

Увод у објектно програмирање:

Функције, процедуре, догађаји, модули

Професор: **др Светлана Штрбац-Савић**

Маил / Кабинет: **svetlanas@viser.edu.rs / 501**

- Процедуре се користе за имплементацију задатака који се понављају, као што су на пример прорачуни који се често користе.
- VB подржава два типа процедура подпрограме (*subroutine*) и функције (*function*).
- Методи представљају имплементацију понашања објекта.
- Ако се неки метод користи у више објекта, довољно је да је само једном написан и преведен у једној класи и да је проглашен за јавни (Public), онда је он доступан свим објектима у тој апликацији.

- Ако се један потпрограм, не само независно пише, већ и независно преводи, онда он може бити позван из било ког програма, и тада је он екстерни.
- Екстерни потпрограми могу формирати библиотеке.
- Када је једном написан и истестиран, екстерни потпрограм може се користити било када и у било ком програму, па се тиме знатно олакшава писање сложених програма.

- Методи се имплементирају помоћу потпрограма, односно функција (*function*), процедура (*Procedure, Subroutine*) или догађаја (*Event*).
- Без обзира на начин имплементације, методима се приликом позивања најчешће морају проследити и неки подаци неопходни за њихово извршавање.
- То су параметри, а они се могу пренети из позивајућег програма на два начина:
- преношењем копије податка (*ByVal*) и
- преношењем референце на податке и објекте (*ByRef*).
- Сваки од ових начина има своје предности и недостатке.

- **Процедуре** се, дакле, састоје из операција или низа операција, али никада не враћају неку вредност.
- Оваква врста процедура дефинише се навођењем службене речи **Sub** и имена процедуре, а у претходним верзијама .NET и VisualStudio позивала навођењем имена или службене речи **Call** и имена процедуре. У .NET 2017.наводи се само име процедуре и листа стварних параметара са којима се она позива.
- На крају сваке процедуре налази се службена реч **End Sub**.
- Из процедуре се може изаћи и раније, када се наиђе на команду **Exit Sub**

- **Функције** за разлику од процедура враћају увек једну и само једну вредност која се уписује у име функције (на пример резултат неког прорачуна).
- VB садржи мноштво већ уграђених (built-in) функција, на пример, функција Date() враћа текући датум и време или функција Val(str) коју смо помињали више пута.
- Осим ових, већ постојећих **функција**, можемо креирати и сопствене кориснички дефинисане функције (user-defined functions, UDF).

- Типична функција има следећи изглед:

```
Public Function ImeFunkcije (ListaArgumenata) As TipPodatka  
    Return Neka Vrednost  
End Function
```

На пример:

```
Public Function PovrsinaKruga (ByVal _ Poluprecnik As Single) As Single  
    Const Pi = 3.14159  
    Return Pi * Poluprecnik*Poluprecnik  
End Function
```

- Функција којом се сабирају два броја:

```
Public Function zbir(ByVal operand1 As _ Double, ByVal operand2 As Double) As Double  
    Return operand1 + operand2  
End Function
```

- Функција је декларисана као јавна (Public) да би била “видљива” унутар целог пројекта.

- Функција се позива тако што у изразу наведе име функције, код се у аргументу функције наводе стварни параметри. Резултат рада функције (повратна вредност) се враћа на место позива функције:

`OsnovaKupe=PovrsinaKruga (Poluprecnik)`

- С обзиром да функције враћају једну вредност, можемо их користити у изразима:

`ZapreminaKupe= (1/3) *VisinaKupe*PovrsinaKruga (Poluprecnik)`

- Једна функција се може током рада програма позивати више пута са различитим параметрима. Вредност функције ће зависити од прослеђених параметара.

Преоптерећење функције - Overloading

- Креирање више функција са истим именом је могуће у VB-у, под условом да им се разликују број или тип аргумената.
- На пример имамо у истом пројекту две функције zbir, али не раде са исти типовима података:

```
Public Function zbir(ByVal operand1 As Double, ByVal operand2 As Double) As Double  
    Return operand1 + operand2
```

```
End Function
```

```
Public Function zbir(ByVal operand1 As String, ByVal operand2 As String) As String  
    Return operand1 + operand2
```

```
End Function
```

- Пренос променљивих по референци има несумњивих предности и практично је једини могући начин за пренос великих блокова података, низова, матрица, слика, итд.
- Овај начин преноса података носи са собом и неке опасности, а главна је могућност случајне (или злонамерне) измене оригиналне вредности податка.
- То ће се десити увек када се име податка који је пренет по референци нађе на левој страни израза за доделу вредности.

