

Uvod u programiranje

Programiranje 1 (550)

Poglavlje 5

Podatkovne strukture

dr. sc. Josip Musić
jmusic@fesb.hr

Originalne slideove izradio:
Teo Žuljević, dipl. ing.
teo.zuljevic@fesb.hr

Pregled ⁽¹⁾

- Podatkovne strukture
- Nizovi (polja)
 - Što je niz (polje)?
 - Deklariranje varijable niza
 - Red (rank) niza
 - Pristupanje elementima niza
 - Provjera granica niza
 - Usporedba nizova sa zbirkama
- Stvaranje nizova
 - Stvaranje primjera niza
 - Inicijalizacija elemenata niza
 - Inicijalizacija elemenata više-dimenzionalnog niza
 - Stvaranje niza izračunske veličine
 - Kopiranje varijable niza

Pregled (2)

- Korištenje niza
 - `System.Array` klasa
 - Svojstva niza
 - Postupci niza
 - Vraćanje niza iz procedure
 - Prosljeđivanje niza kao parametra
 - Plitko i dubinsko kopiranje
 - Korištenje `For Each` strukture sa nizovima
- Nazupčani nizovi
- Sortiranje i pretraživanje nizova
- Podatkovna struktura `Stack`
- Podatkovna struktura `Queue` (Red)
- Stvaranje vlastitih vrijednosnih tipova
 - Strukturni tipovi
 - Nabranja

Podatkovne strukture

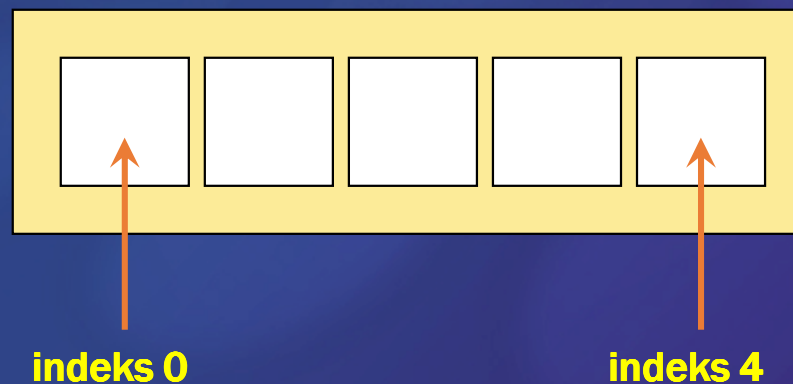
- Dva su osnovna načina grupiranja podataka:
 - strukture
 - nizovi (array)
- **Strukture**
 - grupe povezanih podataka različitih tipova
 - primjer: ime, starost, spol -> struktura osoba
 - pristup pojedinim članovima navođenjem imena polja
 - vrijednosni tip podatka
- **Nizovi**
 - grupe povezanih podataka istog tipa
 - primjer: brojevi zgrada - > ulica
 - pristup pojedinim elementima navođenjem indeksa
 - referencni tip podatka
 - povezani memorijski blok
 - pristup svim elementima niza jednako brzo

Nizovi (polja)

- Što je niz?
- Deklariranje varijable niza
- Red (rank) niza
- Pristupanje elementima niza
- Provjera granica niza
- Usporedba nizova sa zbirkama

Što je niz (array) ?

- Niz je povezani skup elemenata
 - svi elementi niza imaju isti tip podatka
 - pojedinim elementima se pristupa navođenjem cjelobrojnog indeksa



Deklariranje varijable niza

- Prilikom deklariranja varijable niza potrebno odrediti:
 - ime niza
 - tip podatka elementa niza
 - red niza
- Sintaksa:

```
Dim arrayname As datatype()
```

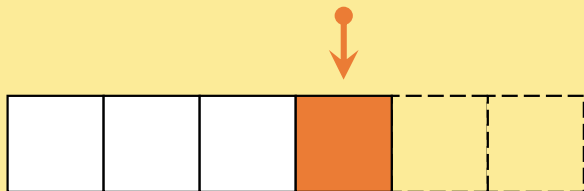
Red niza

- Red niza se naziva i dimenzijom niza

Dim row As Integer()

Red 1: jedno-dimenzionalni niz

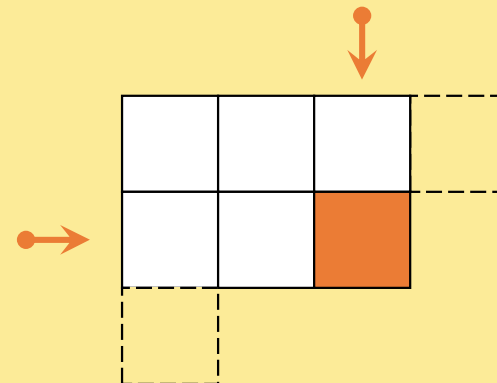
jedan indeks pridružen
svakom **Integer** elementu



Dim grid As Integer(,)

Red 2: dvo-dimenzionalni niz

dva indeksa pridružena
svakom **Integer** elementu

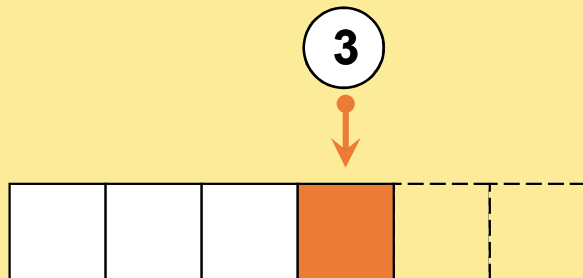


Pristupanje elementima niza

- Potrebno navesti **Integer** indeks za svaki red
 - indeksi su nulto-bazirani

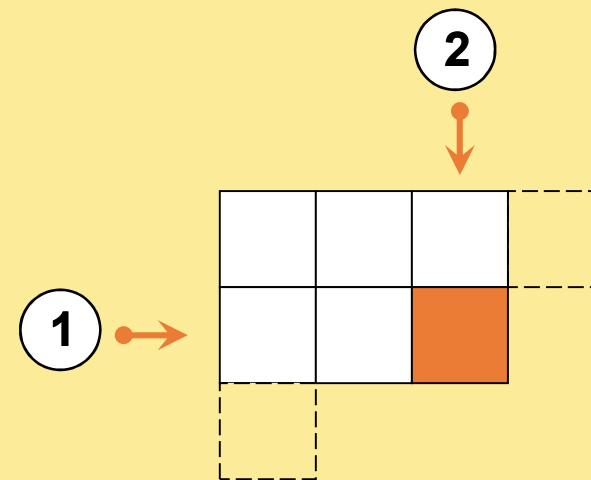
Dim row As Integer()

Red 1: jedno-dimenzionalno niz
jedan indeks pridružen svakom
Integer elementu



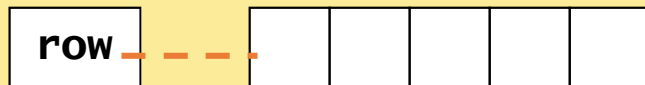
Dim grid As Integer(,)

Red 2: dvo-dimenzionalni niz
dva indeksa pridružena svakom
Integer elementu



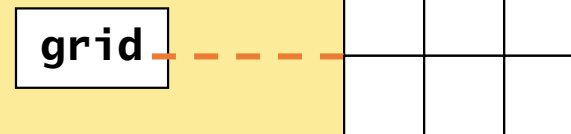
Provjera granica niza

- Nad svim pokušajima pristupa elementima niza se izvodi provjera granica
 - pogrešan indeks podiže **IndexOutOfRangeException** iznimku
 - koristi se **Length** svojstvo i **GetLength** postupak



