

SAKARYA ÜNİVERSİTESİ
NESNEYE DAYALI PROGRAMLAMA PROJE
ÖDEVİ



SAKARYA
ÜNİVERSİTESİ

Adı ve Soyad:Onur KAHVECİ

Numrrası:B201210500

Grubu:1.Öğretim B Grubu

Mail Adresi:onurk61@hotmail.com , onur.kahveci3@ogr.sakarya.edu.tr

İçindekiler

Proje Tanımı

Projenin Çalışan Halinin Ekran Görüntüsü

Komut Örnekleri

UML Diyagramı

Ödevin Bana Kazandırdıkları

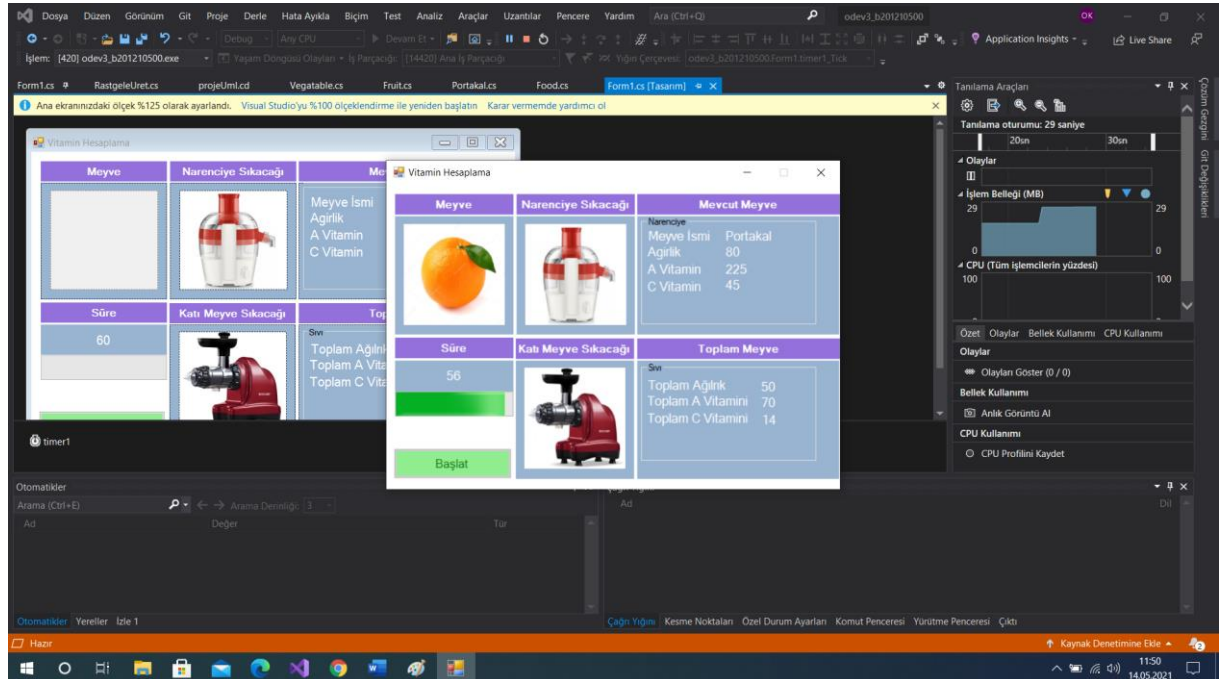
A. Proje Tanıtımı

Aşağıda detayları verilen 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

Kullanılan Veriler

Meyveler (100 gr)	Vitamin A(iu)	Vitamin C(mg)
Portakal	225	45
Mandalina	681	26
Greyfurt	3	44
Elma	54	5
Armut	25	5
Çilek	12	60

Çalışan Halinin Ekran Görüntüsü



Örnek Birkaç Fonksiyon:

Projemizde meyve oluşturmak için ArrayList kullanılarak fonksiyon oluşturduk. Bu fonksiyon ile her çağrıda 6 tane nesne oluşturulmaktadır.

MeyveOlustur() Fonksiyonu

```
List<Fruit> meyveler;  
    public void meyveOlustur()  
    {  
        meyveler = new List<Fruit>();  
meyveler.Add(new Portakal(RastgeleUret.Uret(70, 120), Image.FromFile("portakal.jpg")));  
meyveler.Add(new Mandalina(RastgeleUret.Uret(70, 120), Image.FromFile("mandalina.jpg")));  
meyveler.Add(new Greyfurt(RastgeleUret.Uret(70, 120), Image.FromFile("greyfurt.jpg")));  
meyveler.Add(new Elma(RastgeleUret.Uret(70, 120), Image.FromFile("elma.jpg")));  
meyveler.Add(new Armut(RastgeleUret.Uret(70, 120), Image.FromFile("armut.jpeg")));  
meyveler.Add(new Cilek(RastgeleUret.Uret(70, 120), Image.FromFile("cilek.jpg")));  
    }
```

