Programiranje kroz aplikacije

Subprocedure

Opseg promenljivih

Excel VBA objektni model

Subprocedure

- Subprocedure (makroi) predstavljaju skup VBA naredbi kojima se izvršava određeni zadatak.
- Subprocedure mogu uticati na svoje okruženje (npr. mogu menjati format ćelija u Excel-u) i ne vraćaju rezultat!
- Format subprocedura je:

```
Sub ImeProcedure(arg1 As Tip, arg2 As Tip, ...)
        VBA naredbe
End Sub
```

- Ista pravila važe za ime subprocedure kao za ime funkcije (max 255 karaktera, prvi karakter slovo, a ostali karakteri slova, cifre ili _).
- Kao funkcija, subprocedura može imati proizvoljan broj ulaznih argumenata.
- Izvršavanje subprocedure se može prekinuti sa Exit Sub.

Pozivanje subprocedura

- U programskom kôdu, procedura se može pozvati na bilo koji od sledeća dva načina:

  Call ImeProcedure(arg1, arg2, ...)

  ImeProcedure arg1, arg2, ...

- Kad subprocedura nema argumente, može se izvršiti na još nekoliko načina:
  - Iz dokumenta (Word dokumenta, radne sveske, itd.), tj. bez ulaženja u VBE, vrši se odabirom opcije Tools / Macro / Macros (Office 2000/2003), odnosno sa Developer taba (Office 2007/2010/2013).
  - U VBE, u Immediate prozoru, pozivom imena subprocedure.
  - U VBE, pomoću Debug palete alatki. Pozicioniramo kursor u kôd subprocedure i pritisnemo dugme Run Sub / User Form ili tipku F5.
  - Sa palete alatki, pomoću odgovarajućeg dugmeta.

Primer subprocedure

- Napisati proceduru koja korisniku javlja imena svih radnih listova aktivne radne sveske.

```
Sub ImenaListova()
    Dim Imena As String, I As Integer
    Imena = ""
    For I = 1 To ActiveWorkbook.Worksheets.Count
       Imena = Imena & ActiveWorkbook.Worksheets(I).Name & vbCrLf
    Next
    MsgBox Imena
End Sub
```

Opseg procedura. Private i Public

- Ispred ključnih reči Sub / Function u zaglavlju procedure se mogu naći i dodatne ključne reči **Private** i **Public** koje određuju opseg procedure, tj. njenu vidljivost u odnosu na druge module u projektu.
- Private označava da je procedura dostupna samo procedurama iz istog modula.
- Public označava da je procedura vidljiva svim procedurama iz svih ostalih modula projekta.
- Ukoliko se ne navedu ključne riječi Private i Public, podrazumeva se Public.
- Kreirati novi modul i iz njega pozvati proceduru ImenaListova(), deklarisanu kao Public i kao Private, i videti šta se dešava.

Ključna reč Static

- Nezavisno od Private i Public, ispred Sub / Function se može naći i ključna reč Static koja označava da se promenljive u proceduri čuvaju (ne dealociraju se!) između poziva procedure.
- Drugim rečima, kada se drugi put pozove Static procedura u kojoj je deklarisana neka promenljiva, vrednost te promenljive će biti vrednost sa kraja prvog izvršenja procedure. Znači, promenljive se čuvaju i nakon završetka izvršenja procedure!
- Atribut Static ne utiče na promenljive koje su deklarisane van procedure, čak i ako se koriste u proceduri.

Primer Static procedure

- Šta će biti odštampano u Immediate prozoru nakon izvršenja procedure Prva?

```
Sub Prva()
    Druga
    Druga
    Druga
End Sub

Static Sub Druga()
    Dim x As Integer
    Debug.Print x
    x = x + 2
End Sub
```

Ako se promenljiva ne inicijalizuje, podrazumevana početna vrednost je 0.

- Šta će biti odštampano u Immediate prozoru nakon drugog izvršenja procedure Druga?

Opseg VBA promenljivih

- Svaka promenljiva ima svoj **opseg**, tj. module i procedure gde se promenljiva može koristiti, i **trajanje** (ili životni vek), koje definiše vreme zadržavanja te promenljive u memoriji. Trajanje promenljive je usko vezano sa njenim opsegom.
- U VBA, promenljiva može imati tri tipa opsega:
  - *proceduralni* ili *lokalni*,
  - *privatni* ili *opseg modula*, i
  - *javni*.
- Promenljiva sa lokalnim opsegom je dostupna samo u proceduri gde je definisana.
- Deklarisanje lokalne promenljive se vrši pomoću ključnih reči Dim ili Static unutar procedure.

