



# Programiranje kroz aplikacije

Korisničke forme - nastavak  
Upravljanje greškama

# Pozivanje forme sa palete alatki

- Korisničke forme se mogu prikazati pomoću odgovarajućeg dugmeta sa palete alatki.
- Možemo kreirati sopstvenu paletu alatki i sa nje pozvati formu, ili koristiti neku od postojećih paleta.
- Za oba načina je zajedničko da se dodavanje dugmeta koje prikazuje formu najčešće vrši pri otvaranju radne sveske, tj. u okviru procedure [Workbook\\_Open](#).
- Prikaz forme predstavlja poziv procedure koja u sebi sadrži naredbu [UserForm.Show](#)

# Dodavanje dugmeta postojećoj paleti

- Dugme koje se dodaje paleti alatki je VBA objekat tipa `CommandBarButton`.
- Deklaracija dugmeta i dodavanje istog paleti se vrši na sledeći način:  
`Dim Dugme As CommandBarButton`  
`Set Dugme = Application.CommandBars("Standard").Controls.Add( _`  
`Type:=msoControlButton, Temporary:=True)`
- Dodavanje kontrole se vrši metodom `Add` kolekcije `Controls`. Argument `Type` određuje tip kontrole, dok `Temporary` određuje da li se kontrola automatski briše nakon zatvaranja aplikacije (`True`) ili ne (`False`). Podrazumevano je `False`.

# Osobine dugmeta

- Nakon dodavanja dugmeta paleti, definišemo mu osobine. Na primer:

## With Dugme

```
.Style = msoButtonCaption  
.Caption = "Dugme"  
.OnAction = "Procedura"
```

## End With

definiše dugme samo sa tekстом (osobina **Style**), tekst koji će biti ispisan na dugmetu (osobina **Caption**), kao i proceduru koja se poziva pritiskom na dugme (osobina **OnAction**).

- Ako želimo da, umesto teksta, definišemo ikonicu koja će biti prikazana na dugmetu, imali bi konstrukciju sličnu sledećoj:

## With Dugme

```
.Style = msoButtonIcon  
.Picture = LoadPicture("C:\Slika.bmp")  
.OnAction = "Procedura"
```

## End With

Dozvoljena je i kombinacija teksta i ikonice, kada je stil **msoButtonIconAndCaption**

# Kreiranje nove palete alatki

- Paleta koja se dodaje je VBA objekat tipa `CommandBar`.
- Dodavanje palete se vrši metodom `Add` kolekcije `CommandBars`.

