NESNEYE DAYALI PROGRAMLAMA DERSİ 1. SINIF 2. DÖNEM PROJE ÖDEVİ RAPORU

İçindekiler :

* Projenin Amacı
* Projenin Tasarımı
* Projenin Yöntemi
* Projenin Bana Kazandırdıkları

Projenin Amacı :

Vitamin Deposu oyunu nesneye dayalı programlama prensipleri kullanılarak C# dilinde Windows Forms uygulaması şeklinde yazılacak ve programda kullanılan sınıfların UML sınıf diyagramının olduğu bir rapor dosyası hazırlanacaktır.

Projede nesneye dayalı programlama prensipleri kullanılmalıdır. Programda kullanılan her varlık için bir sınıf oluşturulmalı, sınıflar uygun şekilde başka sınıflardan miras almalı, arayüzler (interface) ve soyut sınıflar (abstract class) vs. kullanılmalıdır.

Özetle, nesneye dayalı programlamanın öğrenildiği yazılan kodlarla gösterilmelidir.

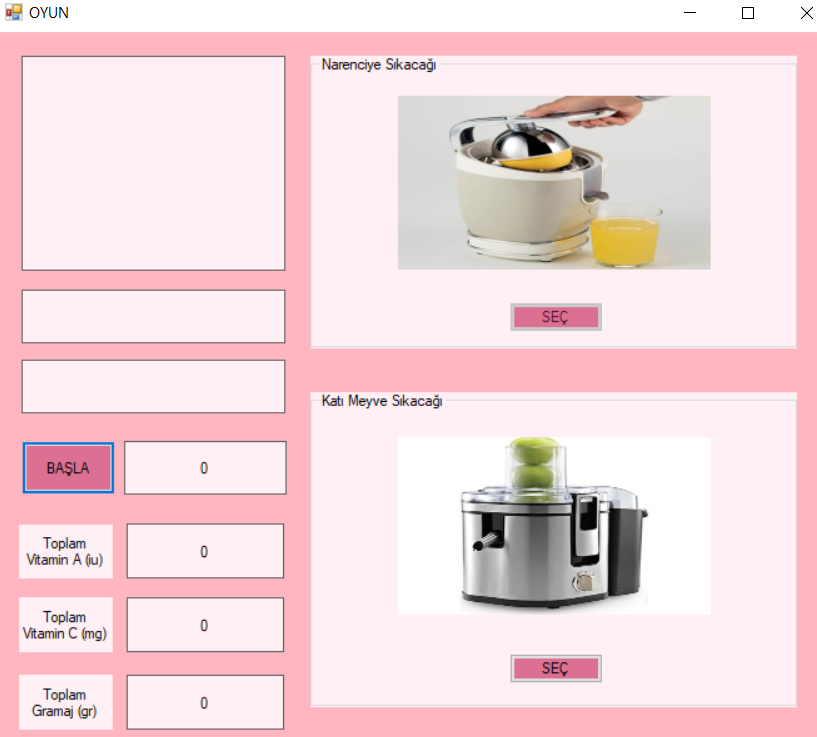
Projenin Tasarımı :

Öncelikle projemin son halinden bahsetmek istiyorum. Ödevde arka plan rengi olarak *LightPink* rengini kullandım. Form elemanlarının arka plan rengini ise biraz daha açık bir ton olan *LavenderBlush* olarak ayarladım.

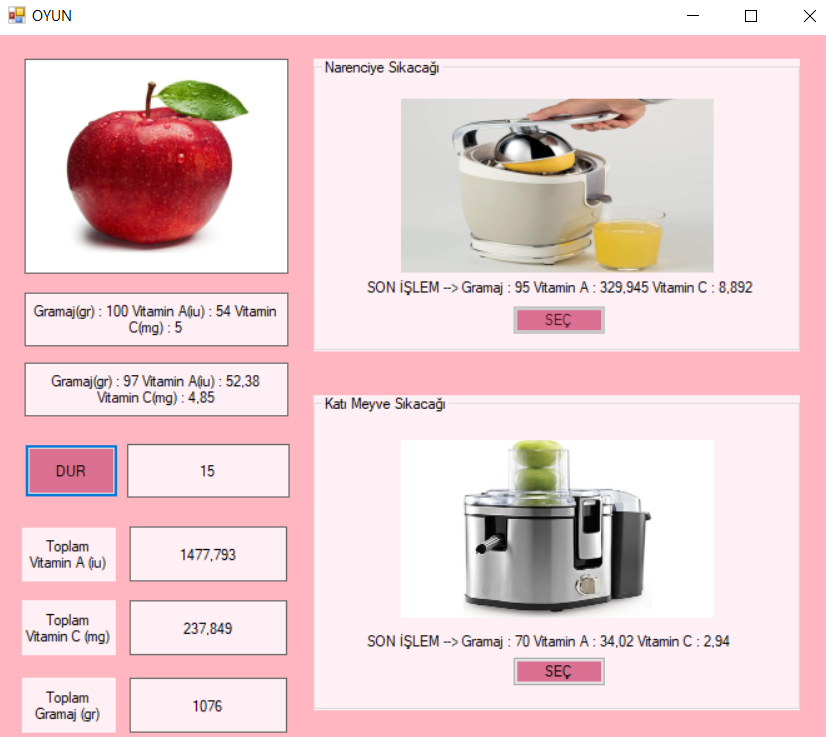
Formumun sağ tarafında İki farklı meyve sıkacağı için iki farklı groupBox kullandım. Her bir groupBox içinde kullanacağım sıkacağın resmini , ismini ve gelen meyveye göre sıkacak seçmek için seç isimli bir button yerleştirdim.