`GetLength(0) = 5`

`Length = 5`



`GetLength(0) = 2`

`GetLength(1) = 3`

`Length = 2 * 3 = 6`

Usporedba nizova za zbirke

- Niz se ne može povećati kada je pun
 - klase zbirke, poput `ArrayList`, mogu
- Niz je namijenjen za pohranu elemenata jednog tipa
 - zbirke su oblikovane za pohranu elemenata različitih tipova
- Elementi niza ne mogu imati pristup samo za čitanje
 - zbirke imaju pristup samo za čitanje
- Nizovi su brži ali manje fleksibilni
 - zbirke su blago sporije ali znatno fleksibilnije

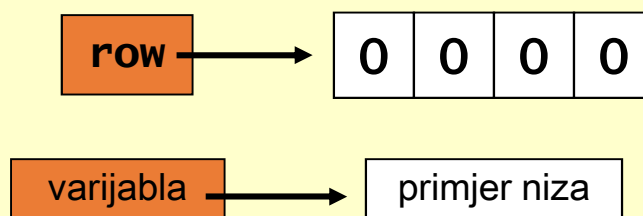
Stvaranje nizova

- Stvaranje primjera niza
- Inicijalizacija elemenata niza
- Inicijalizacija elemenata više-dimenzionalnog niza
- Stvaranje niza izračunske veličine
- Kopiranje varijable niza

Stvaranje primjera niza

- Deklariranje varijable niza ne znači i stvaranje niza
- potrebno je navesti **New** za eksplicitno stvaranje primjera niza
- elementi niza brojanog tipa imaju implicitnu podrazumijevanu vrijednost 0

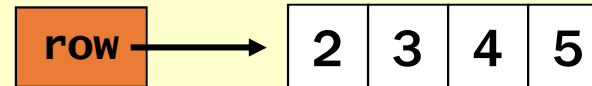
```
Dim row As Long() = New Long(3) {}
```



Inicijalizacija elemenata niza

- Elementi niza mogu biti eksplicitno inicijalizirani
 - pridruživanjem vrijednosti elementu
 - korištenjem inicijalizatora niza
 - kraći oblik inicijalizacije

```
Dim row As Long() = New Long(3) {}  
row(0) = 2  
row(1) = 3  
row(2) = 4  
row(3) = 5
```



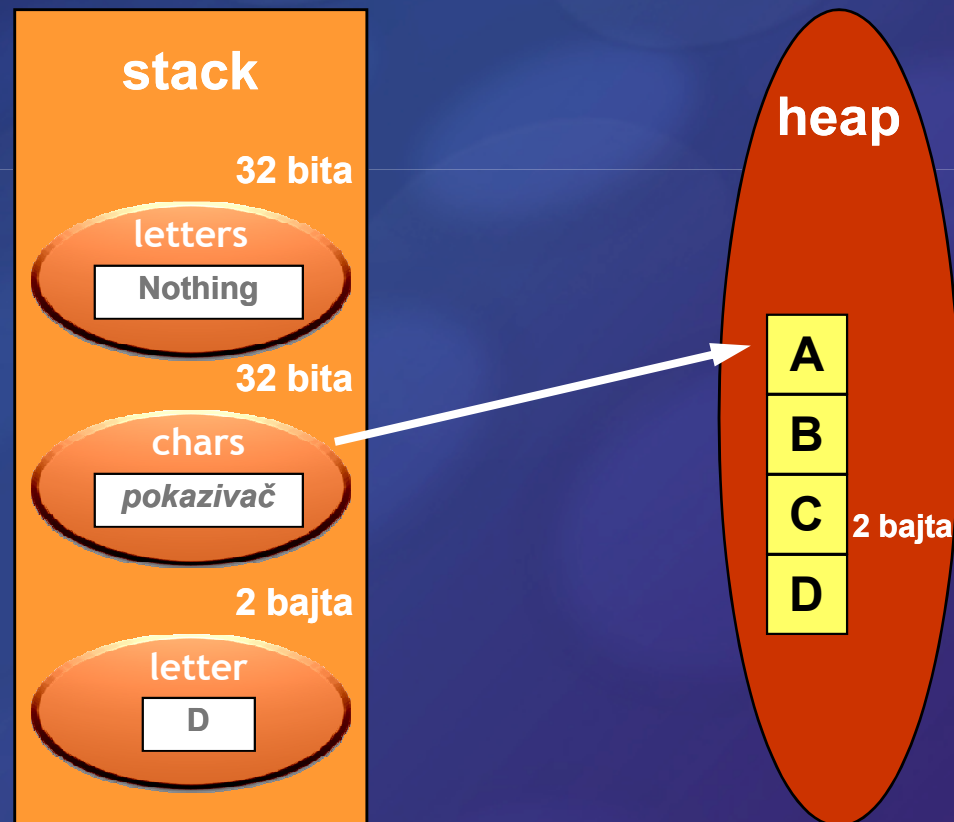
```
Dim row As Long() = New Long() {2,3,4,5}
```

'skraćeni oblik

```
Dim row As Long() = {2,3,4,5}
```

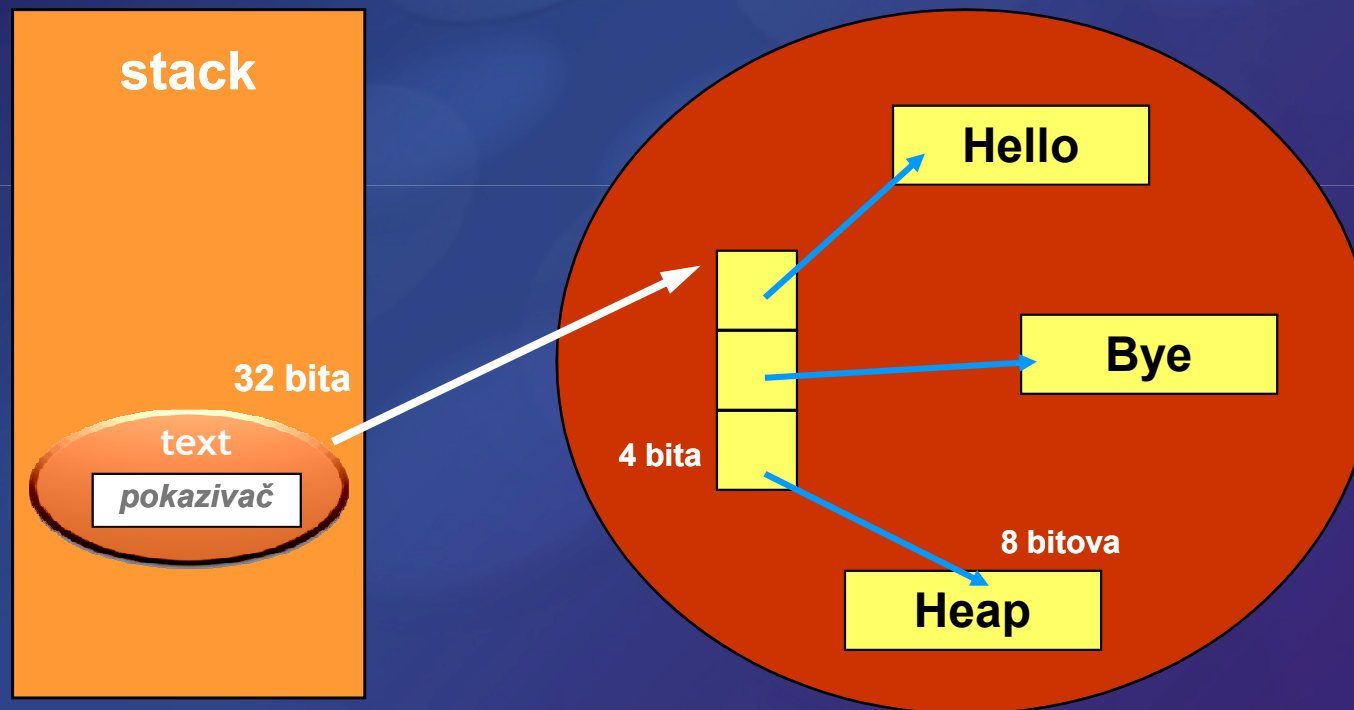

Niz elemenata vrijednosnog tipa

```
Dim letter As Char = "D"c  
Dim chars As Char() = {"A"c, "B"c, "C"c, letter}  
Dim letters As Char()
```