Opseg VBA promenljivih (nastavak)

- Vrednost Static promenljive se čuva tokom poziva procedure.
- Promenljiva sa privatnim opsegom je dostupna svim procedurama modula u kom se nalazi, ali ne i procedurama iz drugih modula.
- Privatne promenljive zadržavaju svoju vrednost sve dok je predmetni projekat otvoren.
- Privatne promenljive se deklarišu pomoću ključnih reči Dim ili Private na početku modula, ispred prve procedure u modulu.
- Reč Private se ne može koristiti unutar procedure.
- Promenljiva sa javnim opsegom je dostupna svim procedurama u svim modulima u projektu koji je sadrži.
- Javne promenljive se deklarišu ključnom rečju Public i navode se ispred prve procedure u modulu.

Opseg VBA promenljivih - Primer

- Šta će se desiti pozivom procedure Druga iz Module 2?

Module 1

```
Public X as Integer
Private Y as Integer

Sub Prva()
    Dim Z As Integer
    Z = 5
End Sub
```

Module 2

```
Option Explicit

Sub Druga()
    X = 14
    Debug.Print X
    Y = 15
    Debug.Print Y
    Debug.Print Z
End Sub
```

Prosleđivanje argumenata procedurama

- Postoje dva načina prosleđivanja argumenata procedurama, **po vrednosti** (by value) i **po referenci** (by reference).
- Prosleđivanje po vrednosti znači da procedura dobija kopiju promenljive, što za rezultat ima da se prava vrednost promenljive ne može promeniti unutar procedure kojoj je ta promenljiva prosleđena. Pre i posle poziva procedure, vrednost promenljive je ista.
- Prosleđivanje po referenci je prosleđivanje adrese promenljive, što dozvoljava proceduri da promeni pravu vrednost prosleđene promenljive. Svaka promena vrednosti promenljive prosleđene po referenci unutar procedure ostaje važeća nakon izlaska iz procedure.
- U VBA, podrazumevani način je prosleđivanje po referenci!!!
- Prosleđivanja po vrednosti i referenci se naglašavaju rečima **ByVal** i **ByRef** ispred imena argumenta.

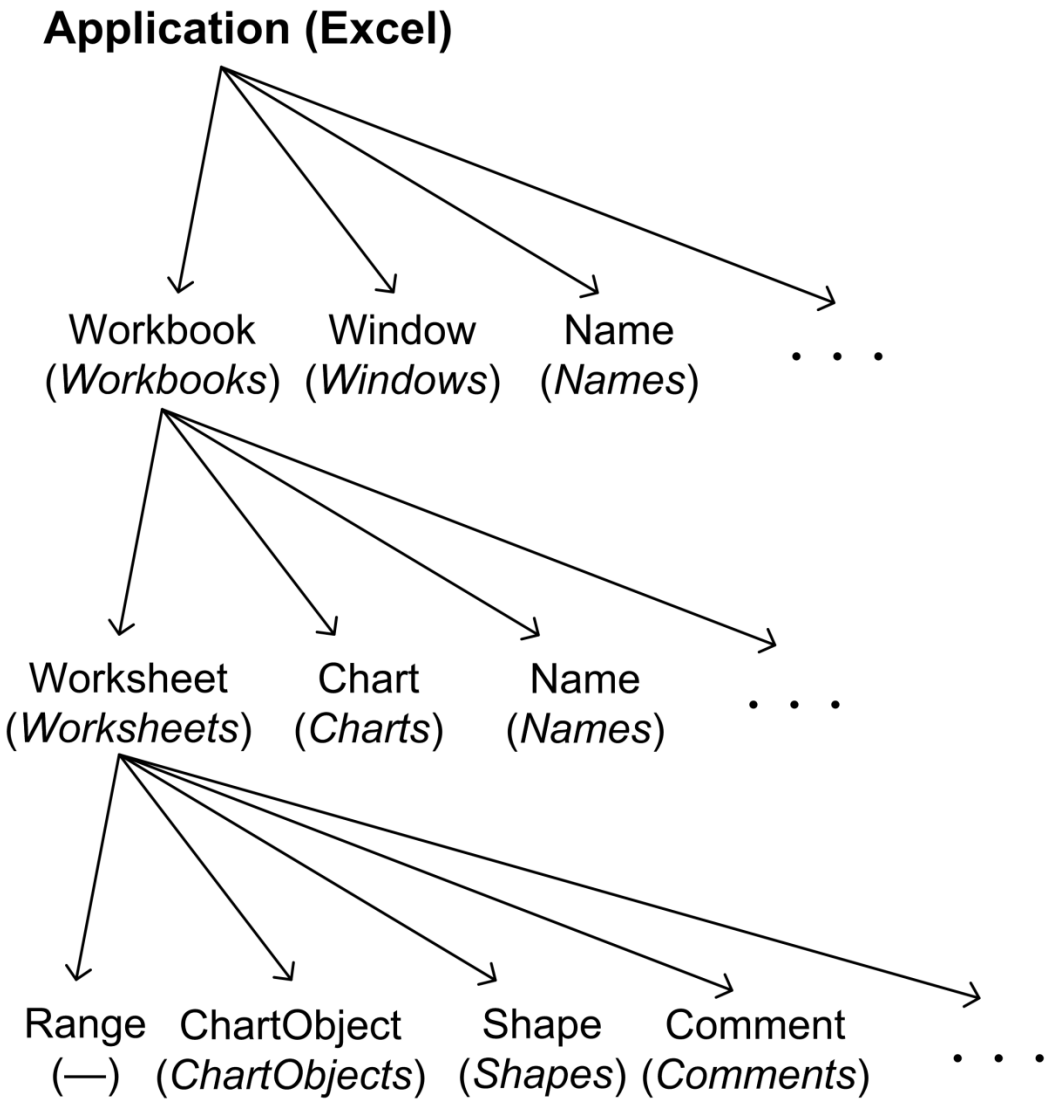
ByVal i ByRef - primer

```
Sub VrednostReferenca()
    Dim N As Integer
    N = 5
    Call PromenaVal(N)
    Debug.Print N
    Call PromenaRef(N)
    Debug.Print N
End Sub

Sub PromenaVal(ByVal Y as Integer)
    Y = Y * 2
End Sub

Sub PromenaRef(ByRef Y as Integer)
    Y = Y * 2
End Sub
```