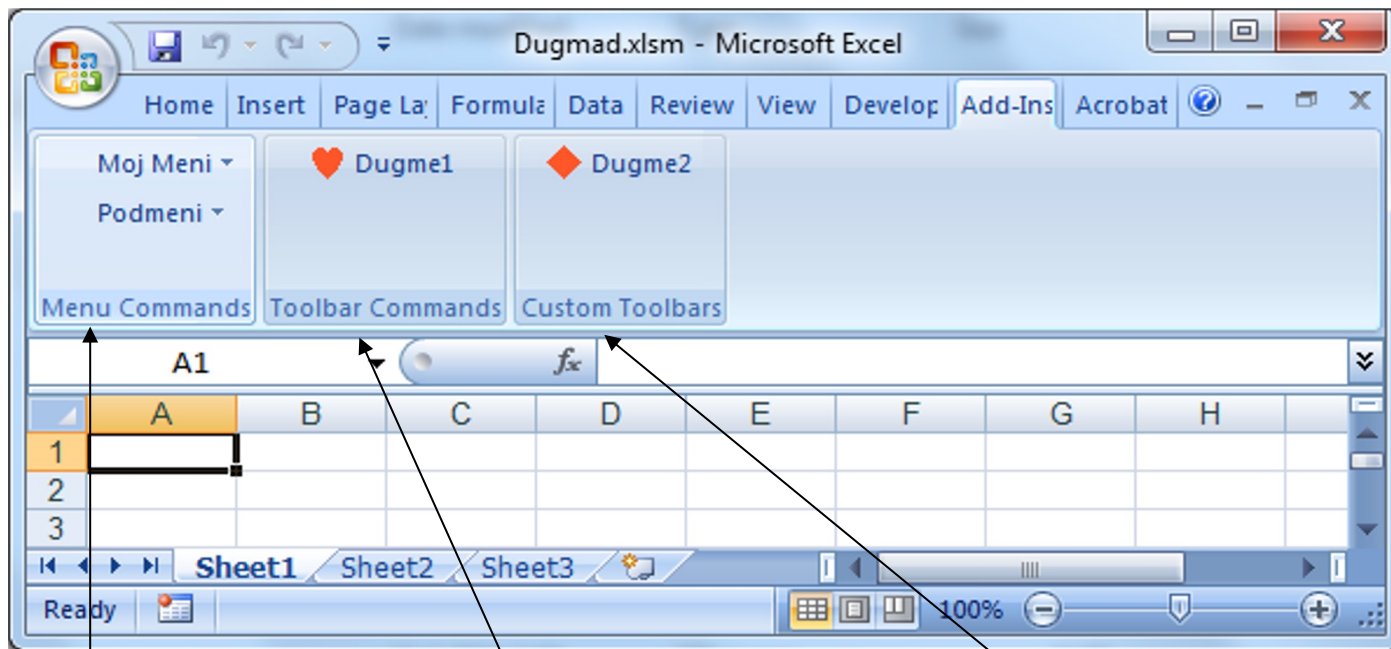
```
Dim Paleta As CommandBar
```

```
Set Paleta = Application.CommandBars.Add(Name:="NasaPaleta", _  
    Position:=msoBarTop, Temporary:=True)
```

```
Paleta.Visible = True
```

- Argument `Name` definiše ime palete. Argument `Position` definiše poziciju palete u odnosu na prozor aplikacije. Konstantom `msoBarTop` se paleta smešta na vrh prozora. Argument `Temporary` određuje da li se paleta automatski briše nakon zatvaranja aplikacije (`True`) ili ne (`False`). Podrazumevano je `False`.
- Kreirana paleta podrazumevano nije vidljiva. Da bi je učinili vidljivom, potrebno je podesiti njenu osobinu `Visible` na `True`.
- Dodavanje dugmeta na paletu `NasaPaleta` se vrši na ranije opisan način, s tim što se paleta referencira sa `CommandBars("NasaPaleta")`.

# Office 2007+



Novokreirani meniji i podmeniji postojećih menija se smestaju u grupu Menu Commands

Dugme dodato postojećoj paleti se smesta u grupu Toolbar Commands

Paleta alatki se smesta u grupu Custom Toolbars

- Od Office-a 2007, aplikacija koristi traku (ribbon) umesto menija i paleta alatki.
- Sve korisničke palete, meniji i dugmad se smestaju na *Add-Ins* tab.
- Dobra vest je da se VBA kôd ne menja.

# Primjer 1

- Kreirati dugme u paleti alatki sa natpisom “Sortiraj”. Dugme treba smjestiti u posebnu paletu alatki koja se zove “Moja paleta”. Klik na dugme otvara Formu sa 3 proizvoljna elementa. Na dnu forme se nalazi dugme Zatvori koje zatvara kreiranu Formu.
- Dugmetu dodati proizvoljnu sliku.
- Kreirati još jednu dugme koje treba dodati u standardnu paletu alatki.

# Upravljanje greškama

- Greške se dele na **sintaksne greške** (greške u pisanju kôda) i **greške prilikom izvršavanja** (eng. run-time errors). Od interesa za nas su greške prilikom izvršavanja.
- Greške prilikom izvršavanja uzrokuju prekid rada procedure uz prikaz dijalog prozora na kome je dat broj greške i njen opis. Na primer, ako se u proceduri nađe  
$$A = B / 0$$
pojaviće se prozor greške deljenja sa nulom.
- Dobro napisana procedura treba da vodi računa o pojavi grešaka prilikom izvršavanja, tj. treba da ih hvata i obrađuje.
- Obrada grešaka podrazumeva preduzimanje koraka kako bi se izvršavanje procedure neometano nastavilo.



