

DATA STRUCTURES

Lab – II

Metotlar, Diziler ve Hata Ayıklama

Doç. Dr. Aybars UĞUR

İçerik

- Debugging konusunun kısaca anlatımı. (10 dk.)
- F11, F10 ve Durma Noktası denemeleri yapılması ve gözlemlerin not edilmesi. (10 dk.)
- Örnek2'nin anlatımı (Metotlar)
Öğrenciler tarafından Ltopla metodunun yazılması, işletimi. (20 dk.)
- matris.cs dosyasının sisteme yüklenmesi.
- Erken bitirenler, C# debugging video izleyebilir.

Debugging (Hata Ayıklama) Olanakları

The screenshot shows the Microsoft Visual Studio IDE with a C# console application. The code is as follows:

```
namespace ConsoleApplication8
{
    class Program
    {
        static void Main(string[] args)
        {
            double[] değerler = { 1.2, 1.3, 1.5, -1, -25 };
            double sonuç = topla(değerler);
            Console.WriteLine(sonuç);
        }

        static int topla(double[] dizi)
        {
            double toplam = 0;
            int elemansayısı = 0;
            for (int i = 0; i < dizi.Length; ++i)
            {
                toplam += dizi[i];
                elemansayısı++;
            }
            return elemansayısı;
        }
    }
}
```

Annotations and labels in the image:

- Debug Menüsü**: Points to the **Debug** menu in the top toolbar.
- İşletilecek sonraki komut**: Points to the **Next** button in the toolbar.
- Durma Noktaları**: Points to the left margin where breakpoints are set.
- Breakpoint 1** and **Breakpoint 2**: Two red dots on the left margin, one at the `return elemansayısı;` line and one at the `Console.WriteLine(sonuç);` line.
- Debug Araç Kutusu**: Points to the **Debug Tools** window on the right.
- Hata! Toplam döndürülmeli**: Points to the `return elemansayısı;` line, which is circled in red.

The **Autos** window at the bottom shows the following data:

Name	Value	Type
args	{string[0]}	string[]
değerler	null	double[]

Debugging

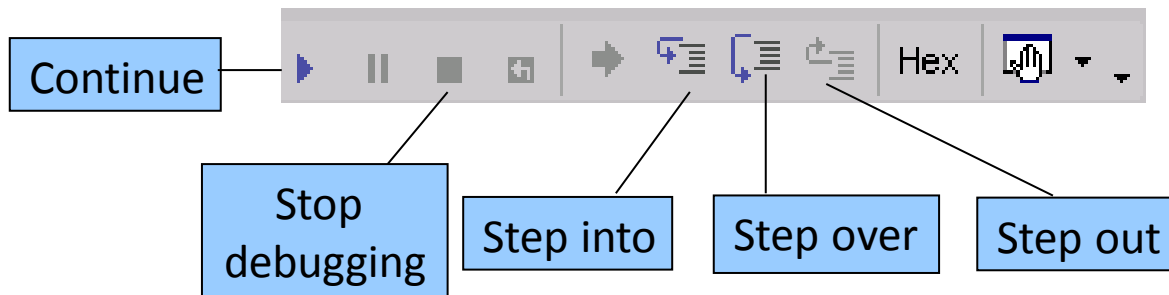
- Program adım adım işletilebilir.
- Her bir adımda, değişkenlerin durumu izlenebilir.
- Durma Noktaları (Breakpoint'ler), programın en solundaki gri çubuk üzerine, ilgili satırın hizasında tıklanarak eklenebilir. İki durma noktası arasındaki kodun adım adım işletilmesine gerek kalmaz. Böylece hataların olabileceği bölgelere odaklanılır.

Debug Menüsü ve Araç Kutusu

- Step Into (F11) : İlgili metodun içine girer. Farklı bir yerde de olsa ilgili sayfa açılır ve oradaki ilk deyimden itibaren hata ayıklamaya devam edilir.
- Step Over (F10) : Tüm metod işletilir. Metodun içine girilmez. Kapsam içinde işleme devam edilir.
- Step Out (Shift+F11) : Bulunulan metottan çıkılarak hata ayıklamaya devam edilir.

Debug Menüsü ve Araç Kutusu

- Continue (F5)
 - Bir sonraki durma noktasına kadar olan tüm komutları işler, yoksa program sonuna kadar işletilerek bitirilir.
- Stop Debugging (Shift + F5) :
 - Debug işlemi, program sonuna kadar işletilmeden durdurulur (kesilir).



Debug Panelleri ve Watch

Değişkenlerin, kontrollerin ve nesnelerin değerlerini izlemek ve incelemek için Debug – Windows alt menü seçeneği kullanılır.

Locals Penceresi (paneli) : İçinde olunan kapsam alanındaki tüm değişkenlerin tiplerini ve o anki bilgilerini (değerlerini) listeler.

Autos Penceresi (paneli) : İşletilmekte olan satırdaki ve bir öncesindeki ifadelerdeki değişkenlerin bilgilerini gösterir.

Watch Penceresi (paneli) : istediğimiz değişkenlerin isimlerini panele yazarak değerlerini gözlemleyebiliriz.

Quickwatch seçeneği ile değerleri anlık olarak görülebilir.

Sağ tuş ile çıkacak menü üzerinden, ilgili panelden izlenmesine gerek kalmayan değişkenler silinebilir veya eklenebilir.

