

חומרי עזר שהוכנו ע"י משתתפי קורס מורים מובילים תשע"א

ניתן להשתמש בחומרים לצורך הוראה בלבד. אסור לפרסם את החומרים או לעשות בהם שימוש מסחרי כלשהו ללא קבלת אישור מראש מצוות הפיתוח

# מחרוזות

# strings

הוכן על-ידי וג'יה כבהה, תיכון כפר קרע

ניתן להשתמש במצגת להוראה ... לא ניתן לעשות שימוש מסחרי ללא רשות

# משתנה מחרוזתי

## הכרזה על משתנה מחרוזתי

```
string x;  
string a,b;
```

## הצבה במשתנה מחרוזתי

```
x="hello";
```

## קליטה למשתנה מחרוזתי

```
a=Console.ReadLine();
```

## הדפסת משתנה מחרוזתי

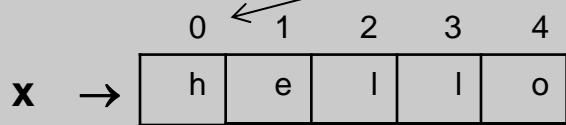
```
Console.WriteLine(x);
```

# משתנה מחרוזתי

ניתן להתייחס למחרוזת כאל מערך של תווים

```
string x,y;
```

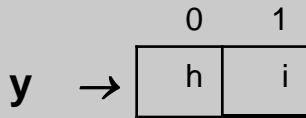
```
x="hello" ;
```



X[0]

X[3]

```
y="hi" ;
```



למחרוזת יש תכונה Length שמחזירה את אורך המחרוזת (מספר תווים)

x.Length - מחזירה 5

y.Length - מחזירה 2

# משתנה מחרוזתי

הדפס תו ראשון במחרוזת x :

```
Console.WriteLine(x[0]);
```

הדפס תו אחרון במחרוזת x :

```
Console.WriteLine(x[x.Length-1]);
```

הדפס את המחרוזת x כל תו בשורה :

```
for(i=0 ; i < x.Length ; i++)  
{  
    Console.WriteLine(x[i]);  
}
```

פעולות על מחרוזות

# שרשור מחרוזות – הפעולה +

- פעולת השרשור + בונה מחרוזת אחת משתי מחרוזות.

```
string a,b,c;
```

	a	b	c
a="abk";	abc		
b="sh";		de	
c=a+b;			abcde
b="***" + b+ "***";		***de***	

# st1.Equals(st2)

הפעולה תחזיר true אם המחרוזת st2 זהה למחרוזת st1 ו false אחרת.

```
If (st1.Equals(st2))
```

```
    Console.WriteLine("Yes");
```

```
else
```

```
    Console.WriteLine("No");
```

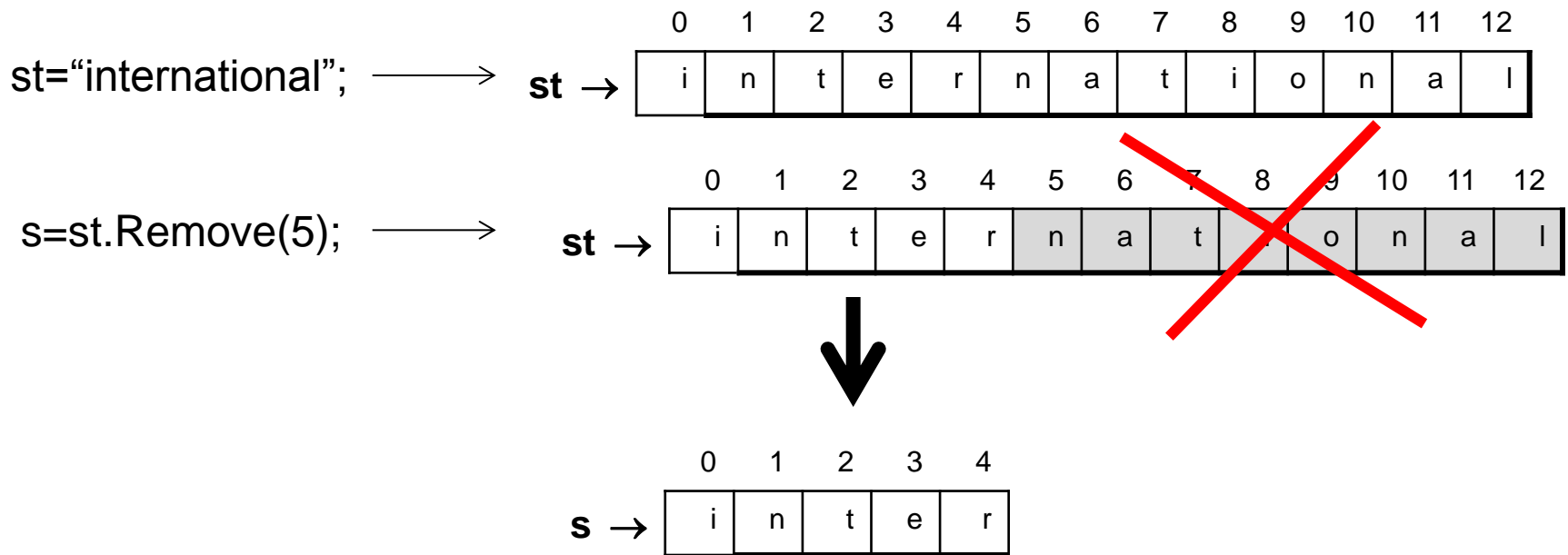
# st1.CompareTo(st2)