Objektni model Excel-a



Kolekcije

- Generalno, objektima u objektnom modelu Excel-a pristupamo počev od najvišeg objekta u hijerarhiji - Application.
- Pojedinačnim objektima se vrlo često pristupa kao članovima odgovarajuće kolekcije.
- Kolekcija predstavlja grupu objekata iste klase. Kolekcija za sebe predstavlja objekat.
- Referenciranje (pozivanje) objekta u kolekciji se vrši navođenjem imena objekta ili njegovog rednog broja u malim zagradama nakon imena kolekcije.
- Na primer, prvi radni list, koji se zove Sheet1, se može referencirati na sledeće načine:

```
Worksheets("Sheet1")
Worksheets(1)
```

Referenciranje objekata

- Kroz objektni model se krećemo koristeći operator tačka (**.**), koji razdvaja objekt **kontejner** (levo od tačke) i objekt **član** (desno od tačke).
- Na primer, puna adresa ćelije A1 prvog radnog lista radne sveske VBA.xls bi bila

  Application.Workbooks("VBA.xls").Worksheets(1).Range("A1")
- Korišćenje punih adresa povećava obim i nepreglednost kôda. Na sreću, nije neophodno koristiti pune adrese.
- Objekat Application se praktično može izostaviti pri referenciranju ostalih objekata u hijerarhiji.
- Ako radimo samo sa jednom radnom sveskom, nema potrebe navoditi o kojoj se svesci radi. Tako se referenciranje ćelije A1 svodi na

  Worksheets(1).Range("A1")

Referenciranje objekata (nastavak)

- Ukoliko je prvi radni list aktivan, referenciranje se može svesti na Range("A1")
- Za očekivati je da u VBA postoji objekat Cell, koji bi predstavljao ćeliju radnog lista. Međutim, objekat Cell ne postoji i njegovu ulogu vrši objekat Range.
- Range objekat, osim što može predstavljati jednu ćeliju, Range("C4"), može predstavljati i selekciju ćelija, Range("C4:D8"), ili selekciju nepovezanih ćelija, Range("A1,B2:B10,C4:D8"). Imenovanom opsegu se može pristupiti sa Range("Ime opsega").
- Referenciranje objekata samo po sebi ne vrši nikakvu konkretnu radnju, već omogućava pristup datom objektu. Konkretna radnja bi podrazumevala čitanje ili promenu osobine objekta ili pozivanje određene metode koja vrši neku radnju nad objektom.

