**Yazılım Tasarım Belgesi**

**Software Design Document (SDD)**

**1. Giriş**

**1.1. Amaç ve kapsam**

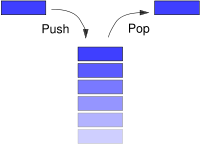
Bu projede, verilen bir metin dosyasında bulunan açılan ve kapanan parantezlerin birbirleriyle eşleşip eşleşmedikleri yığın (stack) yapısı kullanılarak kontrol edilecektir. Aslında çok basit bir derleyici (compiler) olarak da düşünülebilir.

**1.2. Sözlük**

**Yığın (stack):** Tek taraflı giriş ve çıkışlara açık olan. İlk giren son çıkar LIFO (Last in First Out) mantığı ile çalışan bir Abstract Data Type (Soyut Veri Tipi) örneğidir.

Temelde üç fonksiyonu bulunur bunlar: push, pop, peek  
**Push:** Stack içerisine bir bilgi koymaya (Stack’in en tepesine koyar)  
**Pop:** Stack içerisinden bir bilgi almaya (Stack’in en tepesinden alır)

**Peek:** Stack’in en tepesindeki veriyi okur.



**StreamReader:** StreamReader, dosya içerisinden veri okumak için kullanılmaktadır. StreamReader, System.IO namespace’ i altında bulunmaktadır.

**Namespace:** Aynı amaca hizmet eden özellikleri, sınıfları ve fonksiyonları bir çatı altında toplarken, bu çatıya verdiğimiz isime isim alanı (namespace) denir.

**Class:** bir nesnenin şeklini tanımlayan bir şablondur. Sınıf, verileri ve bu verileri işleyecek kodu içerir. Sınıf nesneleri oluşturmak için temel özellikleri içerir.

**1.3. Referanslar**

Bu projede, <http://www.csharpnedir.com/articles/read/?id=65>

<https://stackoverflow.com/questions/16618341/how-to-read-character-in-a-file-1-by-1-c-sharp>

<http://bilgisayarkavramlari.sadievrenseker.com/2007/05/04/stack-yigin/>

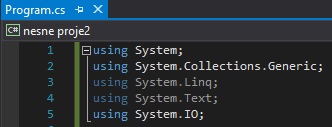
<https://caylakyazilimci.com/post/namespace-nedir>

<http://www.hikmetokumus.com/makale/60/streamreader-ile-dosya-okuma-islemleri>

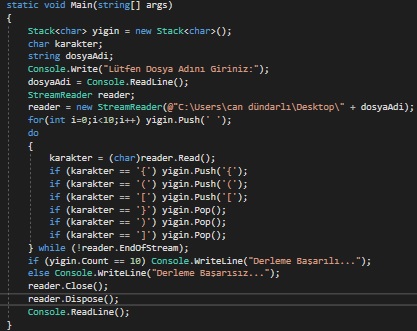
<http://kod5.org/c-sinif-yapisi/>

<https://mertcangokgoz.com/compiler-nedir-ise-yarar/> sitelerinden referans alınmıştır.

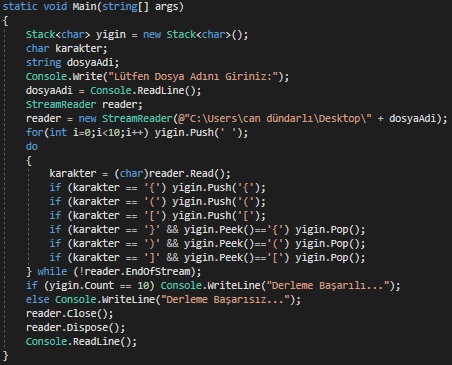
**2. Tasarım Kararları**

****

Programda namespacelerin bulunduğu alan.

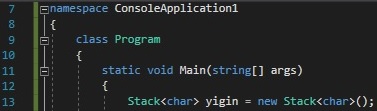


Programın koşulduğu alan

****

Programın dosyadaki farklı parantezlerin birbirini kapatma ihtimaline karşı düzeltilmiş hali

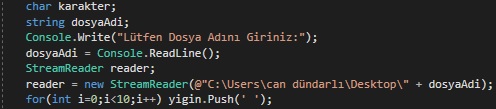
**3. Tasarım Detayları**



“Program” adında temel sınıf yapısı

Programın koşturulduğu ana fonksiyon

Char tipinde bir yığın oluşturulan alan



Dosyayı karakter karakter taraması için char tipinde “karakter” adında bir değişken tanımladım.