Formumun sol kısmında ise gelecek meyvenin resmini içerisine koymak için bir pictureBox, meyvenin ortalama değerierini ve gelecek olan meyvenin ağırlığına göre değişecek olan vitamin değerlerini içerisine yazdırdığım iki adet label ve oyunu başlatıp durdurabilmek için bir tane button ekledim. Butonun hemen yan tarafında içinde geri sayımın yapılacağı yeni bir label daha oluşturdum. Başla butonunun altında kalan diğer alanlarda ise oyunun gidişatına göre meyvelerin toplam gramajları ( gr cinsinden ) , toplam A vitamini değerleri ( iu cinsinden ) ve toplam C vitamini değerleri ( mg cinsinden ) bulunmaktadır.

Projenin çalıştırılmış ve çalışma esnasında durdurulmuş hali şekildeki gibidir.



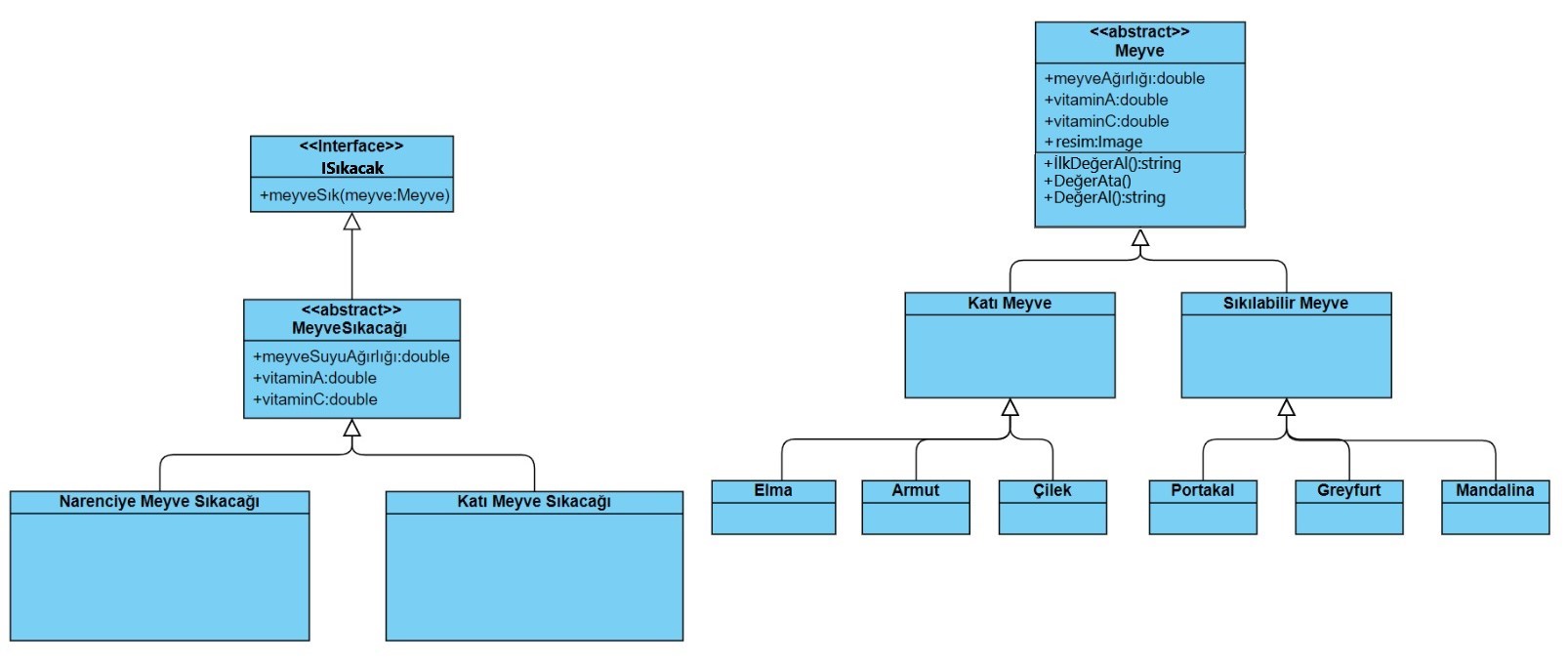
Program ilk çalıştırıldığında ekrana böyle bir form uygulaması gelmektedir.