Osobine objekata

- Svaki VBA objekat ima određeni skup karakteristika koje se nazivaju **osobinama** objekta. Osobine definišu izgled i poziciju objekta.
- Na primer, svaki objekat Window ima osobinu WindowState kojom se dati prozor može prikazati kao minimizovan, maksimizovan ili normalan. Objekat Range ima osobinu Value, pomoću koje se vrednost ćelije može očitati ili promeniti.
- Osobina objekta se referencira koristeći operator tačka, tj. u obliku

  Objekat.Osobina
- Na primer, vrednost ćelije A1 prvog radnog lista se može dobiti sa

  Worksheets(1).Range("A1").Value

  i ta se vrednost može prikazati ili dodeliti nekoj promenljivoj.
- U prethodnoj naredbi, Worksheets(1).Range("A1") predstavlja objekat, a Value osobinu.

Osobine objekata (nastavak)

- Promena vrednosti osobine objekta se vrši na sledeći način:

  Objekat.Osobina = Vrednost

  gde Vrednost predstavlja izraz čija se vrednost dodeljuje datoj osobini objekta. Vrednost može biti bilo kojeg tipa VBA promenljivih, pri čemu ćemo najčešće raditi sa numeričkim, stringovnim i logičkim tipom.

  ```
  Range("A1").Value = 23.11
  Range("A1").Font.Size = 12
  Range("A1").Font.Name = "Times New Roman"
  Range("A1").Font.Bold = True
  ```

- Većina objekata ima *podrazumevanu osobinu*, koja se pri referenciranju može izostaviti. Na primer, podrazumevana osobina objekta Range je Value. Instrukcijom

  ```
  Range("A1") = 23.11
  ```

  bi postigli isti efekat kao i sa Range("A1").Value = 23.11.

Metode objekata

- Objekti imaju i **metode**, koje predstavljaju akcije koje možemo vršiti nad objektima. Metoda se takođe poziva koristeći operator tačka:

  Objekat.Metod

- Na primer, objekat Range ima metodu Clear koja briše sadržaj i formatiranje predmetnog opsega. Na primer, instrukcija

  Worksheets("Sheet1").Range("A1,B2,C3").Clear

  briše sadržaj i formatiranje ćelija A1, B2 i C3. Brisanje sadržaja opsega, uz očuvanje formata, se vrši metodom ClearContents.

- Nekoliko primera metoda:

  Application.Quit

  Workbooks(1).Save

  Worksheets(1).Delete

Metode objekata (nastavak)

- Neke metode zahtevaju argumente kojima se specificira radnja.
- Na primer, ukoliko želimo da iskopiramo sadržaj jednog opsega u drugi, potrebno je koristiti metodu Copy. Ova metoda ima jedan argument koji je opcion i koji predstavlja destinaciju kopiranja. Na primer, ukoliko želimo da iskopiramo sadržaj opsega A1:C3 prvog radnog lista u opseg A4:C7 drugog radnog lista potrebno je pozvati metodu Copy na sledeći način:

```
Worksheets(1).Range("A1:C3").Copy   Worksheets(2).Range("A4:C7")
```

- U prethodnoj naredbi, Worksheets(2).Range("A4:C7") predstavlja destinaciju kopiranja. Destinacija se kao argument može izostaviti i u tom slučaju se sadržaj opsega kopira na Clipboard.

Range objekat. Osobina Range

- U VBA, ćelija ili opseg ćelija radnog lista su objekti tipa Range.
- Range objekat se nalazi unutar Worksheet objekta i pristupa mu se koristeći:
  - osobinu Range objekata Worksheet i Range,
  - osobinu Cells objekata Worksheet i Range,
  - osobinu Offset objekta Range.
- Osobina Range objekta Worksheet se koristi na sledeći način:

`Worksheets(1).Range("C2").Value = 1`	Jedna ćelija
`Worksheets(1).Range("C2:D7").Value = 2`	Opseg ćelija
`Worksheets(1).Range("C2:D7, F4, H8").Value = 3`	Unija opsega
`Worksheets(1).Range("C2:D7  B3:E5").Value = 4`	Presek opsega
`Worksheets(1).Range("MojOpseg").Value = 5`	Imenovani opseg

Osobina Cells

- Najčešći oblik korišćenja osobine Cells je

  Objekat.Cells(rowIndex, columnIndex)

- Na primer, sa

  Worksheets("Sheet1").Cells(3,5) = 51

  se pristupa ćeliji u preseku treće vrste i pete kolone, tj. ćeliji E3, i u nju se upisuje broj 51.

- Maksimalan broj vrste i kolone zavisi od verzije Office-a. Kod Excel-a 2000/2003, imamo 2^8=256 kolona i 2^{16}=65536 vrsta, dok kod Excel-a 2007-2013 imamo 2^{14}=16384 kolona i 2^{20}=1048576 vrsta.