Niz elemenata referencnog tipa

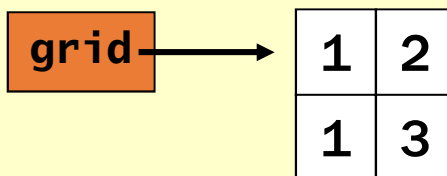
```
Dim text As String() = {"Hello", "Bye", "Heap"}
```



Inicijalizacija više-dimenzionalnog niza

- Korištenjem inicijalizatora
 - svi elementi moraju biti inicijalizirani

```
'implicitno New(1,1)  
Dim grid As Long(,) = {{1, 2}, {1, 3}}
```



Stvaranje niza izračunske veličine

- Veličina niza ne mora biti poznata za vrijeme kompajliranja
 - dozvoljen ispravan cjelobrojni izraz
 - ne smije se koristiti inicijalizator
 - brzina pristupa elementima ostaje ista

'veličina niza određena za
vrijeme oblikovanja

```
Dim row As Long() = New Long(3) {}
```

'veličina niza određena za
vrijeme izvođenja

```
Console.WriteLine("Unesite broj redaka : ");
```

```
Dim s As String
```

```
s = Console.ReadLine()
```

```
Dim size As Integer = Integer.Parse(s)
```

```
Dim row1 As Long() = New Long(size) {}
```

Ispravna/pogrešna inicijalizacija niza

`Dim row As Long() = New Long() _
{Console.ReadLine, Console.ReadLine}` ✓

`Dim row As Long() = New Long(1) {}
row = {2,3}` ✗

`Dim row As Long() = New Long(1){}` ✓

`Dim row As Long() = New Long(1){2}` ✗

`Dim row As Long() = New Long(1){2,3}` ✓

`x = Console.ReadLine()
Dim row As Long() = New Long(x){1,2}` ✗

Kopiranje varijabli niza

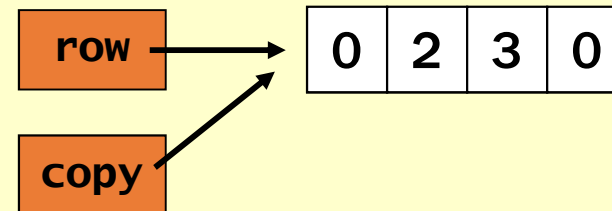
- Kopiranje varijable niza kopira sadržaj varijable (memorijska adresa)
 - ne kopira se primjer niza (podaci)
 - dvije varijable niza mogu upućivati na isti primjer niza

```
Dim row As Long() = New Long(3) {1, 2, 3, 4}  
Dim copy As Long()
```

```
copy = row
```

```
row(0) = 0  
copy(3) = 0
```

```
Console.WriteLine("copy(0)={0}, row(3)={1}", _  
    copy(0), row(3))
```



copy(0)=0, row(3)=0

Is test i Nothing

- **Is** operator vraća **True** ako dvije varijable niza pohranjuju isto upućivanje
- **Nothing** – varijabla niza ne sadrži upućivanje

```
Dim row As Long() = New Long(3) {1, 2, 3, 4}  
Dim copy As Long() = row
```

```
Console.WriteLine(row Is copy)  
row = Nothing  
Console.WriteLine(row Is copy)
```

True
False

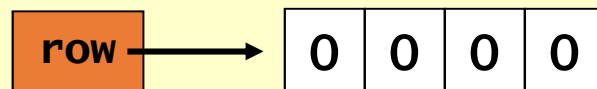
Korištenje niza

- `System.Array` klasa
- Svojstva niza
- Postupci niza
- Vraćanje niza iz procedure
- Prosljeđivanje niza kao parametra
- Plitko i dubinsko kopiranje
- Korištenje `For Each` strukture sa nizovima

Svojstva niza

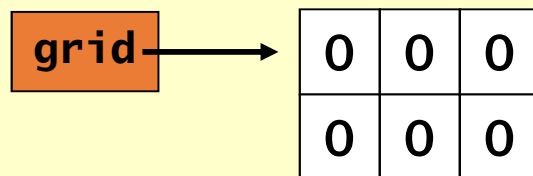
- **Rank** svojstvo
 - samo za čitanje
 - određuje broj dimenzija primjera niza
- **Length** svojstvo
 - samo za čitanje
 - određuje ukupnu dužinu primjera niza

```
Dim row As Long() = New Long(3) {}  
Console.WriteLine("Rank={0}, Length={1}", _  
                  row.Rank, row.Length)
```



Rank=1, Length=4

```
Dim grid As Long(,) = New Long(1,2) {}  
Console.WriteLine("Rank={0}, Length={1}", _  
                  grid.Rank, grid.Length)
```



Rank=2, Length=6

Postupci niza

- Najčešće korišteni postupci niza
 - **Clear**
 - postavlja određeni skup elemenata na 0 ili **Nothing**
 - **Sort**
 - sortira elemente niza reda 1
 - **Clone**
 - kreira kopiju niza
 - **GetLength**
 - vraća dužinu određene dimenzije
 - provjera granica više-dimenzionalnog niza
 - **GetUpperBound**
 - vraća najviši indeks određene dimenzije
 - **IndexOf**
 - vraća indeks prvog pojavljivanja traženog elementa
 - **BinarySearch**
 - izvodi binarno pretraživanje niza

Vraćanje niza iz procedure

- Moguće deklarirati proceduru koja vraća niz
 - nije određena veličina niza kojeg se vraća

```
Sub Main()  
    Dim row As Long() = CreateRow()  
End Sub  
  
Function CreateRow() As Long()  
    Return New Long() {1, 2, 3, 4}  
End Function
```

Prosljeđivanje niza kao parametra ⁽¹⁾

- Prosljeđivanjem **ByVal** parametar niza je kopija varijable niza
 - ne kopija primjera niza
 - mijenja se originalni primjer niza

```
Sub Main()  
    Dim row As Long() = {1, 2, 3, 4}  
    ChangeRow(row)  
    Console.WriteLine(row(0))  
End Sub  
Sub ChangeRow(ByVal array As Long())  
    array(0) = 0  
End Sub
```

0

Prosljeđivanje niza kao parametra (2)

- Nije moguće pridružiti upućivanje na drugi primjer niza.

```
Sub Main()  
    Dim row As Long() = {1, 2, 3, 4}  
    TryChangeReference(row)  
    Console.WriteLine(row(0))  
End Sub  
Sub TryChangeReference(ByVal array As Long())  
    array = Nothing  
End Sub
```

1

Prosljeđivanje niza kao parametra (3)

- Prosljeđivanjem **ByRef** moguće promijeniti ili poništiti upućivanje.

```
Sub Main()  
    Dim row As Long() = {1, 2, 3, 4}  
    TryChangeReference(row)  
    Console.WriteLine(row(0))  
End Sub  
Sub TryChangeReference(ByRef array As Long())  
    array = Nothing  
End Sub
```

Object reference not set to an instance of an object.