Oyunun gidişatına ve gelen meyvelerin ağırlığına göre hesaplanan toplam ağırlık ve değerler şekilde görüldüğü gibi ilgili alanlara yazdırılır.

Projenin Yöntemi :

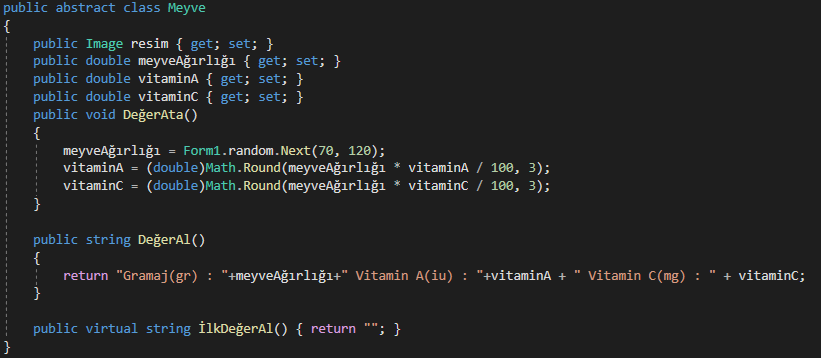
Projenin yönteminden bahsetmeden önce uml diyagramı üzerinden proje ödevinde kullandığım sınıflardan bahsetmek isterim.



Uml diyagramı dikkate alındığıda nesneye dayalı programlama prensiplerinin ödev içerisinde istenildiği gibi kullanıldığı görülür.

Projede kullanılan bütün sınıflar temelde iki farklı parçaya ayrılabilir: ISıkacak isimli interface tipindeki sınıftan kalıtım alanlar ve Meyve isimli soyut sınıftan kalıtım alanlar.

İlk olarak meyve sınıfından bahsedelim.



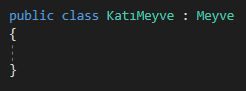
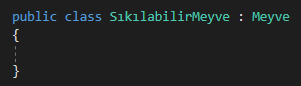
Meyve sınıfı katı veya sıkılabilir farketmeksizin tüm meyveler için kullanıldığından meyvelerin ortak özellik ve fonksiyonlarını public erişim belirteciyle içinde tutar.

Image tipinde resim ,double tipinde meyveAğırlığı , vitaminA ve vitaminC özellikleri ve bu özelliklerin set ve get fonksiyonları burada tutar.

Bunun dışında void tipinde DeğerAta fonksiyonu çağırıldığında meyve ağırlığını 70 -120 gr arasında oluşturur ve oluşan ağırlığa göre meyvenin A ve C vitamini değerlerini hesaplar.

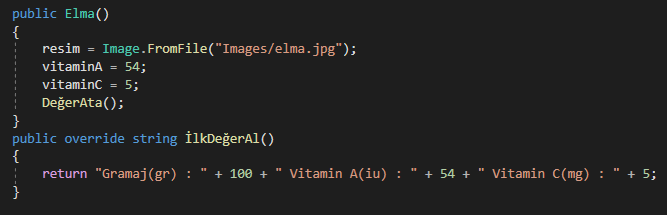
String tipinde DeğerAl() fonksiyonu ise çağırıldığında o anki meyvenin ağırlığını ve vitamin değerlerini döndürür.

