

ניתן להשתמש בקבלת מספר פרמטרים לא ידוע ע"י שימוש בלמשל

ואז ניתן לשלוח מספר לא ידוע של ints לפונקציה. אפשר גם להשתמש בגירסה יותר מופשטת כמו
 params object[] args ואז אפשר לשלוח לפונקציה מספר בלתי ידוע של אובייקטים ולא רק סוג
 מסויים. ראה בדוגמה.

כך ממומשת המתודה Console.WriteLine("bla:{0}, bla{1}", num1, name)

ניתן לבנות prop למחלקת אוסף כאשר הגישה אליה תהייה בצורת מערך, באמצעות סוגריים
 מרובעים []. נבנה property מיוחד שנקרא indexer המאפשר גישה באמצעות [].

נניח שיש לנו במחלקה מערך של Person בשם pArr.

```
//Indexer
public Person this[int idx]
{
    get
    {
        return (Person)pArr[idx];
    }
    set
    {
        pArr[idx] = value;
    }
}

public Person this[string fName]
{
    get
    {
        for (int i = 0; i < Count; i++)
            if (this[i].FName == fName)
                return this[i];

        return null;
    }
}

public void PrintAll()
```

```

{
    for (int i = 0; i < Count; i++)
    {
        //((Person)pArr[i]).Print();
        //Same as:
        this[i].Print();

        Console.WriteLine("/*-/*-/*-/*-/*-/*-/*-/*-");
    }
}

static void Main(string[] args)
{
    PersonCollection pcol = new PersonCollection();
    ...
    pcol[2] = new Person();

    pcol.PrintAll();

    pcol["Eran"].Print();
}

```

CHANGES THROUGH FOREACH

בעקרון בלולאת FOREACH ניתן רק לקרוא מתוך האוסף ולא ניתן לעשות שינויים על האוסף!

אבל יש לשים לב כי ניתן לשנות את המאפיינים של האובייקט שעליו רצים ע"י foreach

[02_2 making changes through Foreach]

```

foreach (Student item in myClass)
{
    Console.WriteLine(item.Name);
    //item = new Student(); //THIS IS AN ERROR BECAUSE IENUMERATOR IS
    READ ONLY BUT THE PROPERTIES CAN BE CHANGED!!!

    if (item.Id % 2 == 0)
    {
        //HERE WE CAN CHANGE THE NAME(FOR EXAMPLE) OF THE REAL COLLECTION
        !
        item.Name += " the graet!";
    }
}

```