



**NESNEYE DAYALI PROGRAMLAMA**  
**PROJE ÖDEVİ**

**2021-2021 Eğitim-Öğretim Yılı**  
**Bahar Dönemi**

**AD:** Yunus Emre

**NUMARA:** b201210101


**SOYAD:** Akıncı

**SINIF:** 1 Öğretim B Grubu

**KONU:**

NDP prensiplerini kullanarak Windows Form ile Vitamin Oyunu tasarlama

Vitamin Deposu



BAŞLAT

SÜRE

0

Sıkılan Toplam Meyve(kg): 2,712

Narenciye Sıkacağı

Meyve	Ağırlık(g)	VitA(iu)	VitB(iu)	Meyve Suyu Ağırlığı(g)
Portakal 96	216	43,2	48,96	
Mandalin 101	687,81	26,26	55,55	
Mandalin 83	565,23	21,58	33,2	
Portakal 115	258,75	51,75	75,9	
Portakal 82	184,5	36,9	31,98	
Portakal 113	254,25	50,85	72,32	
Greyfurt 115	3,45	50,6	75,9	
Greyfurt 78	2,34	34,32	28,86	
Mandalin 107	728,67	27,82	63,13	
Greyfurt 93	2,79	40,92	44,64	
Greyfurt 119	3,57	52,36	82,11	
Mandalin 70	476,7	18,2	21	
Mandalin 90	612,9	23,4	41,4	
Mandalin 98	667,38	25,48	50,96	

Toplam VitA(iu): 4664,34  
Toplam VitB(iu): 503,64  
Toplam Narenciye Suyu(kg): 0,72591

Katı Meyve Sıkacağı

Meyve	Ağırlık(g)	VitA(iu)	VitB(iu)	Püre Ağırlığı(g)
Çilek 101	12,12	60,6	89,89	
Elma 91	49,14	4,55	78,26	
Armut 109	27,25	5,45	99,19	
Armut 74	18,5	3,7	59,94	
Çilek 106	12,72	63,6	96,46	
Armut 102	25,5	5,1	90,78	
Elma 88	47,52	4,4	74,8	
Çilek 75	9	45	60,75	
Çilek 86	10,32	51,6	72,24	
Elma 113	61,02	5,65	105,09	
Çilek 119	14,28	71,4	111,86	
Çilek 103	12,36	61,8	92,7	
Çilek 119	14,28	71,4	111,86	
Çilek 74	8,88	44,4	59,94	

Toplam VitA(iu): 320,89  
Toplam VitB(iu): 498,25  
Toplam Meyve Suyu(kg): 1,19446

Projeyi çalıştırdığımızda karşımıza böyle bir tasarım geliyor. Başlat butonuna bastığımızda ekrana önceden tanımladığımız 6 resimden rastgele bir tanesi geliyor ve süre 60 saniyeden geriye akıyor. Meyvelerin türüne göre sıkacak butonlarına tıklayıp meyveye ait özellikleri listBox'lara yazdırıyoruz. Her sıkacaktan sıkılmış meyvelerin toplam vitA, vitB ve meyve suyu değerleri ayrı ayrı label'lara yazılıyor. Başka bir label da iki sıkacakta sıkılan toplam meyve ağırlığı kilogram cinsinden yazılıyor.

## 1. IMeyve Interface'ı Tanımlandı

```
4 başvuru
interface IMeyve
{
    36 başvuru
    double Ağırlık();
    20 başvuru
    double Verim();
    20 başvuru
    double VitA(double vitDeger);
    20 başvuru
    double VitB(double vitDeger);
    21 başvuru
    Image MeyveResim { get; set; }
    10 başvuru
    Random Rnd { get; set; }
}
```

Interface içinde mecburen public methodlar ve nesneler tanımlandı. Ödevde NDP prensipleri kullanılması istendiği için bu kısmı interface ile yaptım. İstenirse **Abstract Class Meyve** şeklinde tanımlayabilirdik.

## 2. Sıkılabilen ve Sıkılamayan Meyveler için Abstract Sınıflar Oluşturuldu

```
3 başvuru
abstract class SikilamayanMeyveler : IMeyve
{
    10 başvuru
    public Random Rnd { get; set; }
    36 başvuru
    public double Agirlık()
    {
        double agirlık;
        Rnd = new Random();
        agirlık = Rnd.Next(70, 120);
        return agirlık;
    }
    20 başvuru
    public double Verim()
    {
        double verim;
        Rnd = new Random();
        verim = Rnd.Next(80, 95);
        return verim;
    }
    20 başvuru
    public double VitA(double VitDeger)
    {
        return Agirlık() * VitDeger / 100;
    }
    20 başvuru
    public double VitB(double VitDeger)
    {
        return Agirlık() * VitDeger / 100;
    }
    21 başvuru
    public Image MeyveResim { get; set; }
}
```

IMeyve'den kalıtım aldım ve methodların body'sini abstract class içinde doldurdum. VitA() ve VitB() nin parametresi olan VitDeger değişkenini ödev dosyasındaki meyvelerin vitamin değerlerine göre aldım.