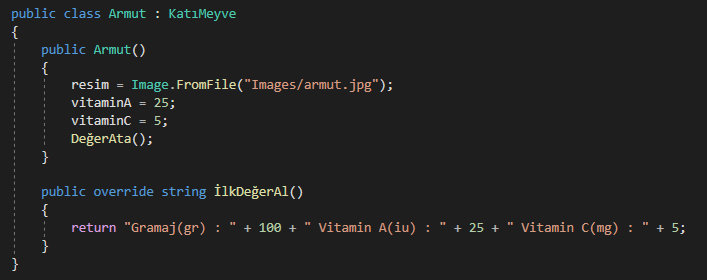
Son olarak virtual olarak ayarlanmış string tipinde İlkDeğerAl() fonksiyonu bulunmaktadır . Bu fonksiyonun içerisinde şimdilik hiçbir şey yoktur daha sonra override edilerek içi doldurulacaktır fakat void tipinde değil de string tipinde olmasını istediğim için fonksiyonun içinde “return “” ; ” ifadesini kullandım.

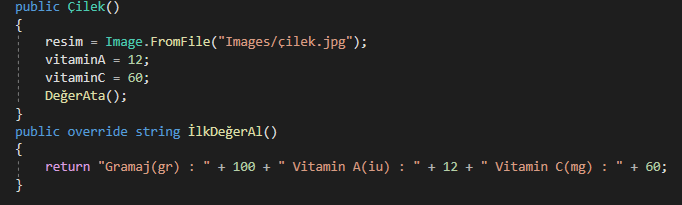
 

Şimdi sırada Meyve sınıfından kalıtım alan KatıMeyve ve SıkılabilirMeyve sınıfları var .Bu sınıfların içlerinde herhangi bir işlem yapmadım. Sadece daha sonrasında tanımlayacağım meyvelerin katı meyve mi yoksa sıkılabilir bir meyve mi olduğunu belirlemek için kullanmak üzere bu sınıfları oluşturdum. Meyvenin katı ya da sıkılabilir olmasına göre program değişeceği için bu gerekliydi.

KatıMeyve sınıfjndan kalıtım alan Elma , Armut ve Çilek sınıfları var sırada .



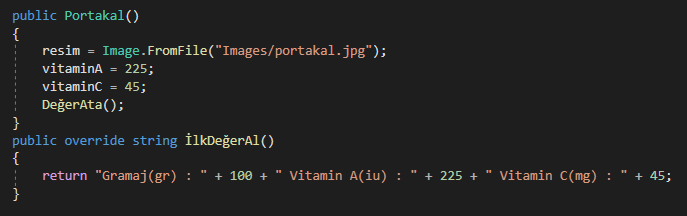


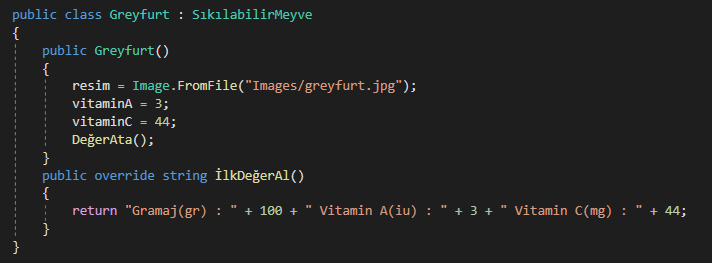


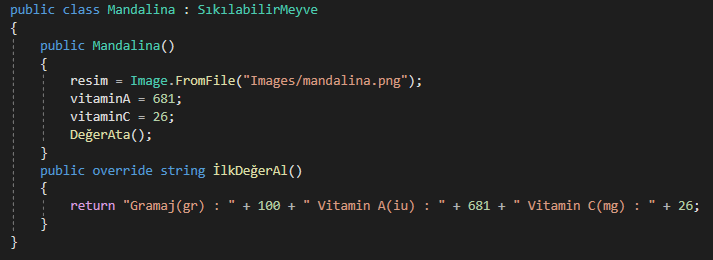
Genel olarak bu üç sınıftan bahsedecek olursak , öncelikle her sınıfın içerisinde meyveden gelen özellikler , değerAta() fonksiyonu ve her bir meyve sınıfı içerisinde override edilmiş İlkDeğerAl() fonksiyonu bulunmaktadır. Yine her bir sınıfta hangi meyvenin sınıfıysa image dosyasının içerisinden o meyvenin resminin yolu resim özelliğine atanmış ve aşağıya ekleyeceğim bilgileri baz alarak her meyvenin vitamin A ve vitamin C değerleri elle girilmiştir. Bu bilgiler meyvenin yüz gramındaki değerlerdir.

VitaminA ve Vitamin C değişkenlerine ilk değerleri atandıktan sonra DeğerAta() fonksiyonu çağrılmştır. Bu fonksiyon meyveye rastgele bir ağırlık vererek bu ağırlığa göre vitamin A ve vitamin C değerleri hesaplanır.