Prosljeđivanje niza kao parametra (4)

- Prosljeđivanje sa `Array.Clone`

- plitko kopiranje
 - kopira se upućivanje a ne sami objekt

```
Sub Main()  
    Dim row As Long() = {1, 2, 3, 4}  
    ChangeRow(row.Clone)  
    Console.WriteLine(row(0))  
End Sub  
Sub ChangeRow(ByVal array As Long())  
    array(0) = 0  
End Sub
```

1

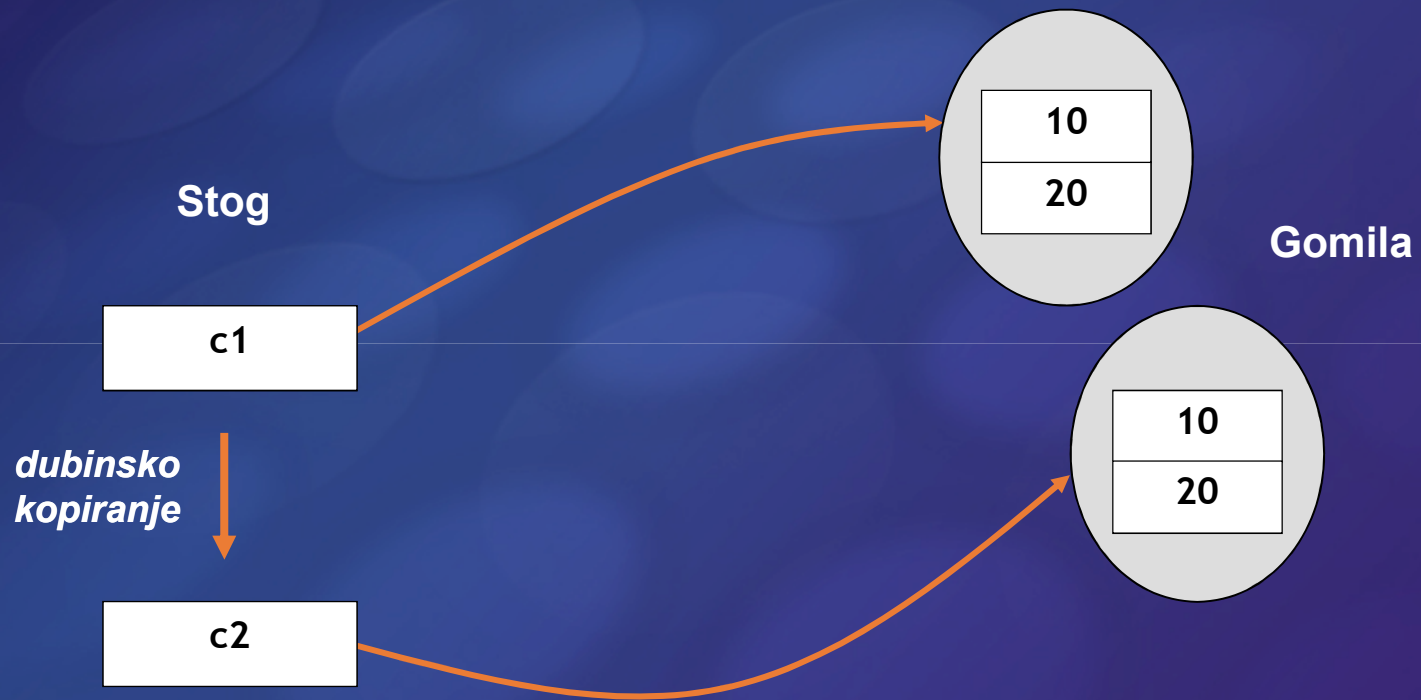
Plitko kopiranje (*shallow copy*)

```
Sub Main()  
  Dim c1 As Point() = New Point(0) {}  
  Dim c2 As Point()  
  Dim pt As New Point  
  pt.X = 10 : pt.Y = 20  
  c1(0) = pt  
  c2 = c1.Clone  
  Console.WriteLine(c1 Is c2) 'False  
End Sub
```

```
Public Class Point  
  Public X As Integer  
  Public Y As Integer  
End Class
```



Dubinsko kopiranje (*deep copy*)



Korišćenje niza sa For Each

- Apstraktni način kruženja kroz niz.

- bez For Each

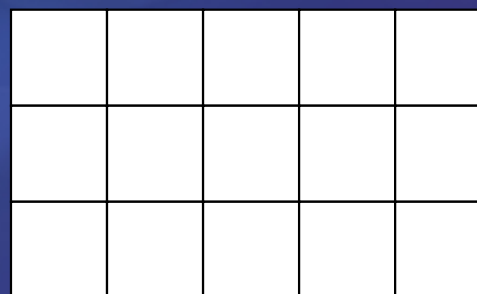
```
Sub Main()  
    Dim row As Long() = {1, 2, 3, 4}  
    Dim cnt As Integer  
    For cnt = 0 To row.Length - 1  
        Console.WriteLine(row(cnt))  
    Next  
End Sub
```

- sa For Each

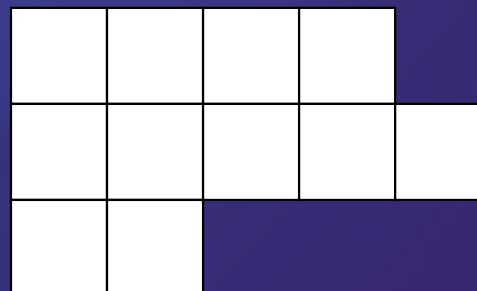
```
Sub Main()  
    Dim row As Long() = {1, 2, 3, 4}  
    Dim item As Long  
    For Each item In row  
        Console.WriteLine(item)  
    Next  
End Sub
```


Nazupčani nizovi

- Više-dimenzionalni niz
 - pravokutni
 - isti broj elemenata u svim dimenzijama



- nazupčani
 - različiti broj elemenata



Sortiranje i pretraživanje

- Sortiranje nizova
 - algoritmi sortiranja
- Pretraživanje nizova
 - algoritmi pretraživanja

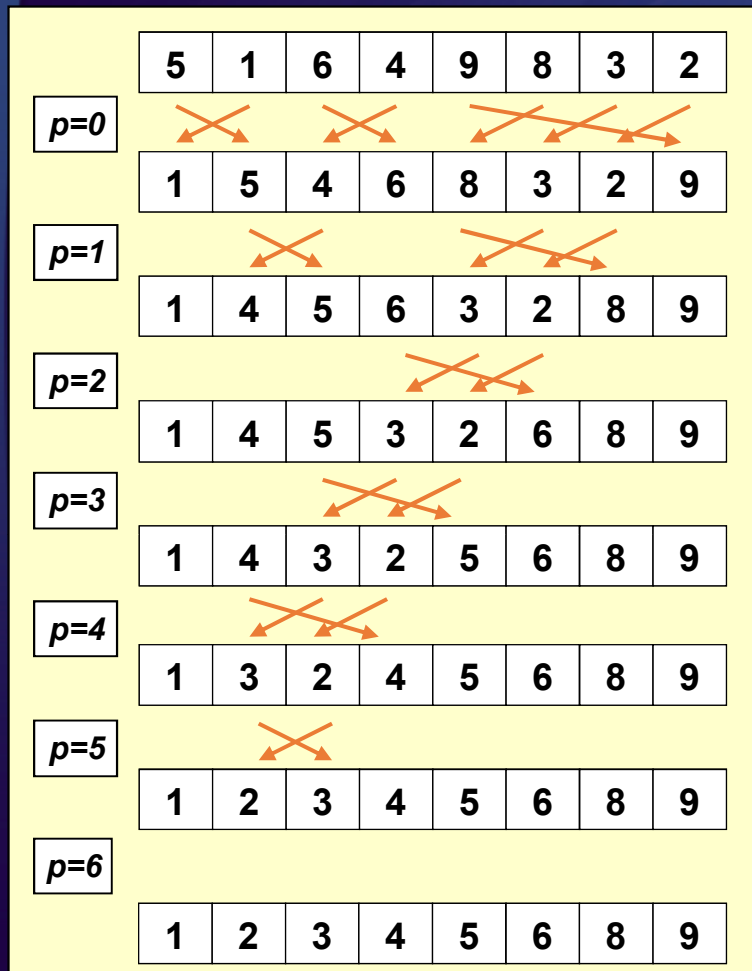
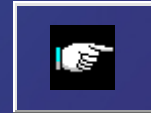
Sortiranje nizova

