.
<code>SelectionChange</code>	Selekcija na radnom listu je promenjena.

- Procedure za upravljanje ovim događajima se smeštaju u kodni modul odgovarajućeg lista.

Događaj BeforeRightClick (radni list)

- Ovaj događaj se dešava pri operaciji desni klik, i to neposredno pre podrazumevane radnje inicirane desnim klikom. Podrazumevana radnja inicirana desnim klikom je pojava padajućeg menija.
- Odgovarajuća procedura ima dva argumenta, **Target** i **Cancel**. **Target** označava tekuću selekciju na radnom listu ili, ako ništa nije selektovano, ćeliju na koju je korisnik kliknuo, a **Cancel** određuje da li će se izvršiti podrazumevana radnja.
- Programirati događaj **BeforeRightClick** tako da se prikaže message box sa porukom koliko selektovani opseg ima ćelija, koliko praznih ćelija, koliko pozitivnih i negativnih brojeva, i koliko nula. Onemogućiti pojavu padajućeg menija.

```

Private Sub Worksheet_BeforeRightClick(ByVal Target As Range, Cancel As Boolean)
    Dim Poz As Integer, Neg As Integer, Nule As Integer
    Dim Prazne As Integer, S As String, Celija As Range
    Prazne = 0: Poz = 0: Neg = 0: Nule = 0
    For Each Celija In Target.Cells
        If Celija.Value = "" Then
            Prazne = Prazne + 1
        ElseIf IsNumeric(Celija.Value) Then
            If Celija.Value > 0 Then
                Poz = Poz + 1
            ElseIf Celija.Value < 0 Then
                Neg = Neg + 1
            Else
                Nule = Nule + 1
            End If
        End If
    Next
    S = "Ima " & Target.Count & " ćelija, od toga:" & vbCrLf
    S = S & Prazne & " praznih ćelija," & vbCrLf
    S = S & Poz & " pozitivnih brojeva," & vbCrLf
    S = S & Neg & " negativnih brojeva i " & vbCrLf & Nule & " nula."
    MsgBox S
    Cancel = True
End Sub

```

Prolazak kroz opseg i uvećavanje odgovarajućih brojača na osnovu sadržaja ćelije. Funkcija **IsNumeric** vraća True ako je sadržaj ćelije broj i False u suprotnom.

Formiranje stringa za prikaz pomoću MsgBox-a

Onemogućavanje pojave padajućeg menija

Događaji koji nisu pridruženi objektu

- Postoje i događaji koji nisu pridruženi objektima, kao što su `OnTime` i `OnKey`. Ovim se događajima pristupa koristeći istoimene metode objekta `Application`.
- Događaj `OnTime` startuje određenu proceduru u tačno određeno vreme, pri čemu procedura i vreme izvršenja predstavljaju argumente metode. Na primer, pozivom `Application.OnTime Now + TimeValue("00:00:05"), "UradiNesto"` postićemo da se naša procedura `UradiNesto` izvrši tačno 5 sekundi nakon pozivanja metode `OnTime`.
- Ova naredba se može naći u bilo kojoj proceduri projekta, pa sami biramo gde ćemo je staviti. Procedura `UradiNesto` se obično stavlja u standardni kodni modul.
- Primer sa `OnTime` u `Workbook_Open` proceduri.

Događaj OnKey

- Događaj `OnKey` startuje određenu proceduru pritiskom određene kombinacije tastera, pri čemu procedura i kombinacija tastera predstavljaju argumente metode. Na primer, pozivom `Application.OnKey "+{ESC}", "UradiNesto"` postizemo da se procedura `UradiNesto` izvrši pritiskom kombinacije tastera `Shift` i `Esc`. Pogledati Help radi više informacija o načinu zadavanja kombinacije tastera.
- Događaj i proceduru stavljamo u modul po želji.