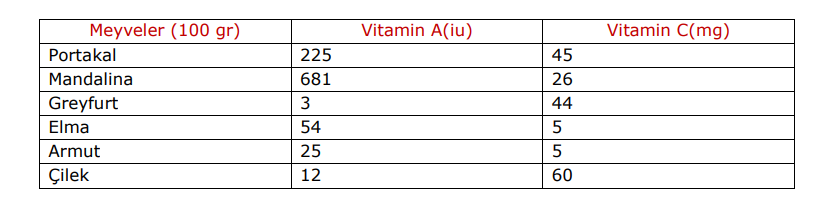
Daha sonrasında meyvenin hangi meyve olduğuna bağlı olarak İlkDeğerAl() fonksiyonu override edilmiştir. Bu fonksiyon daha sonra çağrıldığı label’ın içerisine meyvenin 100 gramındaki A ve C vitamini değerlerini yazdıracaktır.





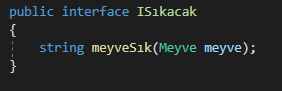


Sıkılabilir meyveler de katı meyvelerle aynı özellikleri ve fonksiyonları barındırır ve çalışma prensipleri aynıdır.

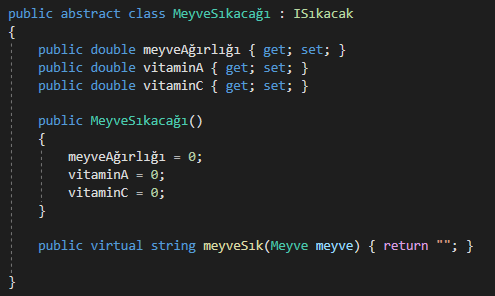


Override edilen İlkDeğerAl() fonksiyonu bu tabloya göre düzenlenmiştir.

Meyve sınıfından kalıtım alan sınıflardan bahsettikten sonra sıra ISıkacak isimli arayüzden ve bu arayüzden kalıtım alan sınıflardan bahsetmek istiyorum.



ISıkacak isimli sııfımızın içerisinde Meyve tipinden bir nesne alan string değer döndüren meyveSık fonksiyonundan başka herhangi bir şey yoktur. meyveSık() fonksiyonu parametre olarak aldığı meyveye göre yeniden yazılacaktır.

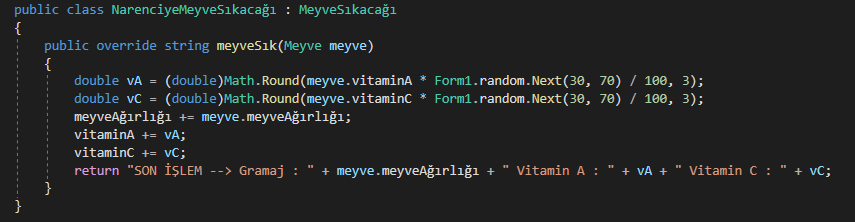


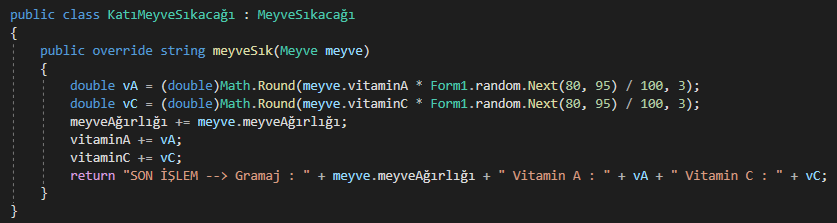
MeyveSıkacağı isimli soyut sınıf ISıkacak isimli sınıftan kalıtım almaktadır. Bu sınıftan nesne üretilmediği için abstract olarak tanımlanmıştır. İçerisinde her biri public olmak üzere double tipinde meyveAğırlığı , vitaminA ve vitaminC özellikleri ve bu özelliklerin set ve get fonksiyonları tanımlsnmıştır.

Daha sonra MeyveSıkacağı sınıfının kurucu fonksiyonunda bu özelliklere ilk değer olarak sıfır atanmıştır.

String tipinde meyveSık fonksiyonu ise Meyve tipinden bir nesneyi parametre olarak almış ve virtual olarak tanımlanmıştır.

Sırada ise MeyveSıkacağı sınıfından kalıtım alan NarenciyeSıkacağı ve KatıMeyveSıkacağı sınıfları var.





