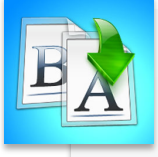


SINIF VE NESNE İLİŞKİSİ



İÇİNDEKİLER

- Sınıf ve Nesne Kavramları
 - Sınıf Kavramı
 - Nesne Kavramı
- Sınıf ve Nesne İlişkisi
 - Sınıf ve Nesne İlişkisine Örnekler



HEDEFLER

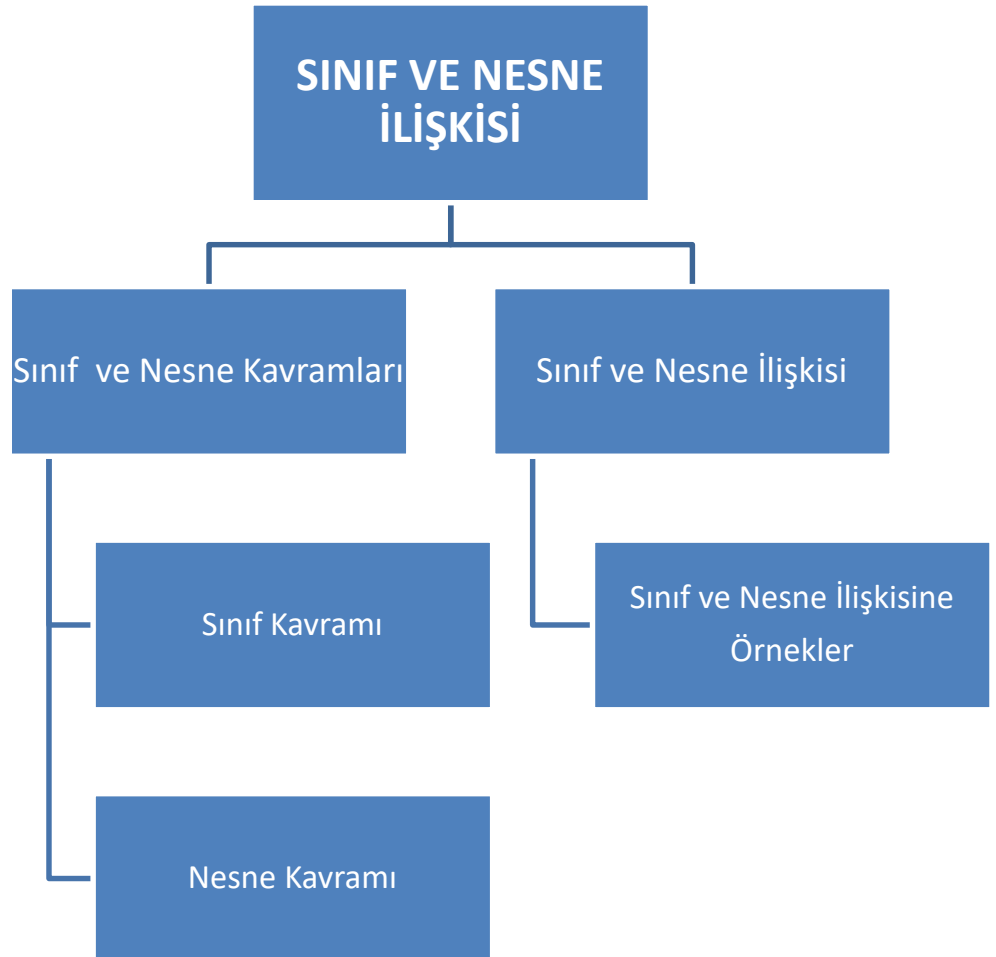
- Bu üniteyi çalıştıktan sonra;
 - Sınıf ve nesne kavramlarının ne olduğunu hatırlayabilecek,
 - Sınıf ve nesne kavramlarının ortak ve farklı özelliklerini kıyaslayabilecek,
 - Sınıf ve nesne kavramlarını kullanarak örnek uygulamalar yapabileceksiniz.



Atatürk Üniversitesi
Açıköğretim Fakültesi

NESNE TABANLI PROGRAMLAMA I İşhak Metehan SİS

ÜNİTE 5



GİRİŞ

Nesne tabanlı programlama, önceki ünitelerde de belirtildiği üzere yazılım tasarımı işlevler ve mantık yerine, sınıf ve nesneler etrafında düzenleyen bir programlama yaklaşımıdır. Nesne tabanlı programlama, programcıların işlemleri için gereken mantıktan ziyade işlem yapacağı nesnelere odaklanır. Nesne tabanlı programlamada büyük, karmaşık ve aktif olarak güncellenen veya bakımı yapılan programlar için çok uygundur. Bu yöntem işbirlikçi geliştirmeye de elverişlidir ve projeler böylece gruplandırılabilir.