- Postoji niz algoritama za sortiranje nizova.
 - mjehuričasto, selekcijsko, umetanjem, brzo, prebrojavajuće ...
- Različiti algoritmi se različito ponašaju zavisno o broju stavki niza te broju ponavljanja istih stavki.
 - **ne postoji algoritam jednako dobar za sve slučajeve .**
- Sortiranje
 - interno
 - podaci učitani u memoriju
 - jednako brz pristup svim stavkama
 - vanjsko
 - podaci na traci ili disku
 - slijedni pristup stavkama
 - više vremena se potroši na čitanje i zapisivanje podataka na medij nego na samo sortiranje
 - Vrijeme izvođenja algoritma sortiranja
 - mjehuričasto - N^2 za sortiranje N stavki
 - brzo – $N \log N$ za sortiranje N stavki
 - ne mora biti dobro za mali N ili kod učestalog ponavljanja stavki

Mjehurićasto sortiranje ⁽¹⁾

- Manje vrijednosti elemenata niza postepeno prodiru prema vrhu liste, poput mjehurića zraka u vodi, a veće vrijednosti propadaju prema dnu liste.
 - tehnika zahtijeva broj prolazaka proporcionalan s N^2 , gdje je N broj stavki, što je loš rezultat.
 - u svakom prolazu se uspoređuju parovi susjednih elementa.
 - ako su vrijednosti para u porastu (ili su jednake), program ostavlja elemente na svojim mjestima.
 - ako su vrijednosti u padajućem slijedu, elementi para zamjenjuju mjesta
- Radi dobro sa kratkim listama stavki i sa djelomično sortiranim listama.

Mjehurićasto sortiranje (2)



- algoritam je brz za djelomično sortirani niz ili za mali broj stavki
- ako je u nizu veliki broj stavki izvan bilo kojeg reda, nije optimalan izbor

```
'algoritam mjehurićastog sortiranja
Private Sub BubbleSort(ByVal array As Integer())
    Dim p, i As Integer
    For p = array.GetUpperBound(0) To 1 Step -1
        For i = 0 To p - 1
            If array(i) > array(i + 1) Then Swap(array, i)
        Next
    Next
End Sub

'zamjena pozicija susjednih elemenata niza
Private Sub Swap(ByVal swapArray As Integer(), _
    ByVal lowerIndex As Integer)
    Dim temp As Integer = swapArray(lowerIndex)
    swapArray(lowerIndex) = swapArray(lowerIndex + 1)
    swapArray(lowerIndex + 1) = temp
End Sub
```

Selekcijsko sortiranje

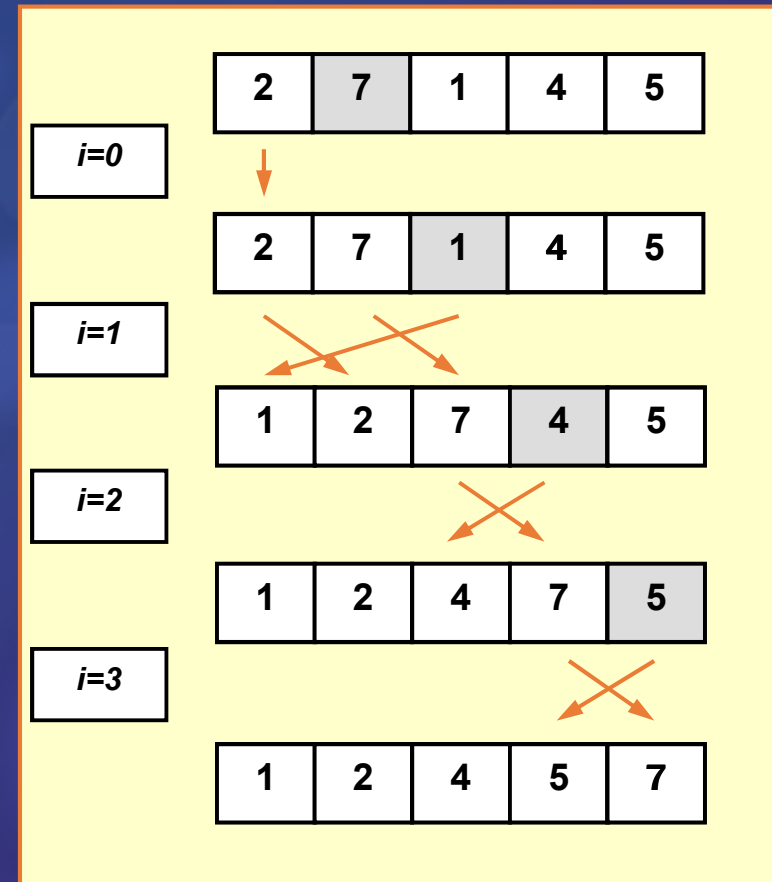


- Pretraživanje niza kako bi se pronašla stavka s najmanjom vrijednošću
 - takva stavka se postavlja na prvo mjesto niza
 - postupak se ponavlja sve dok sve stavke niza nisu sortirane
 - vrijeme izvođenja zanemarivo ovisi o stupnju sortiranosti niza
 - otprilike jednako vrijeme je potrebno za sortiranje djelomično sortiranog niza i u cijelosti sortiranog niza.
 - koristi se za male nizove, i velike nizove malog raspona vrijednosti

```
Public Sub SelectionSort(ByVal numbers As Integer())  
    Dim i, j, n, min, temp As Integer  
    n = numbers.GetUpperBound(0)  
    For i = 0 To n-1  
        min = i  
        For j = i + 1 To n  
            If numbers(j) < numbers(min) Then  
                min = j  
            End If  
        Next  
        temp = numbers(i)  
        numbers(i) = numbers(min)  
        numbers(min) = temp  
    Next  
End Sub
```