Aslında bu iki sınıfın çalışma mantığı birbirinin tıpatıp aynısıdır. Aralarındaki farkın sebebi Narenciye Sıkacağı ve Katı Meyve Sıkacağı kullanıldığında meyvenin veriminin fark göstermesidir.

Her iki sınıfta da meyveSık fonksiyonu override edilmiştir. meyveSık fonksiyonunda parametre olarak alınan meyvenin vitaminA ve vitaminC değerleri narenciye sıkacağında 30-70 arasında üretilen rastgele bir sayıyla çarpılarak yüze bölünür . Yani yüz gramında bulunan değerinden yararlanlarak meyvenin o anki ağırlığına göre vitamin değerleri bulunur. Daha sonra Math.Round fonksiyonuyla yuvarlanarak double tipindeki vA ve vC değişkenlerine yazılır. Bu değişkenler içlerinde rastgele olarak ağırlığı verilmiş meyvenin ağırlığına göre hesaplanmış A ve C vitamini değerlerini tutmaktadır.

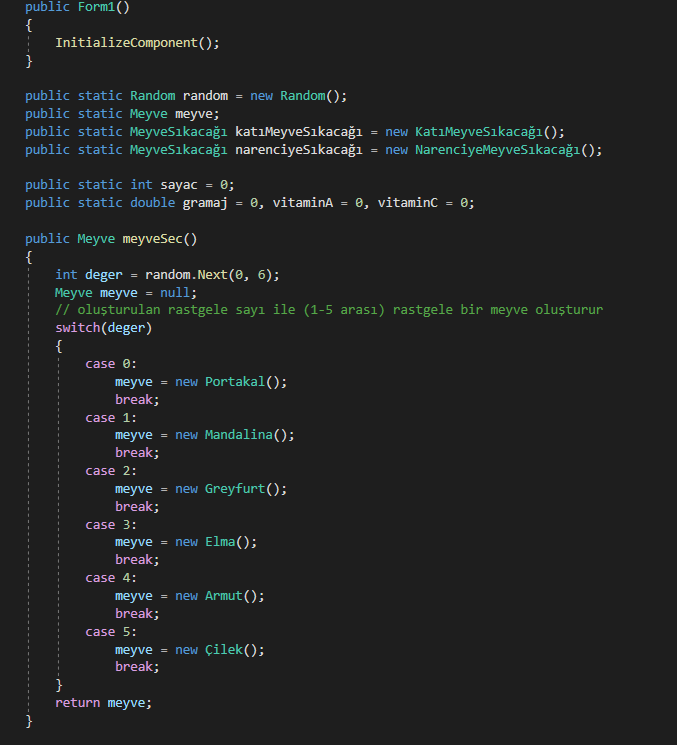
Daha sonra o ana kadarki tüm meyvelerin ağırlık ve vitamin değerleri toplanır.

meyveSık() fonksiyonu çağrıldığı yere meyvenin ağırlığını ve vitamın değerlerini döndürür.

Bulunan tüm bu değerler form üzerindeki label’lara yazılacaktır.

Şu ana kadar proje ödevinde kullanılan bütün sınıfların çalışma mantığı kısaca anlatılmıştır.

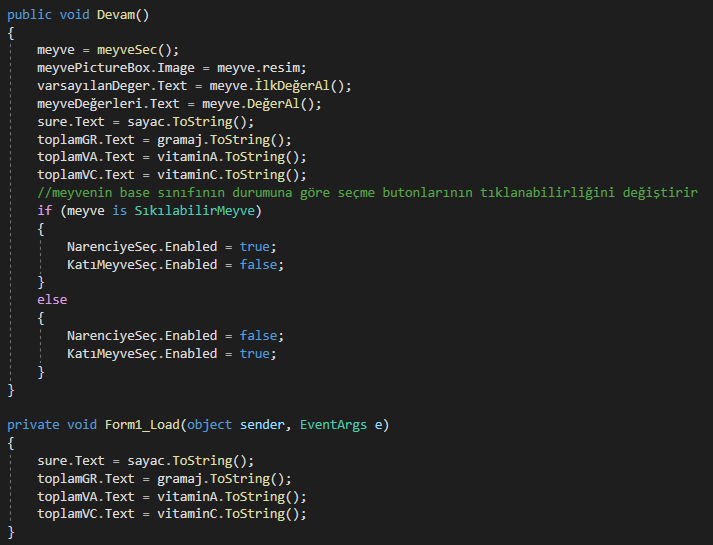
Son olarak form uygulamasından kısaca bahsedilecektir.