Nesne tabanlı programlama yaklaşımı, gerçek hayatta var olan problemi çözmek üzere tasarlanacak modelin, sınıflarlar yardımıyla oluşturulan nesneler ve bu nesneler arasındaki ilişkilerden faydalanılarak kurulmasını ilke edinmiştir.



Nesne tabanlı programlamada kullanılacak nesneler sınıflar yardımıyla oluşturulmaktadır.

Sınıf ve nesne bu programlama dili modelinin en önemli iki yapı taşıdır. Kullanılacak nesneler sınıflar yardımıyla oluşturulmaktadır. Sınıflar, nesneleri oluşturmak için önceden oluşturulmuş bir kalıp veya ilk örnek (prototip) olarak da düşünülebilir. Bu yaklaşıma göre bir nesne, özellikleri birbiri ile benzemeyen nitelik ve davranışlardan meydana gelen bir veri alanı olarak tanımlanabilir.

Nesne, içinde veri ve bu veriler üzerinde işlem yapacak olan metotları (fonksiyon) bulunduran yazılım bileşenidir. Nesne herhangi bir uygulamada tekrar tekrar kullanılabilir. Nesneler, sınıfların içinde tanımlanır. Sınıfta, nesneyi oluşturan üye değişkenler ve metotlar açıkça tanımlanır.

Bu ünite kapsamında sınıf ve nesne kavramlarının kısaca ne olduğu hatırlatılarak bu yapılar arasındaki ilişkilerden bahsedilmiştir. Üniteye yer alan bütün kod örnekleri Visual Studio 2019 geliştirme ortamında C# programlama dili kullanılarak yazılmıştır.

SINIF VE NESNE KAVRAMLARI

Sınıf Kavramı

Sınıf (Class), nesne yönelimli programlamanın en önemli ögesidir. Sınıflar sayesinde program birkaç bölüme ayrılır ve karmaşıklıkları azalır (İTÜ/BİDB, 2021).

Oluşturulan yöntemler ve özellikler bir sınıftır ve bir sınıf birden çok kez kullanılabilir. Bir sınıf, içinde oluşturulacak nesnenin bir dizi üyesini içermelidir. Bu üyelere örnek olarak özellikler, metotlar, yapıcılar, olaylar sıralanabilir. Sınıf aynı zamanda nesne için bir şablon görevi görür. Bir başka ifadeyle sınıf, nesnenin durumuyla ilgili işlemleri ve nitelikleri tanımlar. Ortak niteliklere sahip nesnelere ait veriler ve işlemler bir sınıfta toplanır. Program içindeki nesneleri tanımlamak için bu tür bir sınıf yapısı sıklıkla kullanılır.



Sınıfta tanımlanan metot ve değişkenlere sınıfın üyeleri denir.

Sınıf kavramının en büyük faydası, bir kez kodlama yapılarak her zaman kullanılabilir nesneler oluşturma olanağı sağlamasıdır. Bir sınıftan nesne sayıda nesne, örnek olarak oluşturulabilir. Oluşturulan tüm nesneler, daha önce sınıfta belirlenen özelliklere ve davranışlara sahip olacaktır.

Sınıfın hangi özelliklere sahip olacağı, yani hangi bilgileri saklayacakları, sınıfta tanımlanan alanlar (üye değişkenler) yardımı ile belirlenebilir. Sınıfların hangi davranışlara sahip olacağı veya hangi işlevleri yerine getirecekleri ise sınıf kapsamında tanımlanan metotlar (yöntemler) ile belirlenir (Uyanık, 2017).



Örnek

- Örnek bir sınıf tanımlamasının aşağıdaki gibi yapıldığını hatırlayalım.
- `class sınıfAdı`
`{`
`.... //Sınıf gövdesi (Sınıf bloğu)`
`}`

Nesne Kavramı

Nesne tabanlı programlamada nesneler sınıflardan türetilir (Hüsem, 2018).

Bu nedenle nesneler, sınıfların aksine canlıdır ve kimlikleri vardır. Aynı sınıftan üretilen iki nesnenin özellikleri benzerdir ancak bu özelliklerin değerleri farklı olabilir. Bir ifadeyi nesne yapmak için bellekte bir konum belirtmelidir. Sınıflar kullanarak uygulamadaki nesneleri tanımlanır ve bu nesnelere mesaj gönderilir. Gönderilen mesaja göre yeni bir nesneye ihtiyaç duyulduğunda bu nesne oluşturulur. Yanıt mesajı sınıfın işidir.