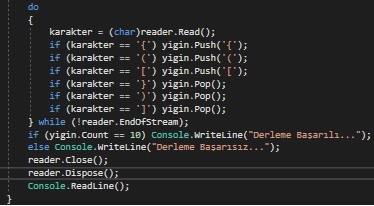
Dışarıdan dosya adını atayabileceği string tipinde “dosyaAdi” adında bir değişken tanımladım.

Kullanıcının ne yapması gerektiği hakkında ekrana bilgilendirme metni yazdırdım.

Kullanıcının girdiği dosya adını “dosyaAdi” adında oluşturduğum değişkene atadım.

Dosyanın açıldığı ve yolunun belirlendiği kısım.

Yığın boşken pop işlemi gerçekleştirme ihtimaline karşı hata vermesini kontrol etmek için yığına 10 adet boşluk karakteri yerleştirdim.



Dosyadan okuduğumuz karakteri “karakter” değişkenine atadım.

Parantezleri kontrol ederek yığının içine koyduğum kısım.

Yine parantezleri kontrol ederek yığından çıkardığım kısım.

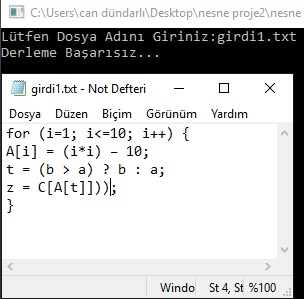
Dosyadaki karakterler bitene kadar döngünün devam etmesini sağlayan kısım.

Derlemenin başarılı olup olmadığını kontrol ettirdim.

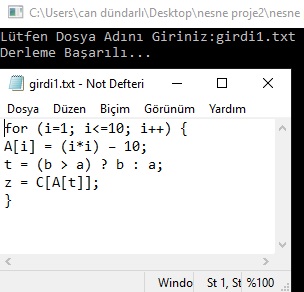
Açılan dosyayı kapattım.

Programın koştuktan sonra ekranda durması için gerekli olan kısım.

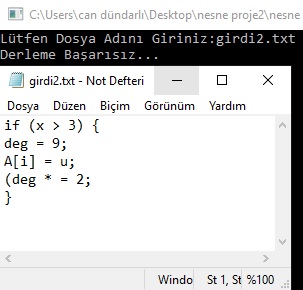
**4. Girdi/Çıktı Örnekleri**

****

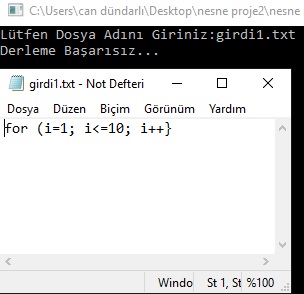
Burada “girdi1.txt” dosyasının içine fazladan parantez koydum.

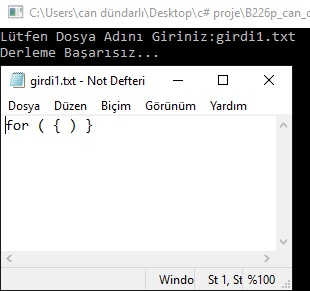


“girdi1.txt” dosyasının orijinal hali



“girdi2.txt” dosyası

****



Programı ilk yazdığımda dosyadan bir parantezi farklı bir parantezle kapatma ihtimalini düşünmemiştim onun kontrolünü koyup programı yeniden koşturdum.

**5. Sonuç**

Bu program bir compilerın (derleyicinin) neredeyse en basit halidir. Compiler bir programcının işini kolaylaştıran ve yazılan herhangi bir dilin bilgisayarın anlayabileceği bir biçime (0 ve 1) çeviren programdır. Eğer compilerlar geliştirilmeseydi. Şu an kullandığımız yüksek seviyeli programlama dillerini kullanamayacaktık, onun yerine kullanması daha zor olan ve daha fazla uğraş gerektiren makine diliyle uğraşmak zorunda kalacaktık. Belki de şu an kullandığımız programların birçoğu hala geliştirilememiş olacaktı. Compilerlar hemen hemen bilgisayarın bulunduğu her alanda bulunmaktadır. Şuan bile yazılımcıların işlerini bu kadar kolaylaştırıyorken ileride daha da üst seviye dillerin çıkması durumunda her insanın program yazmasında olanak sağlayacak duruma getirebilir.