Form sınıfının içerisinde öncelikle rastgele değerleri üretmek için random sınıfının referansı oluşturulmuştur. Daha sonra meyve isimli Meyve tipinde bir nesne, MeyveSıkacağı tipinde katıMeyveSıkacağı referansına sahip katıMeyveSıkacağı nesnesi ve yine MeyveSıkacağı tipinde NarenciyeSıakcağı referansına sahip NarenciyeSıkacağı nesnesi oluşturulmuştur.

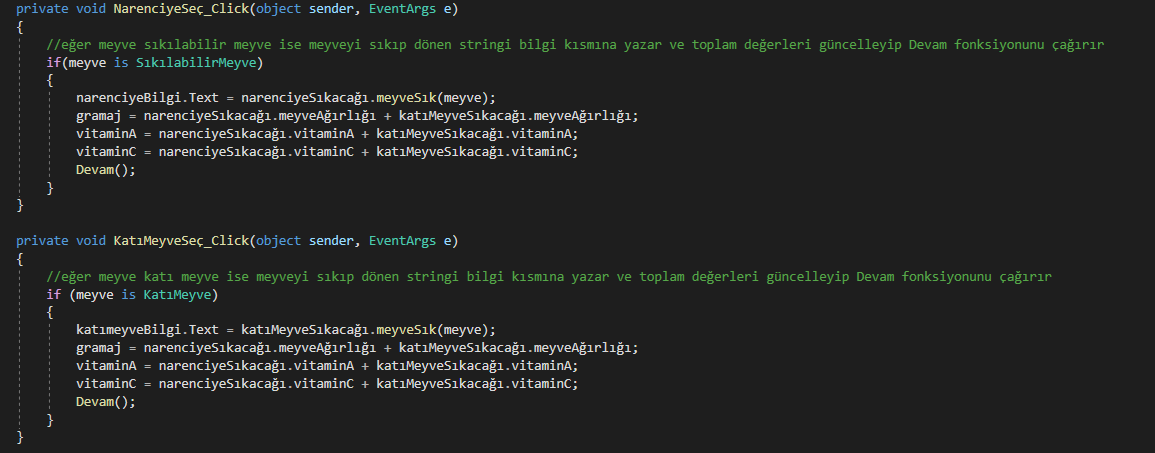
Daha sonra ilk değer olarak sayac , gramaj , vitaminA ve vitaminC değişkenlerine 0 değeri atanmıştır. Bu değerler form ilk çalıştırıldığında ekrana gelecek olan değerlerdir.

MeyveSec() fonksiyonu içerisinde bir değişkenin içinde rastgele üretilen 0 ile 6 arasında bir sayı tutulmuş , daha sonra üretilen sayıya göre meyveler oluşturulmuştur .Böylelikle form ekranına gelecek olan meyveler rastgele oluşturulmuş ve meyve nesnesine referansı verilmiş oldu.



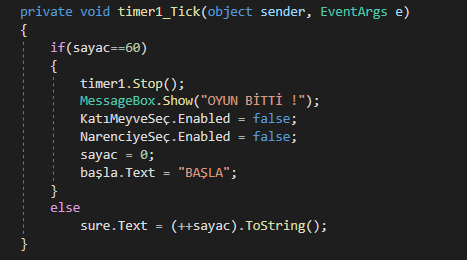
Devam() fonksiyonu süre bitmediği sürece oyun esnasında durmadan yeni meyve üretir. Daha sonra form üzerindeki PictureBox’un içerisine rastgele seçilen meyvenin resmini koyar. VarsayılanDeger label’ının içerisine İlkDeğerAl() fonksiyonunun döndürdüğü stringi yazar. Aynı şekilde meyveDeğerleri label’ının içerisine ise DeğerAl() fonksiyonunun döndürdüğü değer yazılır.

Tüm bu form elemanları içerisine yazılacak olanlar bittikten sonra bir kontrol ifadesi oluşturulur. Bu kontrolün amacı gelen meyve eğer katı bir meyveyse kullanıcının sadece katı meyve sıkacağını , eğer sıkılabilir bir meyveyse sadece narenciye sıkacağını seçtiğinden emin olmaktır. Eğer meyve sıkılabilir bir meyveyse katı meyve sıkacağını seçmemizi sağlayan butonun tıklama özelliği kapatılırken , meyve katı bir meyveyse narenciye sıkacağını seçmemizi sağlayan butonun tıklama özelliği kapatılır.