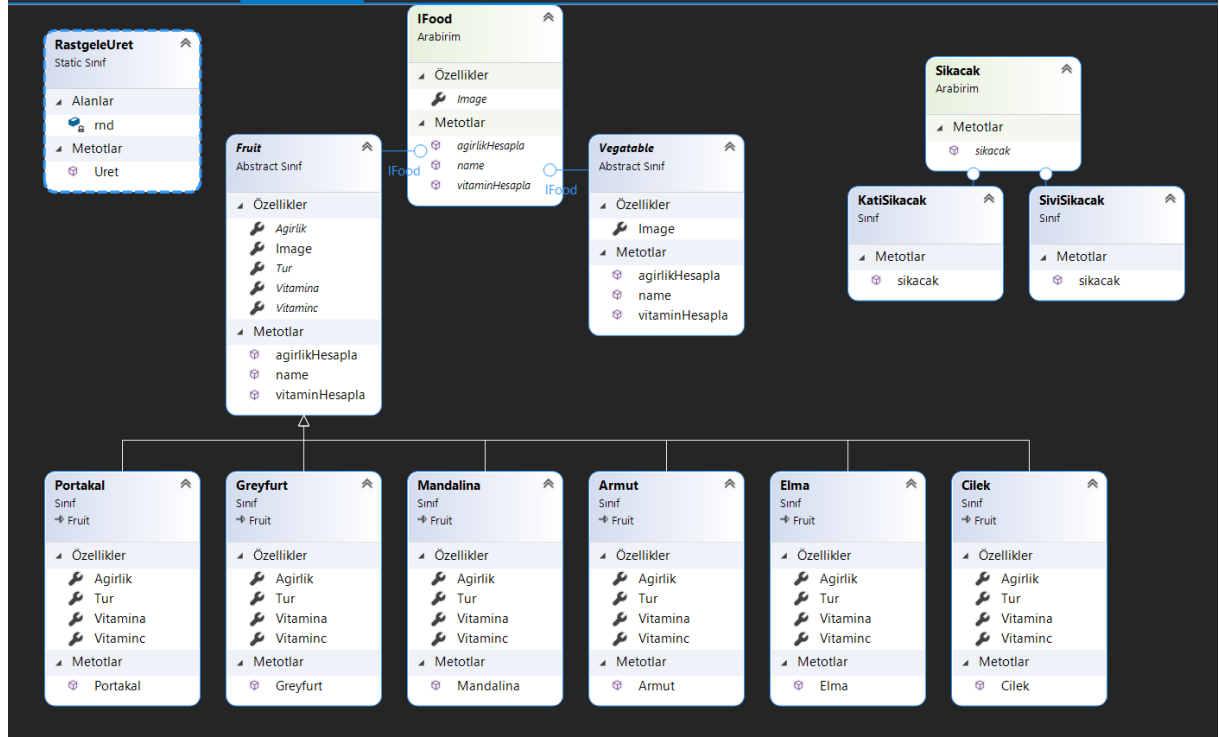
Ard arda gelen meyvelerin farklı olması için projemize

```
do  
{  
    gecici = RastgeleUret.Uret(0, 6);           //bu komut satırı ile aynı meyve gelmemiş oluyor  
}  
while (gecici == id);                           //tabi bu benim senaryom için geçerli  
    id = gecici;  
    meyve_pBox.Image = meyveler[id].Image;
```

komutları ekledik. Böylece ürünlerin farklı farklı seçilmesini sağladık. Tabi bunun böyle olmasını istemzsek üstteki komutların yerine sadece

`id = RastgeleUret.Uret(0, 6);` kullanmamız yeterli idi.

UML Diyagramı



Ödevin Bana Kazandırdıkları

Bu projeyele OOP için en önemli olan sınıf yapısını tasarlamayı öğrenmiş olduk .Abstract, İnterface gibi kavramları somut bir örnek için uyguladık.Bu sayede veri tekrarı azalmış , nesne yönelim bir programlama yapmış olup geliştirilebilir bir proje için ilk adımı atmış olduk.Tabi ki bu proje için farklı tasarımlar yapabiliriz.Hatta tek bir sınıf ile dahi yapabiliriz fakat istediğiniz kısmen gerçekleşmiş olsa da süreklilik anlamında amacımıza ulaşamamış olurduk.

Ne demek istiyorum biraz daha açayım. Sadece 1 sınıf kullanarak yaptığımızı varsayalım. Proje süresince farklı ürünlerin geldiğinde tasarımı baştan yapmak zorunda kalacaktık. Aslında küçük projelerde pek göze gelmese de büyük projeler için büyük maliyet demektir.

OOP'nin en önemli amacı da zaten geliştirilebilir projeler yapmamıza imkan sağlamasıdır. Elimizde bir hamur ve biz o hamurdan farklı varlıklar elde edebiliyoruz bu yapı kullanarak

İşte tüm anlattıklarımızı bizler bu projede öğrenmiş olduk.