**DÖKÜMANTASYON**

**Projenin Adı:** “Bir Kelime Bir İşlem” Oyununun Tasarımı

**Erişim Bilgileri:**

Trello: <https://trello.com/b/AXVS0CEm>

Github: (master) <https://github.com/oguzcanozuilerigiden/BirKelimeBirIslem>

**Ekler:**

- Proje Raporu

- Trello Ve Github Ekran Görüntüleri

- Proje Kod Dosyası (Katmanlı mimari kullanıldığı için proje dosyası üzerinden inceleyiniz.)

- Video Anlatım

**Hazırlayanlar:**

OĞUZCAN ÖZÜİLERİGİDEN (192802093 Müh. Tam.) ARMAĞAN KOCAOĞLU (192802089 Müh. Tam.)

**Projenin Amacı**:

Kod karmaşası olmadan, kötü kod yazımından kaçınarak ve anlaşılır bir algoritma ile “Bir Kelime Bir İşlem” oyununu tasarlamak.

**Projede Kullanılan Teknolojiler:**

- Visual Studio 2017/ C#

- .NetFramework 4.7.1

- %100 Nesne Tabanlı Programlama

- DevExpress 19.2 Gui Tasarım

- Katmanlı Mimari

- Github Versiyonlama ve İş Paylaştırma (Branch)

- Trello ile Github ve Visual Studio programını senkronize olarak ilişkilendirme

**Projenin Tasarımı:**

a) Proje BlankSolution olarak oluşturulmuş, DAL, Model, UI.Win Katmanları sonradan eklenmiştir. Bu sayede farklı kodlama dilleri ve projelerinin de sisteme adapte edilebilirliği gerçekleşmiştir.

b) Tüm proje, içeriklere göre ayrı ayrı klasöre alınarak, her sayfa kendi klasöründe geliştirilmiştir.

c) Tüm olaylar OOP kullanılarak class yapısına aktarılmıştır. (Tek sorumluluk prensibi kullanılmıştır.)

d) Sayfalar BaseForm üzerinden impemente edilmiş, tek bir tasarım ile tüm sayfaların menüye sahip olması sağlanmıştır.

e) DevExpres in güçlü tasarım özelliklerinden faydalanılmıştır.(Tasarım, Supertipe, Kısayol tuşu)

f) Tasarımda kullanılan Toolbox Componentler base’den alınarak düzenlenmiş, kendi kullanacağımız Toolbox oluşturulmuştur.

g) Model katmanında gerekli olan Entity ler oluşturulmuş, ortak özellikler baseclass’a aktarılmış, kullanılacak olan entitiler bu sınıftan kalıtım alarak türemiştir.

h) Dal kamanında Excel ile bağlantı kuran yapı kodlanmıştır.(Singleton Tasarım deseni)

i) 87000 kelimelik bir veri tabanı oluşturulmuştur. Sadece oyuna başlandığında database tanımlı entity’e aktarılmış, her seferinde veri çekme işlemi yapılmadığı için performans sağlamıştır.

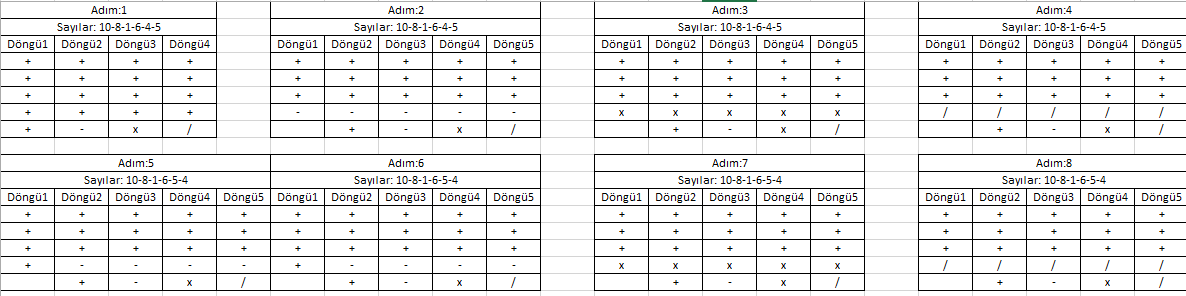
j) Projenin hızlı çalışabilmesi için gereksiz değişken, nesne tanımlamalarında kaçınılmış, erişim belirleyicilerinin private ve protect olmasına özen gösterilmiştir.

k) Gerekli olan testcase’ler denenmiştir.(WhiteBox)(Null geçme, farklı değer girme, numeric,character,………)

l) Açıklama satırlarına yer verilmiştir.

**Projenin Çalışması:**

İşlem Oyunu: Sayılar sistem tarafından yada kullanıcı tarafından belirlenebilmektedir. Class a gönderilen sayılar list yapısı altında işleme alınmaktadır. Enum ile işlem operatörleri(+-\*/) saklanmaktadır. Rakam listesi son karakterden itibaren işaret değiştirerek(diğer sayılar sabit) taranmaktadır. Sonrasında sırası ile 5., 4. ….1. sayı işaret değiştirerek 6n kere tarama yapılmaktadır. Tüm operatörler, rakamlar ve oluşan ara değerler ile bu opsiyonlar denenmektedir. İşlem adımlarının bir bölümünün özeti Şekil1 de gösterilmiştir. Yaklaşık olarak 20.000 ile 100.000 adım arasında deneme gerçekleşmektedir. Tostring() methodu override edilerek veriler listbox a aktarılmaktadır.



Şekil 1 Veri Tarama

Kelime Oyunu: Harfler sistem tarafından yada kullanıcı tarafından belirlenebilmektedir. Harfler ve dosyadan alınan kelimeler classlara gönderilmektedir. Kelimenin kopyası alınıp harf listesindeki karakterlere sahip olup olmadığı incelenmektedir. Eğer harf kelime içinde var ise kopya kelimeden o harf silinmektedir. Son olarak kopya kelimede kalan harf sayısı 0 yada 1 ise harf sayısı, kelime ve anlamı list’ e alınmaktadır. Bu esnada en max. Harf adedi detespit edilmektedir. Taramanın ardından elde edilen list’deki adet max değer ile karşılaştırılmakta, eşit ise ekrana batıılacak olan list’e aktarılmaktadır. Son olarak Tostring() methodu override edilerek veriler listbox a aktarılmaktadır.