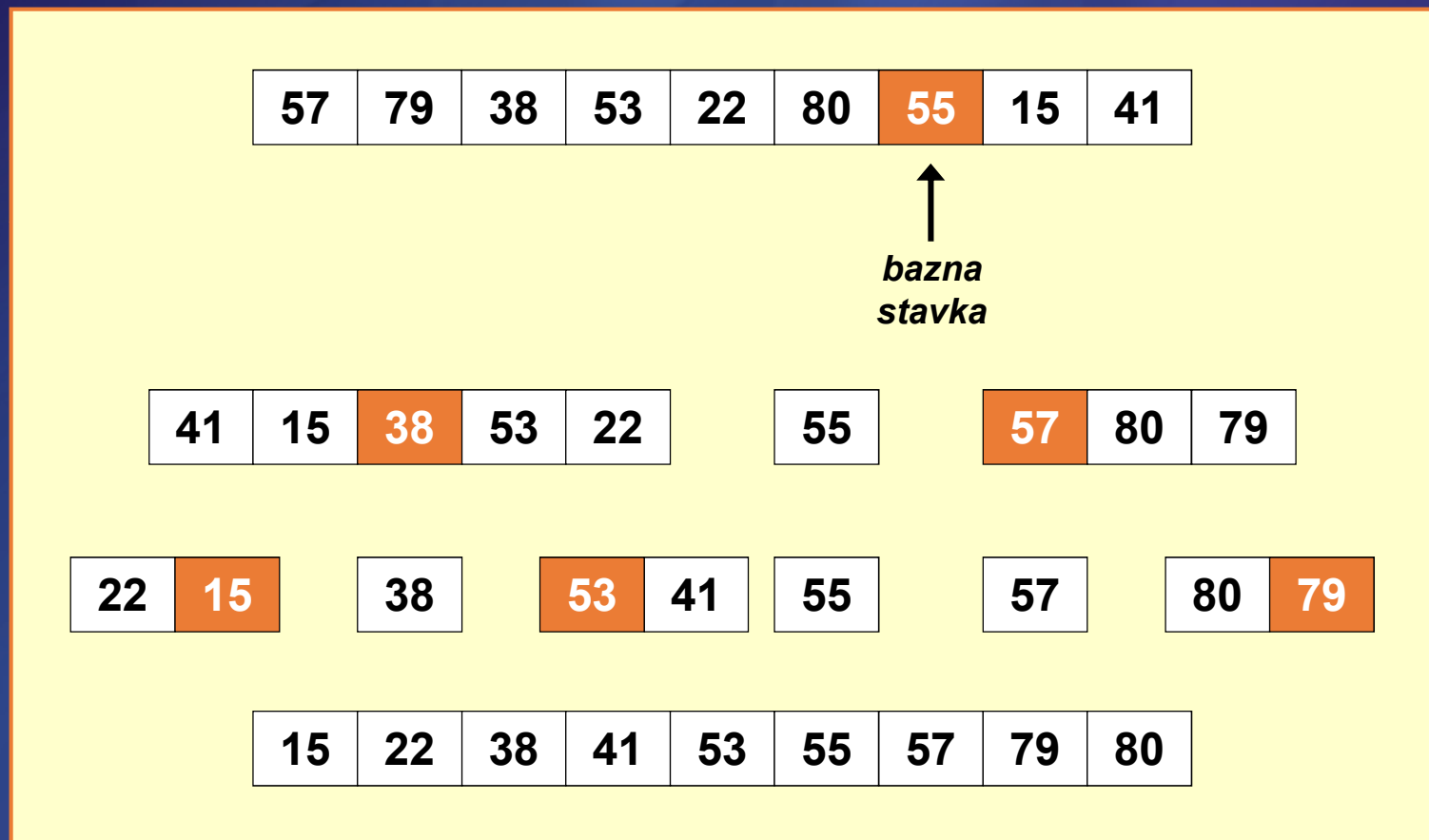

Sortiranje umetanjem

- Razmatra odjednom samo jedan element i ubacuje ga na prikladno mjesto između već razmotrenih elemenata, ostavljajući ih sortiranim.
- najprije pravi prostor za element koji treba biti umetnut pomicanjem većih elemenata jednu poziciju u desno a zatim ubacuje element na ispražnjeno mjesto.
- elementi lijevo od tekućeg indeksa su u sortiranom redoslijedu ali ne i na svojoj konačnoj poziciji jer mogu biti pomaknuti kako bi napravili mjesto manjem elementu koji je pronađen kasnije.



Brzo sortiranje (Quicksort)

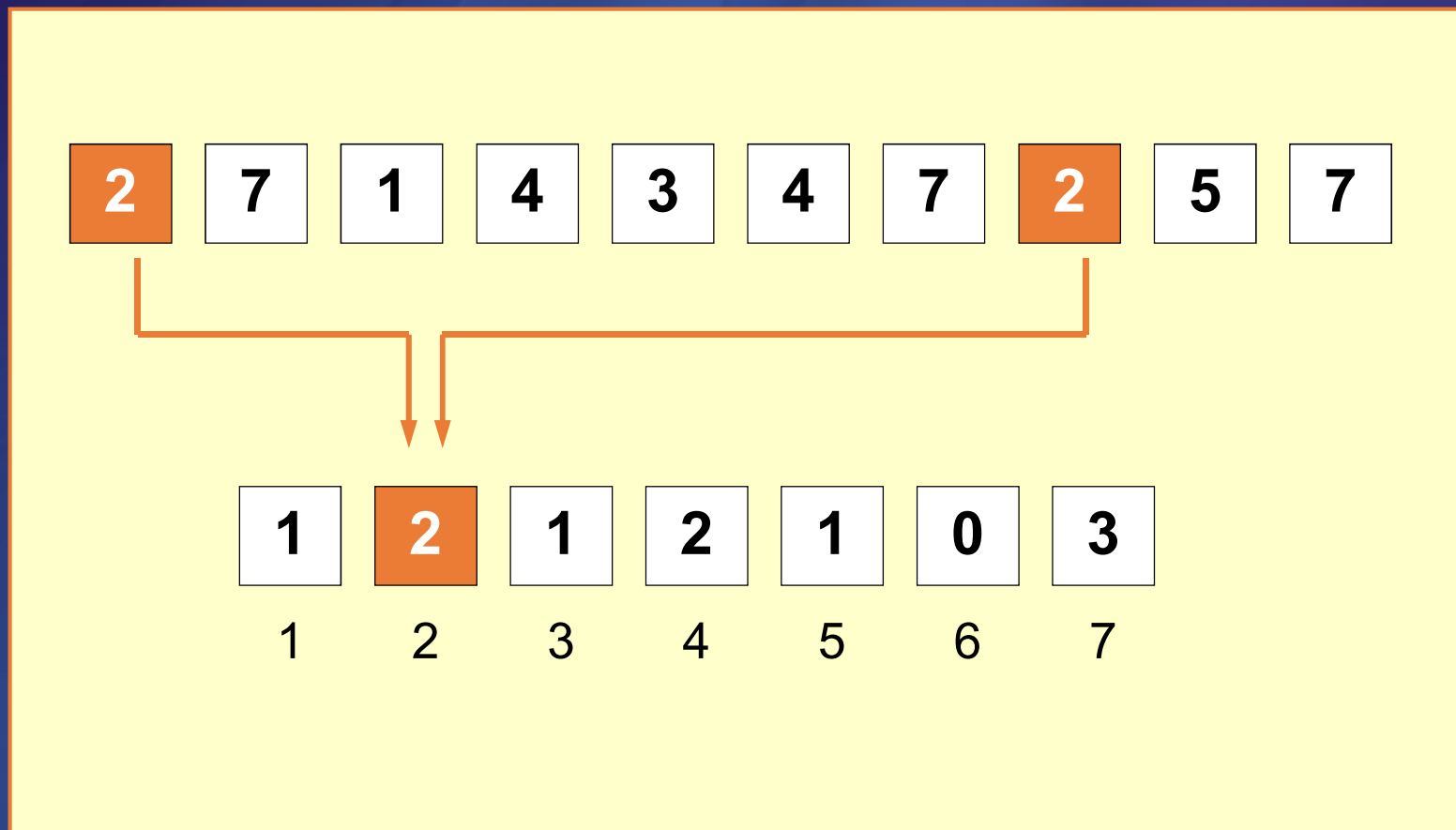
- koristi rekurzivne pozive
- implementiran u `Array.Sort`



Prebrojavajuće sortiranje



- Broji ponavljanja iste stavke



Linearno pretraživanje

- Uspoređuje vrijednost svakog elementa niza sa ključem pretraživanja.
- koristi se kad niz nije sortiran

```
Sub Main()  
    Dim array As Integer() = {1, 5, 7, 9, 2}  
    Console.WriteLine(LinearSearch(array, 7))  
End Sub  
  
Public Function LinearSearch(ByVal array() As Integer, _  
                             ByVal key As Integer) As Long  
    Dim cnt As Integer  
    For cnt = 0 To array.GetUpperBound(0)  
        If array(cnt) = key Then  
            Return cnt  
        End If  
    Next cnt  
    Return Long.MinValue  
End Function
```

Binarno pretraživanje



- Pretraživanje sortiranih nizova.
- Uspoređuje ključ pretraživanja sa stavkom u sredini liste.

| | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 11 | 15 | 22 | 37 | 38 | 41 | 53 | 57 | 59 | 70 | 79 | 98 | 99 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|



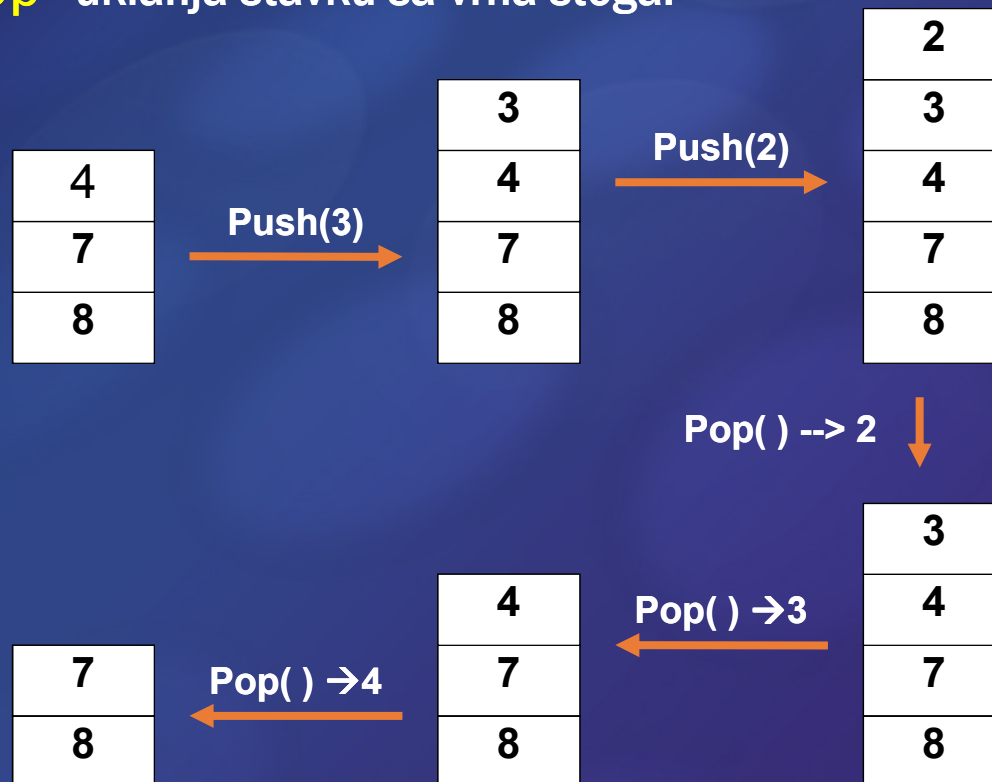
srednja
stavka

| | | | | | |
|----|----|----|----|----|----|
| 57 | 59 | 70 | 79 | 98 | 99 |
|----|----|----|----|----|----|

| | | |
|----|----|----|
| 79 | 98 | 99 |
|----|----|----|

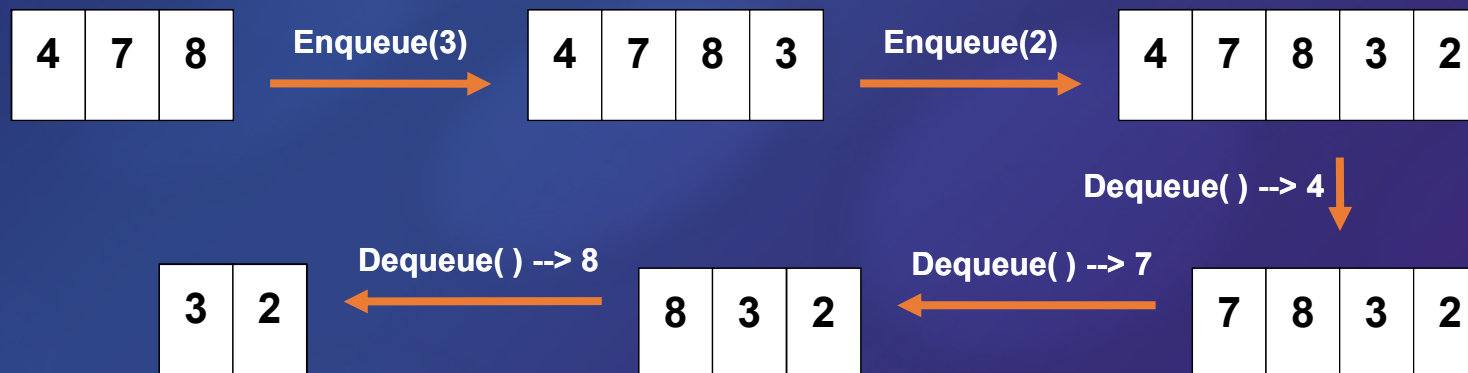
Podatkovna struktura Stack

- **LIFO** (*last in, first out*) zbirka
 - posljednje gurnuta stavka je prva istisnuta stavka
 - procedure za rukovanje stogom
 - **Push** - dodaje novu stavku na vrh stoga
 - **Pop** - uklanja stavku sa vrha stoga.



Podatkovna struktura Red (Queue)

- **FIFO** (*first in, first out*) zbirka.
 - stavke se dodaju na jedan kraj a uklanjaju sa drugog.
 - poput reda kupaca koji čekaju da budu posluženi.
 - Operacije ubacivanja i uklanjanja stavki iz reda:
 - Enqueue
 - Dequeue



Stvaranje vlastitih vrijednosnih tipova

- Uvod
- Strukturni tipovi
 - definiranje strukturnih tipova
 - korištenje strukturnih tipova
 - memorijsko predstavljanje
 - kopiranje varijabli struktura
 - funkcionalni članovi struktura
- Razlike i sličnosti struktura i klasa
- Nabrajanja

Uvod



| Primitivni tipovi | |
|-------------------|----------|
| Boolean | Single |
| Byte | Double |
| Int16 | Decimal |
| Int32 | DateTime |
| Int64 | Char |

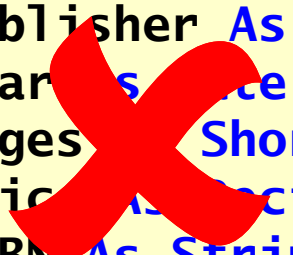
Strukturni tipovi (1)

- **Skalarne varijable**

- jednostruka vrijednost
- problem sa predstavljanjem stvarnih objekata
 - Primjer, objekt knjiga

- naslov
- autor
- izdavač
- datum izdavanja
- broj stranica
- kratki opis
- poglavlja
- cijena
- ISBN
- ...

```
Sub Main()  
    Dim Title As String  
    Dim Author As String  
    Dim Publisher As String  
    Dim Year As Integer  
    Dim Pages As Short  
    Dim Price As Decimal  
    Dim ISBN As String  
End Sub
```



- **Idealno samo jedna varijabla, npr. Book koja uključuje sve parametre.**

- rješenje problema: definiranje vlastitog vrijednosnog tip korištenjem struktura

Strukturni tipovi (2)

- Strukture se koriste za stvaranje objekata koji se ponašaju poput ugrađenih vrijednosnih tipova.
- Strukture su korisne kada se želi da jedna varijabla drži nekoliko povezanih dijelova informacije.
- Strukture su spremnici tipova.
 - moguće povezati podatke različitih tipova
- U .NET Framework-u, jednostavni tipovi podataka poput `Integer`, `Short`, `Boolean`, `Double`, `Char` su ugrađene strukture.