Bir nesne, kendisi ve gerçekleştirebileceği işlemler hakkında bilgi depolar. Programın gerektirdiği tüm veriler nesne tarafından kaydedilir. Gerçekleştirilen işlemin türüne bağlı olarak, nesne farklı bilgiler içerir.



Örnek

- Örnek bir nesne tanımlamasının aşağıdaki gibi yapıldığını hatırlayalım.
- `sınıfAdı nesneAdı = new sınıfAdı();`

Nesne tabanlı programlamada bir nesneyi modellemek için ilk başta sınıf oluşturulur. Daha sonra sınıfa ait tüm özellik ve fonksiyonlar sınıf içerisine eklenir.

SINIF VE NESNE İLİŞKİSİ

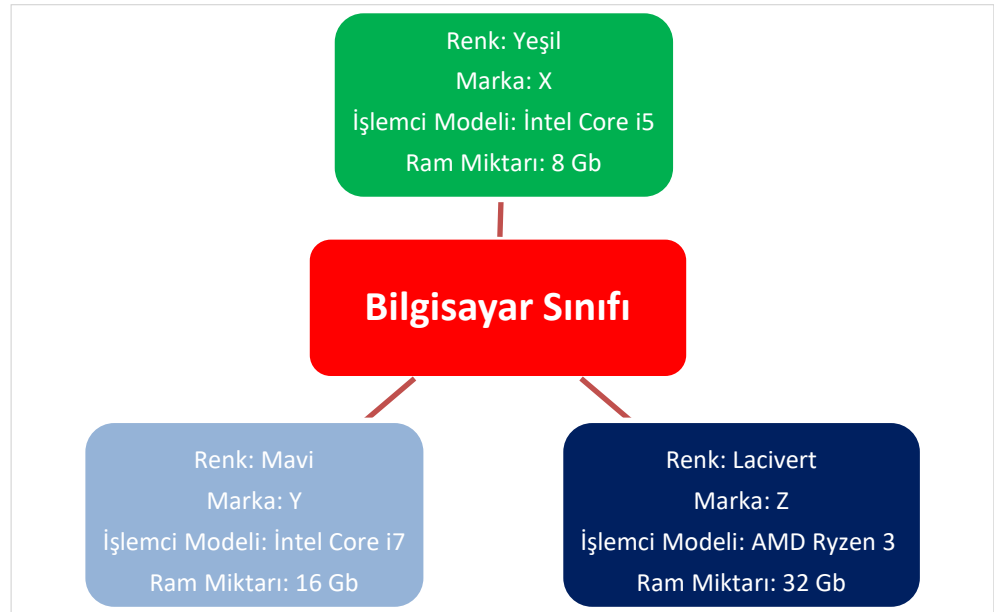
Sınıf, soyut bir veri tipidir. Nesne ise sınıftan türetilen somutlaşan bir varlıktır. Sınıflar, nesnelerde olması gereken özelliklerden durum ve davranışları içerir. Bu sınıftan oluşturulan örneğe (instance) de nesne denildiği önceki nesne konusunda bahsedilmişti. Dolayısıyla bir sınıftan birçok nesne üretilebilmektedir.



Şekil 5.1. Sınıf ve Türetilen Nesneler

Sınıf ve nesne arasındaki ilişkide en önemli noktalardan birisi sadece aynı sınıftan oluşturulan nesnelerin tipleri aynı olabilmektedir. Örneğin bir bilgisayar sınıfından üretilen nesnenin özellikleri ile bir ev sınıfından üretilen nesnenin özellikleri birbirinden farklıdır.

Bu kısımda dikkat edilmesi gereken önemli bir nokta ise, aynı sınıftan üretilen nesneler aynı ortak özelliklere sahip olabilirler ancak bu özelliklerin değerleri farklıdır. Örnek olarak bir bilgisayar sınıfından üretilen renk, marka, işlemci modeli veya ram miktarı özellikleri aynı olsa da bu özelliklerin değerleri birbirinden farklıdır. Şekil 5.2’de örnek olarak bu bilgisayar sınıfı ve nesneleri gösterilmiştir.



Şekil 5.2. Örnek Sınıf ve Türetilen Nesneler



Değişkenler ad, soyad gibi bilgileri saklamaktadır. Metotlar ise, kullanıcı kaydı gibi bir görevi yerine getiren alt programlardır.