- Подаци се некада морају прослеђивати функцијама и процедурама по референци. Треба водити рачуна да ће оригинални подаци бити промењени у колико функција односно процедура изврши доделу вредности овим подацима.
- Пример: Функција за сабирање низова:

```
Public Sub zbir(ByRef NizA() As Double, ByRef NizB() As Double, ByRef NizC() As Double, ByVal n As Int16)  
  Dim i As Int16  
  
    For i = 0 To n - 1  
      NizC(i) = NizA(i) + NizB(i)  
    Next  
End Sub
```

- Функције могу да позивају друге функције или саме себе
- Рекурзивне функције су оне функције које позивају саме себе. Приликом реализације ових функција мора се водити рачуна да се обезбеди излазак из рекурзије.
- Пример: Функција за рачунање факторијела:

```
Public Function faktoriel(ByVal n As Int16) As Integer  
If n=1 Then Return 1  
    Else  
        Return faktoriel(n-1) *n  
    End If  
End Function
```

- За све податке који се користе у било ком делу програма (главном програму, потпрограмирама или блоку инструкција) значајна су два атрибута:
- област важења (*scope*) и
- циклус, тј. век трајања (*extent*).

- **Област важења податка** је онај део програма у коме име тог податка означава увек исту величину.
- Унутар потпрограма (или блока инструкција) могу постојати неке променљиве које се користе само у њему, а у другим деловима програма нису познате.
- Такве променљиве се зову локалне.
- Локалне променљиве у разним потпрограмима, или деловима програма, могу имати исто име, али то име означава различите податке.

- Међутим, у програмима се могу користити и подаци чије је име познато (и исто) у свим деловима програма.
- Ти подаци су заједнички за главни програм (Main) и за све његове потпрограме (Procedure, Function, Subroutine), и под тим именом се могу позивати у било ком потпрограму.
- Такви подаци називају се глобални.

- Једну врсту глобалних променљивих чине и променљиве које нису познате у свим деловима програма већ су познате у неколико његових потпрограма.
- Употреба глобалних променљивих такође носи са собом опасност да дође до нежељене промене оригиналне вредности, а такође отежава исправљање грешака и смањује читљивост програма.

- **Циклус трајања податка** означава временски период постојања тог податка у меморији рачунара, и у уској је вези са извршавањем програма.
- Неки подаци постоје у меморији рачунара само за време док се извршава један блок обраде или неки потпрограма, и они се називају **аутоматски** подаци.
- Чим се изврши тај део програма они ослобађају меморију, тј. не постоје у другим деловима програма.
- Подаци који се налазе у меморији све време извршавања програма зову се **статички**.
- Ове особине треба имати у виду када се преносе подаци између програма и потпрограма

- У Visual Basic-у постоје следећи опсези важења променљиве:
 - **на нивоу блока** (нпр. унутар If ... End If и For ... Next) – могу се користити само унутар блока инструкција у којем су декларисане,
 - **локални** (на нивоу процедуре) – важи само за време извршавања процедуре у којој је декларисана,
 - **на нивоу обрасца** – важи само док је отворен образац или извештај у коме је дефинисана. Ове променљиве се декларишу у одељку **Declarations** модула обрасца, помоћу службене речи **Private**

- **на нивоу модула** – променљиве које важе у свим процедурама и функцијама модула у коме су декларисане
- **са глобалним опсегом важења** – јавне променљиве чија је област важења у оквиру целог пројекта, дефинишу се службеном речју Public испред имена.

- Функција за сабирање матрица:

```
Public Sub zbir(ByRef MatricaA(,) As Double, ByRef MatricaB(,)
As Double, ByVal RedMatrice As Int16)
Dim vrsta, kolona As Int16

    For vrsta = 0 To RedMatrice - 1
    For kolona = 0 To RedMatrice - 1
        C(vrsta, kolona) = MatricaA(vrsta, kolona) + MatricaB(vrsta, kolona)
    Next
    Next
End Sub
```