Definiranje strukturnih tipova

- Predložak strukture:

```
[accessibility] Structure structurename  
    ...  
End Structure
```

- dostupnost: **Private**, **Public**, **Friend** i **Protected**
 - podrazumijevana dostupnost **Public**
 - **Private** - nedostupna izvan modula
 - **Friend** – dostupna i izvan modula ali unutar projekta
 - **Public** – dostupna i u drugim projektima rješenja
 - **Protected** – jedino u klasama
- sadrže barem jednu varijablu, podatkovni član

Primjer 1: Strukturni tip - Box

```
Public Structure Box
    Dim Length As Integer    'dužina
    Dim Height As Integer    'visina
    Dim Breadth As Single    'širina
End Structure
```

- Potrebno deklarirati svaku varijablu i odrediti njezinu dostupnost
 - Dim = Public !!!
 - Private
- Length, Height, Breadth su članovi ili polja strukture
- Nije dozvoljeno inicijaliziranje članova strukture unutar same strukture
 - samo preko varijable deklarirane strukturnim tipom

Korištenje strukturnog tipa

```
Sub Main()
```

```
    Dim myBox As Box      'deklaracija varijable  
    myBox.Length = 10     'pridruživanje vrijednosti
```

```
    myBox.Height = 20  
    myBox.Breadth = 3.5
```

```
    'dohvat vrijednosti
```

```
    Console.WriteLine("Volumen je: {0}", _  
        myBox.Length * myBox.Height * myBox.Breadth)
```

```
End Sub
```

700

myBox – varijabla strukture

- deklarira se bez **New**
- pristup elementima navođenjem imena varijable, točke “.” i imena člana

```
Dim myBox As Box  
myBox.|
```



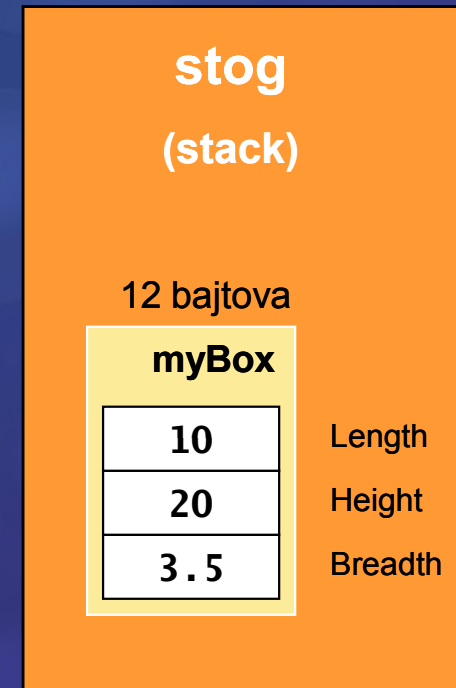
A screenshot of an IDE's IntelliSense dropdown menu. The menu is open, showing a list of members for the 'myBox' variable. The first item, 'Breadth', is highlighted with a blue background. Below it are 'Equals', 'GetHashCode', 'GetType', 'Height', 'Length', and 'ToString'. Each item is preceded by a small icon: a blue diamond for properties (Breadth, Height, Length) and a purple diamond for methods (Equals, GetHashCode, GetType, ToString).

- Breadth
- Equals
- GetHashCode
- GetType
- Height
- Length
- ToString

Memorijsko predstavljanje

```
Public Structure Box
    Dim Length As Integer
    Dim Height As Integer
    Dim Breadth As Single
End Structure
```

```
Sub Main()
    Dim myBox As Box
    myBox.Length = 10
    myBox.Height = 20
    myBox.Breadth = 3.5
End Sub
```



- Varijabla strukture “troši” najmanje zbroj memorijskih zahtjeva njenih podatkovnih članova.

Kopiranje varijabli struktura

- Kopiranje varijable
 - pridruživanjem drugoj varijabli
 - proslijeđivanjem u proceduru **ByVal**
 - kada se vraća iz funkcije

```
Sub Main()  
    Dim box1, box2 As Box  
    box1.Length = 10  
    box1.Height = 20  
    box1.Breadth = 3.5  
    box2 = box1  
    box2.Length = 20  
    PrintVolume(box1)  
End Sub  
Sub PrintVolume(ByVal box3 As Box)  
    Console.WriteLine("Volumen je: {0}", _  
        box3.Length * box3.Height * box3.Breadth)  
End Sub
```

700

Primjer 2: Strukturni tip - Person

```
Private Structure Person
    Public FirstName As String
    Public LastName As String
    Public Address As String
    Public Birthday As Date
End Structure

Sub Main()
    Dim person1 As Person
    person1.FirstName = "Ivan"
    person1.LastName = "Ivić"
    person1.Birthday = #4/27/1970#
    'ispis imena osobe
    Console.WriteLine(person1.FirstName & _
                        " " & person1.LastName)
End Sub
```

Funkcionalni članovi struktura

```
Private Structure Person
    Public FirstName As String
    Public LastName As String
    Public Address As String
    Public Birthday As Date
    Public Function CompleteName() As String
        Return FirstName & " " & LastName
    End Function
End Structure

Sub Main()
    Dim person1 As Person
    person1.FirstName = "Ivan"
    person1.LastName = "Ivić"
    person1.Birthday = #4/27/1970#
    'ispis imena osobe
    Console.WriteLine(person1.CompleteName)
End Sub
```




- Address
- Birthday
- CompleteName**
- Equals
- FirstName
- GetHashCode
- GetType
- LastName
- PhoneNumber
- ToString

Primitivni tipovi - strukture

- Svi primitivni tipovi su uređeni kao strukture.
- Primjer tip **Char**

```
Dim letter As Char  
Char.
```



A screenshot of a Visual Basic IDE showing the members of the Char structure. The members are listed in a scrollable list box, each preceded by a small icon. The icons are purple diamonds for methods (IsSurrogate, IsSymbol, IsUpper, IsWhiteSpace, Parse, ToLower, ToString, ToUpper) and cyan diamonds for properties (MaxValue, MinValue). The list box has a scrollbar on the right side.

- IsSurrogate
- IsSymbol
- IsUpper
- IsWhiteSpace
- MaxValue
- MinValue
- Parse
- ToLower
- ToString
- ToUpper

Strukture i klase

- Sličnosti

- imaju podatkovne članove
- imaju funkcionalne članove
- imaju polja, svojstva, postupke i događaje
- dozvoljavaju konstruktor

- Razlike

- struktura je vrijednosni tip pa se alocira na stogu a ne na upravljanoj gomili!
- varijable strukture ne koriste **New** za alokaciju memorije

Nabrajanja (Enumerations) ⁽¹⁾

- Koriste se kada varijabla može poprimiti samo specificirani skup vrijednosti.
- Definiranje nabiranja
 - ime
 - skup polja (konstanti)
 - tip podatka (**Byte** , **Short** , **Integer** i **Long**)

```
Enum Vehicle As Integer
    Crysler 'vrijednost 0
    Fiat    'vrijednost 1
    Citroen 'vrijednost 2
    Hyundai 'vrijednost 3
End Enum
```

Nabrajanja (Enumerations) (2)

- Korištenje nabrajanja

```
Module Module1
    Enum Vehicle As Integer
        Crysler = 20
        Fiat = 40
        Citroen = 50
        Hyundai = 90
        Unknown = -999
    End Enum
    Sub Main()
        Dim car As Vehicle=

    End Sub
End Module
```



- Vehicle.Citroen
- Vehicle.Crysler
- Vehicle.Fiat
- Vehicle.Hyundai
- Vehicle.Unknown

Pitanja ?



Zadatak 1: Pohrana 2D koordinata

- Potrebno je napisati program koji pohranjuje 2D koordinate (x i y) u jednodimenzionalno polje.

Zadatak 2: Igrače karte

- Podijeliti igrače karte tako da ista karta ne bude podijeljena dvaput.

Za više informacija...

- Odgovaram na mail-ove (uglavnom uvijek :-)
- jmusic@fesb.hr