- הפעולה מקבלת מחרוזת st2 .
- אם st1 זהה ל st2 הפעולה תחזיר 0.
- אם st1 גדולה מ st2 הפעולה תחזיר מספר חיובי
- אם st1 קטנה מ st2 הפעולה תחזיר מספר שלילי.



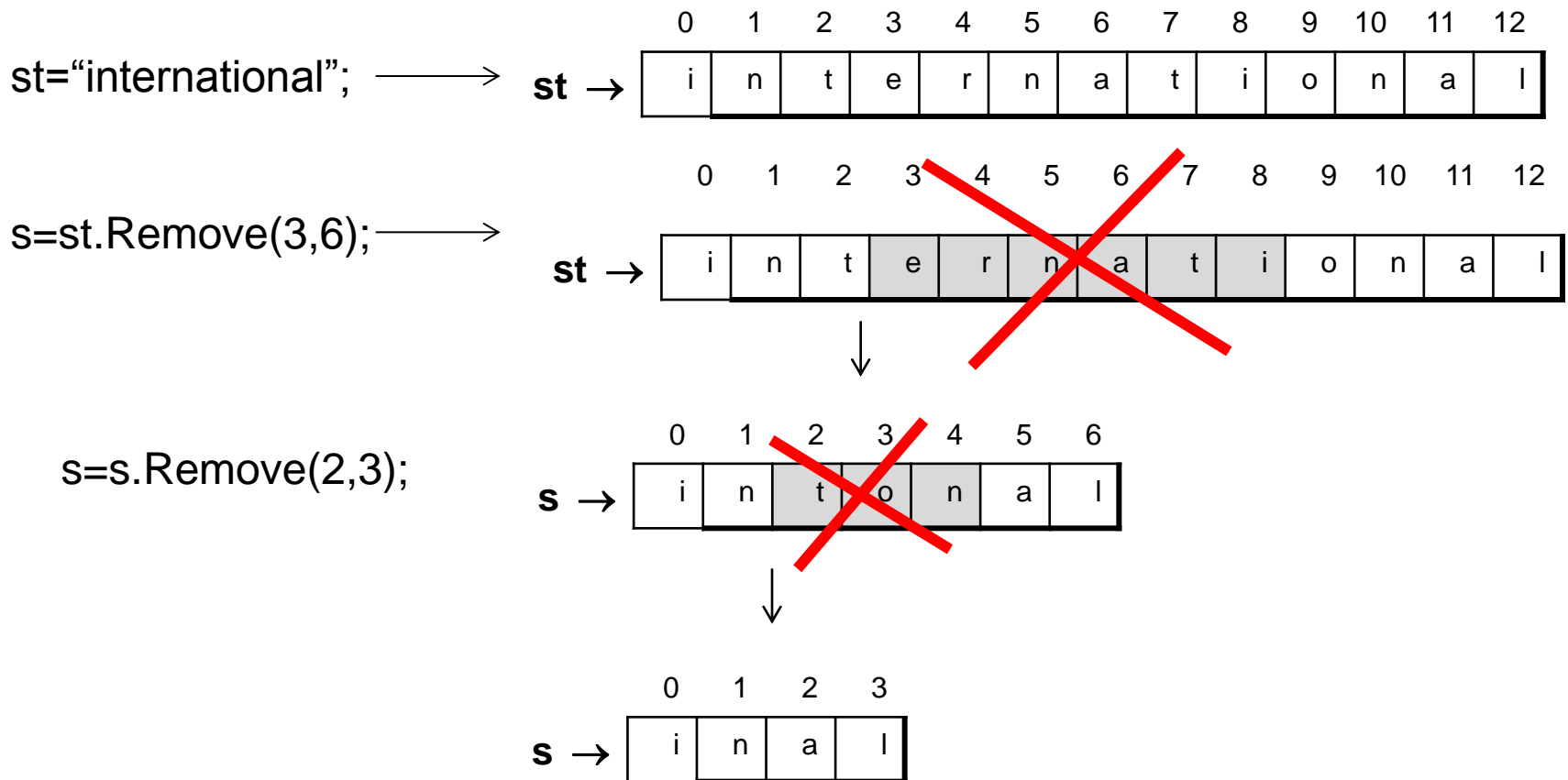
# st.Remove(k)