# Naredba On Error

- Naredba **On Error** omogućava hvatanje (detekciju) grešaka i koristi se u dva osnovna oblika:

On Error GoTo linija

On Error Resume Next

- U prvom obliku, **linija** predstavlja labelu ili broj programske linije na koju će “skočiti” kontrola toka u slučaju greške, tj. prvu liniju kôda za obradu greške.
- Ova linija se mora naći u istoj proceduri gde se nalazi naredba **On Error**, tj. kôd za obradu grešaka je ograničen na proceduru gde se greška pojavljuje.
- Naredba **On Error** se stavlja ispred dela kôda gde očekujemo grešku.

# Primer – Deljenje sa nulom

```
Sub ObradaDeljenjaSaNulom()  
  Dim A As Integer, B As Integer  
  On Error GoTo ObradiGresku  
  B = 15  
  A = B / 0  
  B = A ^ 2  
  Exit Sub  
ObradiGresku:  
  MsgBox "Greška broj " & Err.Number & ": Pokušaj deljenja sa nulom."  
End Sub
```

- Svaka greška u VBA ima pridružen jedinstven kôd (ceo broj).
- Kôd greške se dobija pomoću osobine `Number` objekta `Err`. Objekat `Err` sadrži informacije o greškama prilikom izvršavanja i njegova podrazumevana osobina je `Number`.
- Ukoliko nije došlo do greške, osobina `Number` ima vrednost 0.

# On Error Resume Next

- Kod **On Error Resume Next**, kontrola se prebacuje na prvu naredbu nakon naredbe koja je uzrokovala grešku, tj. *greška se ignoriše*.
- Jedan broj grešaka se može ignorisati. Na primer, ako ćelija ne sadrži komentar, doći će do greške pri pokušaju čitanja komentara.
- U nastavku je dat primer procedure koja ispisuje komentare svih ćelija datog opsega **R**.

```
Sub Komentari()  
    Dim R As Range, celija As Range  
    Set R = Worksheets(1).Range("A1:C10")  
    On Error Resume Next  
    For Each celija In R.Cells  
        MsgBox "Ćelija " & celija.Address & vbCrLf & _  
            "Komentar: " & celija.Comment.Text  
    Next  
    On Error GoTo 0  
End Sub
```

# On Error GoTo 0

- Naredba **On Error GoTo 0** onemogućava hvatanje grešaka u tekućoj proceduri. Iako se ovaj oblik može protumačiti kao prvi oblik naredbe, gde **0** predstavlja broj linije kojom počinje kôd za obradu greške, to nije slučaj. Čak i da postoji linija procedure označena sa **0**, ova naredba onemogućava hvatanje grešaka.
- Ukoliko procedura ne sadrži liniju **On Error GoTo 0**, onemogućavanje hvatanja grešaka se postiže automatski, izlaskom iz procedure.
- Hvatanje grešaka je ponekad jednostavnije koristiti nego da sa mnoštvom uslova predviđamo svaku moguću grešku koja se može desiti u proceduri.
- Na primer, kako možemo proveriti da li postoji određena radna sveska? Umesto da proveravamo postojanje sveske pre njenog otvaranja, možemo uhvatiti grešku koja se pojavljuje u pokušaju da je otvorimo.

# Primer sa korišćenjem grešaka

- Napisati proceduru koja utvrđuje da li aktivna radna sveska sadrži list Spisak i korisniku javlja odgovarajuću poruku.

```
Sub TrazenjeLista()  
  On Error Resume Next  
  ActiveWorkbook.Worksheets("Spisak").Activate  
  If Err.Number = 0 Then  
    MsgBox "Postoji list Spisak"  
  Else  
    MsgBox "Ne postoji list Spisak"  
  End If  
End Sub
```