Osobina Cells (nastavak)

- Druga dva načina korišćenja osobine Cells su:
  - Objekat.Cells(rowIndex)
  - Objekat.Cells
- Kad se navodi samo jedan argument, ćelijama opsega se pristupa kao da je opseg razvijen u niz, vrstu po vrstu. Na primer, instrukcijama:

  Worksheets("Sheet1").Cells(5) = 11

  Worksheets("Sheet1").Cells(258) = 12

  se pristupa ćelijama E1 i B2 (kod Office-a 2000/2003).
- Osobina Cells bez argumenata vraća sve ćelije referenciranog objekta. Na primer, instrukcija

  Worksheets(2).Cells = 45

  u čitav drugi radni list upisuje broj 45.

Osobina Cells (nastavak)

- Kada se osobina Cells koristi sa Range objektima, argumenti u zagradi će definisati relativnu adresu ćelije u odnosu na ćeliju koja se nalazi u gornjem levom uglu referenciranog opsega. Na primer:

  Range("A1:C5").Cells(2,2) = 5

  menja sadržaj ćeliji B2.

- Na ovaj način ćemo često koristiti osobinu Cells da obiđemo čitav opseg ćelija, ćeliju po ćeliju.

Obilazak opsega pomoću osobine Cells

- Ukoliko želimo da obiđemo opseg A1:D5, ćeliju po ćeliju, i da izvršimo određenu operaciju nad svakom ćelijom (recimo, sabiranje brojeva koji se nalaze u ćelijama), to možemo uraditi pomoću dve petlje i osobine Cells na sledeći način:

```
For I = 1 To Range("A1:D5").Rows.Count
    For J = 1 To Range("A1:D5").Columns.Count
        Suma = Suma + Range("A1:D5").Cells(I, J)
    Next
Next
```

- Broj vrsta i kolona predmetnog opsega se dobija pomoću osobine Count objekta Range.

Osobina Offset

- Osobina Offset objekta Range vraća Range objekat. Sintaksa ove osobine je:

  Objekat.Offset(rowOffset, columnOffset)

- Argumenti ove osobine definišu relativni pomeraj (eng. *offset*) ćelije u odnosu na gornji levi ugao opsega. Ovi argumenti mogu biti pozitivni (krećemo se dole, odnosno desno), negativni (krećemo se gore, odnosno levo), ili nula. Na primer, instrukcijama:

  Range("C3").Offset(1,0).Value = 5

  Range("C3").Offset(-1,0).Value = 6

  menjamo vrednost ćelija C4 i C2. Ukoliko je objekat Range("A1"), argumenti osobine Cells ne mogu biti negativni, tj. desiće se greška pri unosu negativnog pomeraja, jer tražene ćelije ne postoje. Specijalno, Offset(0,0) odgovara samom objektu.

- Razmislite sami o obilasku opsega, ćeliju po ćeliju, koristeći Offset.

Primer 1

- Napisati proceduru koja u opsegu C2:D5 prvog radnog lista prolazi kroz sve ćelije i u svaku upisuje jedan slučajan broj. Ukoliko je upisani broj veći od 0.5, datoj ćeliji postaviti veličinu fonta na 14.

```
Sub Upisi()
    Dim I As Integer, J As Integer, Broj As Single
    For I = 1 To Range("C2:D5").Rows.Count
        For J = 1 To Range("C2:D5").Columns.Count
            Broj = Rnd
            Range("C2:D5").Cells(I, J) = Broj
            If Broj > 0.5 Then
                Range("C2:D5").Cells(I, J).Font.Size = 14
            End If
        Next
    Next
End Sub
```

Primer 1 – Poboljšano rešenje

```
Sub Upisi()
    Dim I As Integer, J As Integer, Broj As Single
    Dim R As Range
    Set R = Range("C2:D5")
    For I = 1 To R.Rows.Count
        For J = 1 To R.Columns.Count
            Broj = Rnd
            R.Cells(I, J) = Broj
            If Broj > 0.5 Then
                R.Cells(I, J).Font.Size = 14
            End If
        Next
    Next
End Sub
```

Upotreba ključne reči Set za davanje vrednosti objektima

Primer 2

- Napisati proceduru koja određuje maksimalan broj selektovanog opsega i prikazuje ga pomoću MsgBox-a.

```
Sub MaksBroj()
    Dim I As Integer, J As Integer, Maks As Double
    Maks = Selection.Cells(1, 1)
    For I = 1 To Selection.Rows.Count
        For J = 1 To Selection.Columns.Count
            If Selection.Cells(I, J) > Maks Then
                Maks = Selection.Cells(I, J)
            End If
        Next
    Next
    MsgBox "Najveći broj je " & Maks
End Sub
```

Objekt Selection predstavlja tekuću selekciju ćelija