הפעולה מחזירה מחרוזת חדשה המורכבת מהמחרוזת st לאחר מחיקת חלק מהמחרוזת החל מהתו k עד התו האחרון.



# st.Remove(k,count)

הפעולה מחזירה מחרוזת חדשה המורכבת מהמחרוזת st לאחר מחיקת חלק מהמחרוזת החל מהתו k באורך count.



# st.Substring(k)

- הפעולה מחזירה חלק מהמחרוזת st החל מהמקום k.

st="international"; → st →

0	1	2	3	4	5	6	7	8	9	10	11	12
i	n	t	e	r	n	a	t	i	o	n	a	l

s=st.Substring(6); → st →

0	1	2	3	4	5	6	7	8	9	10	11	12
i	n	t	e	r	n	a	t	i	o	n	a	l

s=s.Substring(5);

s →

0	1	2	3	4	5	6
a	t	i	o	n	a	l

s →

0	1
a	l

# st.Substring(k,count)

- הפעולה מחזירה חלק מהמחרוזת st החל מהמקום k באורך count.

st="international"; → st →

0	1	2	3	4	5	6	7	8	9	10	11	12
i	n	t	e	r	n	a	t	i	o	n	a	l

s=st.Substring(4,6); → st →

0	1	2	3	4	5	6	7	8	9	10	11	12
i	n	t	e	r	n	a	t	i	o	n	a	l

s=st.Substring(1,3); → s →

0	1	2	3	4	5
r	n	a	t	i	o

s →

0	1	2
n	a	t

# st.IndexOf(x)

הפעולה תחזיר מספר שלם שהוא המקום שבו מתחילה המחרוזת x אם היא נמצאת במחרוזת st .  
אם x לא נמצאת בתוך st יוחזר -1 .

# st1.IndexOf(ramz))

הפעולה תחזיר מספר שלם שהוא המקום שבו  
התו ramz נמצא במחרוזת st .  
אם ramz לא נמצא יוחזר -1 .

# st1.Insert(k,st2)

הפעולה מחזירה מחרוזת חדשה לאחר הכנסת  
המחרוזת st2 למחרוזת st1 במקום k

st="abcdefg"; →

	0	1	2	3	4	5	6
st →	a	b	c	d	e	f	g

st=stInsert(3,"\*\*\*"); →

	0	1	2	3	4	5	6	7	8	9
st →	a	b	c	d	*	*	*	e	f	g

st=stInsert(1,"12"); →

	0	1		2	3	4	5	6	7	8	9	
st →	a	b	1	2	c	d	*	*	*	e	f	g

# st.Replace(st1,st2)

הפעולה מחזירה מחרוזת חדשה המורכבת מהמחרוזת st לאחר החלפת כל מופעי תת-המחרוזת **st1** בתת מחרוזת **st2**.

	<b>s</b>
s="cat don't eat cat"	cat don't eat cat
s=s.Replace("cat","dog");	dog don't eat dog
s=s.Replace("o","*");	d*g d*n't eat d*g



# string.copy(st)

הפעולה מחזירה העתק של המחרוזת st

```
string a,b,st;  
int k;  
bool bl;
```

k	bl	st	a	b	
			abcdefcd		a="abcdefcd";
				sss	b="sss";
		123			st="123";
	false				bl=a. <b>Equals</b> (b)
-1					n=a. <b>CompareTo</b> (b)
2					n=a. <b>IndexOf</b> (b)
		cdefcd			st=a. <b>Substring</b> (2)
		bcd			st=a. <b>Substring</b> (1,3)
			assefcd		a=a. <b>Replace</b> (st,b)
		asfcd			st=a. <b>Remove</b> (2,3)