

Sınıf yapısı içinde Çiftli Linked List

- Eleman ekleme çıkarmaların en üstten (top) yapıldığı veri yapısına yığın (stack) adı verilir. Bir eleman ekleneceğinde yığının en üstüne konulur.
- Yığın (Stack) sınıfını kullanmak için program kodunun baş tarafına
- `using System.Collections;` eklememiz gerekir.
- Bir eleman çıkarılacağı zaman yığının en üstündeki eleman çıkarılır. Bu eleman da yığındaki elemanlar içindeki en son eklenen elemandır. Bu nedenle yığınlara LIFO (Last In First Out : Son giren ilk çıkar) listesi de denilir.
- Yığın sınıfının `Push()` metodu yığına yeni bir eleman ekler. `Pop()` metodu ile yığının en üstündeki elemanı yığından siler ve silinen elemanı geriye döndürür.

Burada yazılan kodları bilgisayarınızda denemeniz önemle tavsiye edilir.

```
class Stack
{
    public int Count {get; set;}
    public string Dizin { get; set; }

    public string Push(string data)
    {
        Count++;
        Dizin = data;
        return data;
    }

    internal object Pop()
    {
        return Dizin;
    }
}

static public int[] stack = new int[100];
static public int sp = -1;

static public int Push(int data)
{
    if(sp == stack.Length)
    {
        return stack[(stack.Length - 1)];
    }
    sp++;
    stack[sp] = data;
    return stack[sp];
}

static public int Pop()
{
    if(sp == 0)
    {
        return 0;
    }

    int data = stack[sp];
    sp--;
    return data;
}
```

```
static void Main(string[] args)
{

Random rnd = new Random();
for(int i=0; i<100; i++)
{
    Push(rnd.Next(50));
}
string a = "e";
int rastgele = 0;
while (a == "e") {
    rastgele = Pop();
    Console.WriteLine("Rastgele Seçilen : " + rastgele);
    Console.WriteLine("SP Değeri : " + sp);
    Console.WriteLine("Devam Etmek istiyormusunuz? e ? h ?");
    a = Console.ReadLine();
}
//Yukarıda tanımlanan Stack verilerini sıfırlıyoruz
sp = -1;
stack = new int[100];

Console.WriteLine("Belli Ekleme ve çıkarmalar için son değer
sorgulama örnekleri");
Push(1);
Push(10);
Push(3);
Push(40);
Push(10);
Push(15);
Push(5);
Push(4);
Pop();
Pop();
Push(Pop()+Pop());
Console.WriteLine("SP Değeri : " + sp);
Console.WriteLine(stack[sp]);

Console.WriteLine("Devam Etmek için Enter");
Console.ReadLine();
```

```
//Stack sınıfının bir örneği oluşturuluyor
Stack s = new Stack();

//Hangi dizinden okunmaya başlanacağı söyleniyor
s.Push(@"C:\Program Files (x86)");

while (s.Count > 0)
{
    Object o = s.Pop();

    System.IO.DirectoryInfo di = new
System.IO.DirectoryInfo(o.ToString());

    System.IO.DirectoryInfo[] Directory = di.GetDirectories();
    foreach (DirectoryInfo dir in Directory)
    {
        s.Push(dir.FullName);
        Console.WriteLine("-----" + dir.FullName + "\r\n");
    }
}

Console.WriteLine("Yukarıdaki işlem hocanın derste söylediği stack
ile dizin içlerindeki tüm dizinleri gösterme");
Console.ReadLine();
```