Narenciye sıkacağı seçildiğinde meyvenin tekrar sıkılabilir bir meyve olduğu kontrol edildikten sonra narenciye sıkacağının meyveSık() fonksiyonu çağırılır ve geriye dönen değerler olması gereken yere yazılır. Daha sonra form üzerinde toplam gramajı , toplam A vitaminini ve toplam C vitaminini gösterecek yerlere her iki meyve sıkacağındaki değerler de toplanıp yazılır.

Tüm bu işlemler Katı Meyve Sıkacağının seçildiği durumlarda da gerçekleşir.

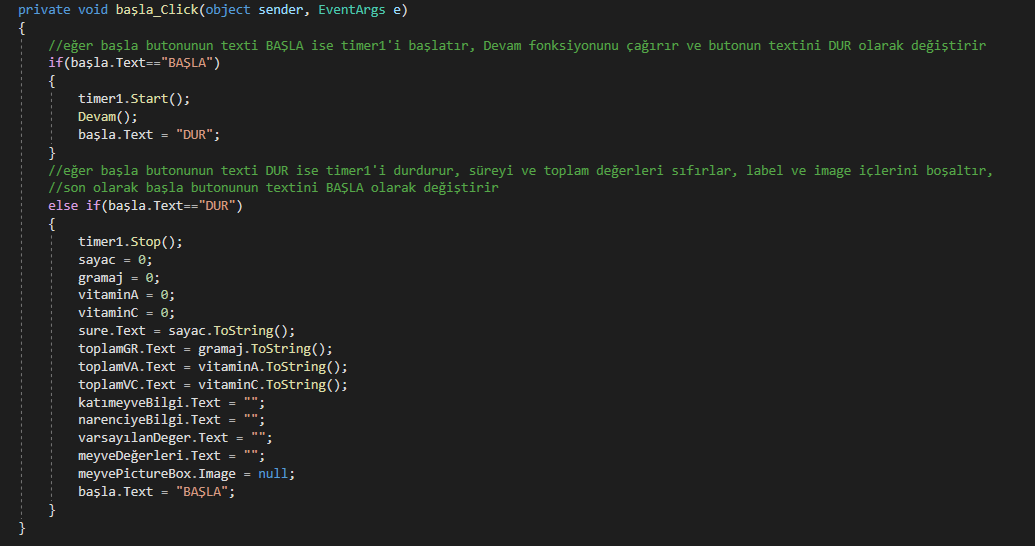


Burada ise sürenin durumu kontrol edilir. Oyunda süre 0’dan başlayıp 60’a kadar devam eder. 60 olduğunda oyun biter ve her şey varsayılan haline döner. Bu durumu sağlamak için ikili bir kontrol ifadesi oluşturulmuştur.

Eğer sayaç 60 değerine ulaştıysa timer durdurulur ve ekrana “OYUN BİTTİ ! “ mesajı çıkartılır. Sıkacakları seçebilmemizi sağlayan butonlar tıklanamaz haline getirildikten sonra her şey varsayılan haline döner.

Eğer sayaç 60 değerine ulaşmamışsa o zaman sayacın değeri birer birer artmaya devam eder.

Sırada formumuzun son birkaç işlemi kaldı.



Başla butonu iki farklı görevi üstleniyor. Ya oyunu başlatabilir ya da oyunu sonlandırabilir. Eğer oyun başlatıldıysa butonun adı DUR olarak değişirken oyun durdurulduğunda ya da başlatılmadığında BAŞLA olarak değiştirilir.

Eğer butonun text’i BAŞLA iken butona tıklanıldıysa bu oyunun başlatılmak istenildiiğini gösterir. Bu sebeple önce sayaç başlatılır, sonra devam() fonksiyonu çağırılır , son olarak da başla butonunun text’i DUR olarak değiştirişir.

Eğer butonun text’i DUR iken butona tıklanıldıysa bu oyunun durdurulmak istenilidiğini gösterir. Bu sebeple önce sayaç durdurulur , sonra tüm değerler varsayılan duruma getirilir ve son olarak başla butonunun text’i BAŞLA olarak değiştirilir.

Ödevimi yaparken kullandığım yöntemlerle alakalı olarak anlatacaklarım bu kadar.

Projenin Bana Kazandırdıkları :

Bu projeyi yaparken nesneye dayalı programlama dersinde öğrendiğim bilgileri pekiştirmiş oldum. Nesneye dayalı programlama prensiplerini, sınıflar arası etkileşimleri , interface ve soyut sınıf kavramlarını uygulayarak öğrendim. İlk defa form uygulaması üzerinde bir oyun tasarlamış oldum.