Önemli

- Kod yazımına devam etmek için hata ayıklamayı sonlandırmak gerekir. Stop Debugging.
- Debug sırasında yeni Durma Noktası eklenebilir.
- İşletilecek deyim olmayan satırlara Durma noktası konulamaz.

Örnek 1:

Programın Adım Adım İzlenmesi

Aşağıdaki örneği yazınız ve bahsedilen Debug özelliklerini deneyiniz. Gerektiğinde Araştırma Görevlilerine sorunuz. Metot içerisinde dizi elemanlarından birisini değiştiriniz. Metoda bir de değer tipinde parametre ekleyip çağırınız. Debug ile değişkenleri izleyiniz. Bir kağıda bulgularınızı, sonuç ve deneyimlerinizi yazınız.

```
class Program
{
    static void Main(string[] args)
    {
        double[] değerler = { 1.2, 1.3, 1.5, -1, -25 };
        double sonuç = topla(değerler);
        Console.WriteLine(sonuç);
    }

    static double topla(double[] dizi)
    {
        double toplam = 0; int elemansayısı = 0;
        for (int i = 0; i < dizi.Length; ++i)
        {
            toplam += dizi[i];
            elemansayısı++;
        }
        return toplam;
    }
}
```

Örnek 2: Metotlar

- 1) Aşağıdaki matrisi Main metodunda tanımlayıp oluşturunuz.

$$\text{Matris} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 2 & 5 & 7 & 11 \\ 13 & 17 & 19 & 23 \end{bmatrix}$$

- 2) Verilen bir matrisin tüm elemanlarının toplamını bulduran metodu yazınız ve Main metodundan çağırınız.

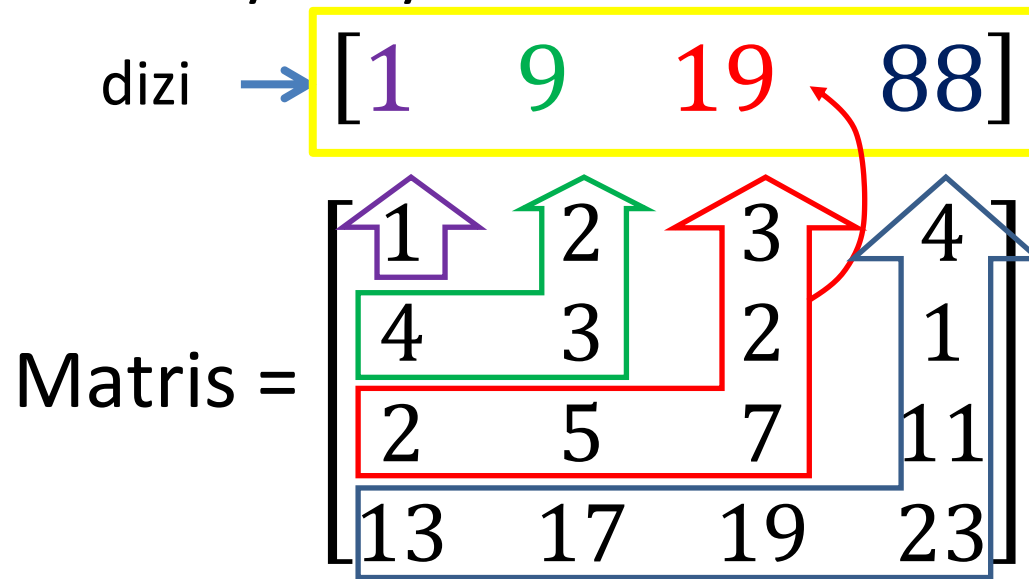
Program

```
class Program
{
    static void Main(string[] args)
    {
        int[,] matris = { { 1, 2, 3, 4 }, { 4, 3, 2, 1 }, { 2, 5, 7, 11 }, {
13, 17, 19, 23 } };
        int toplam = topla(matris);
        Console.ReadKey();
    }

    static int topla(int[,] m)
    {
        int toplam = 0;
        for (int i = 0; i < m.GetLength(0); ++i)
            for (int j = 0; j < m.GetLength(1); ++j)
                toplam += m[i,j];
        return toplam;
    }
}
```

Lab. Ödev

- Verilen bir matrisin elemanlarını L şeklinde toplayarak bir **diziye** atan **Ltopla metodunu** yazınız, ana metottan çağırınız, .cs uzantılı programı **lab. sonuna kadar** Moodle'a yükleyiniz :



Ltoplam
metodu ile
elde edilmeli

Homework 2

- **Lab'dan sonra yapılacaktır, El ile yazılarak 22 Ekim Salı günü derse getirilmesi yeterlidir.**

1) Lab. Ödevi'nin Ltopla metodunu, programı işletmeden gözle takip ederek, i ve j değişkenlerinin her bir değeri için $dizi[i]$ 'nin aldığı değerleri tablo üzerinde doldurunuz.

2) Örnek 1'e ilişkin gözlemlerinizi düzenleyerek el ile yazınız. F11, F10 ve Durma noktalarının hangi durumlarda yararlı olabileceğini ve farklarını kısaca belirtiniz. (yarım sayfa yeterli)

Adım	i	j	Dizi[i]
0	0	-	1
1	1	-	3
2	1	0	9
3	2	-	7
...	2	0	...
...	2	1	...
...			
...			
...			
9			

Ödev kağıdı için :

HW_II_Paper.pdf