Görüldüğü üzere Agirlık(), VitA(), VitB() methodları birbirinin aynısıdır. En başta interface kullanmak yerine abstract sınıf kullanıp body'lerini orada tanımlayabilirdik.

```
3 başvuru
abstract class SikilabilenMeyveler : IMeyve
{
    10 başvuru
    public Random Rnd { get; set; }
    36 başvuru
    public double Agirlık()
    {
        double agirlık;
        Rnd = new Random();
        agirlık = Rnd.Next(70, 120);
        return agirlık;
    }
    20 başvuru
    public double Verim()
    {
        double verim;
        Rnd = new Random();
        verim = Rnd.Next(30, 70);
        return verim;
    }
    20 başvuru
    public double VitA(double vitDeger)
    {
        return Agirlık() * vitDeger / 100;
    }
    20 başvuru
    public double VitB(double vitDeger)
    {
        return Agirlık() * vitDeger / 100;
    }
    21 başvuru
    public Image MeyveResim { get; set; }
}
```

### 3. Abstract Sınıflardan Türüne Göre Kalıtım Alan Meyve Sınıfları Oluşturuldu

```
2 başvuru
class Portakal : SikilabilenMeyveler
{
    1 başvuru
    public Portakal(Image portakal)
    {
        MeyveResim = portakal;
    }
}

2 başvuru
class Mandalina : SikilabilenMeyveler
{
    1 başvuru
    public Mandalina(Image mandalina)
    {
        MeyveResim = mandalina;
    }
}

2 başvuru
class Greyfurt : SikilabilenMeyveler
{
    1 başvuru
    public Greyfurt(Image greyfurt)
    {
        MeyveResim = greyfurt;
    }
}
```

**SikilabilenMeyveler** ve **SikilamayanMeyveler** abstract sınıflarından aldığım kalıtım ile meyve sınıfları oluşturdum. Sınıfların her birine parametre olarak **Image nesnesi** atadım.

```
2 başvuru
class Elma : SikilamayanMeyveler
{
    1 başvuru
    public Elma(Image elma)
    {
        MeyveResim = elma;
    }
}

2 başvuru
class Armut : SikilamayanMeyveler
{
    1 başvuru
    public Armut(Image armut)
    {
        MeyveResim = armut;
    }
}

2 başvuru
class Cilek : SikilamayanMeyveler
{
    1 başvuru
    public Cilek(Image cilek)
    {
        MeyveResim = cilek;
    }
}
```

↓ Artık Form1.cs kısmına geçebiliriz ↓

```
List<IMeyve> meyve = new List<IMeyve>()
{
    new Portakal(Image.FromFile("portakal.jpg")),
    new Mandalina(Image.FromFile("mandalina.jpg")),
    new Greyfurt(Image.FromFile("greyfurt.jpg")),
    new Elma(Image.FromFile("elma.jpg")),
    new Armut(Image.FromFile("armut.jpg")),
    new Cilek(Image.FromFile("cilek.jpg")),
};
```

List<IMeyve> şeklinde bir collection tanımlayarak meyve classlarımıza burada referans verdim. Kurucu methodların parametlerine önceden indirdiğim meyve görsellerini atadım.

```
4 başvuru
public partial class Form1 : Form
{
    1 başvuru
    public Form1()...

    int sure = 60;
    Random rnd = new Random();
    double toplamMeyve = 0;

    double vitAToplamNarenciye = 0;
    double vitBToplamNarenciye = 0;
    double toplamNarenciye = 0;

    double vitAToplamKati = 0;
    double vitBToplamKati = 0;
    double toplamKati = 0;

    1 başvuru
    private void Form1_Load(object sender, EventArgs e)...

    1 başvuru
    private void timer1_Tick_1(object sender, EventArgs e)...

    1 başvuru
    private void btnBasla_Click(object sender, EventArgs e)...

    1 başvuru
    private void btnNarenciye_Click(object sender, EventArgs e)...

    1 başvuru
    private void btnKati_Click(object sender, EventArgs e)...

    List<IMeyve> meyve = new List<IMeyve>()...;
}
```

Form1[tasarım] içinde kullandığım nesnelere ait olayları ve değişkenleri burada tanımladım  
Daha detaylı hali ödevin içinde yorum satırları içinde anlatılmıştır.

# UML SINIF DİYAGRAMI

UML sınıf diyagramı ödev dosyasındaki **UML\_Class\_Diagram.cd** içinde de mevcuttur. VS 2019 Class Designer ile yapılmıştır.