Tablo 5.1. Sınıf ve Nesne Arasında Karşılaştırma

Karşılaştırma için temel	Sınıf	Nesne
Tanım	Bir veya daha fazla nesnenin oluşturulduğu şablon ya da plan olarak bilinmektedir.	Bir sınıfın örneği olarak bilinmektedir.
Varlık türü	Mantıksal	Fiziksel
Oluşturma	Class, sınıf anahtar sözcüğü kullanılarak bildirilmektedir.	Nesne yeni anahtar kelime tarafından çağrılmaktadır.
Bellek ayırma	Bir sınıfın oluşumu hafıza ayırmamaktadır.	Nesnenin oluşturulması hafızayı tüketmektedir.
Değişkenler	Bir türdür.	Bir değişkendir.
Oluşturma	Anahtar sözcüğü kullanılarak bildirilmektedir.	Yeni anahtar kelime tarafından çağrılmaktadır.
Amaç	Veri gruplandırmasını sağlamak	Veri soyutlama ve daha fazla kalıtım
Tip	Değer	Referans



Bireysel Etkinlik

- Sizler de sınıf ve nesne arasındaki bu ilişkiye dikkat ederek başka örnekler oluşturunuz.

Sınıf ve Nesne İlişkisine Örnekler

Buraya kadar öğrendiklerimizi pekiştirmek için uygulamalar geliştirerek ilerleyelim. *Örnek uygulamaları geliştirmek için Visual Studio 2019 ortamında Windows Forms proje türü seçilerek C# programlama dili kullanılmıştır.*

İlk olarak şekil 5.2'dekine benzer bir sınıf nesne ilişkisi uygulamaya dökelim. Bunun için aşağıdaki adımları izlememiz gerekmektedir

Adım 1: Bilgisayar adında bir sınıf oluşturulur.

Adım 2: Bilgisayar sınıfında kullanılacak ilgili değişkenler tanımlanır.

Adım 3: Bilgisayar sınıfından 3 adet nesne türetilir.

```
public class Bilgisayar
{
    public string renk;
    public string marka;
    public string islemciModeli;
    public int ramMiktari;
}

private void btnTuretGuncelle_Click(object sender, EventArgs e)
{
    //nesne türetimi
    Bilgisayar bilgisayar1 = new Bilgisayar();
    bilgisayar1.renk = "Yeşil";
    bilgisayar1.marka = "X";
    bilgisayar1.islemciModeli = "Intel Core i5";
    bilgisayar1.ramMiktari = 8;

    //değer atama
    lblRenk1.Text = bilgisayar1.renk;
    lblMarka1.Text = bilgisayar1.marka;
    lblModel1.Text = bilgisayar1.islemciModeli;
    lblRam1.Text = bilgisayar1.ramMiktari.ToString();
    //ToString() kullanımı ile int değeri string değere dönüştürülür

    Bilgisayar bilgisayar2 = new Bilgisayar();
    bilgisayar2.renk = "Mavi";
    bilgisayar2.marka = "Y";
    bilgisayar2.islemciModeli = "Intel Core i7";
    bilgisayar2.ramMiktari = 16;

    lblRenk2.Text = bilgisayar2.renk;
    lblMarka2.Text = bilgisayar2.marka;
    lblModel2.Text = bilgisayar2.islemciModeli;
    lblRam2.Text = bilgisayar2.ramMiktari.ToString();

    Bilgisayar bilgisayar3 = new Bilgisayar();
    bilgisayar3.renk = "Lacivert";
    bilgisayar3.marka = "Z";
    bilgisayar3.islemciModeli = "AMD Ryzen3";
    bilgisayar3.ramMiktari = 32;

    lblRenk3.Text = bilgisayar3.renk;
    lblMarka3.Text = bilgisayar3.marka;
    lblModel3.Text = bilgisayar3.islemciModeli;
    lblRam3.Text = bilgisayar3.ramMiktari.ToString();
}
```



Sınıf ve nesne ilişkilerinde hangi erişim belirleyicisinin kullanıldığı çok önemli bir husustur.

Bilgisayar1	Bilgisayar2	Bilgisayar3
Renk : Renk1	Renk : Renk2	Renk : Renk3
Marka : Marka1	Marka : Marka2	Marka : Marka3
İşlemci Modeli : iModel1	İşlemci Modeli : iModel2	İşlemci Modeli : iModel3
Ram Miktarı : Ram1	Ram Miktarı : Ram2	Ram Miktarı : Ram3

Resim 5.1. Sınıf ve Nesne ilişkisi Örnek Uygulama

Resim 5.1’de örnek bir bilgisayar sınıfından türetilen 3 adet bilgisayar nesnesi gösterilmektedir.

Bilgisayar1	Bilgisayar2	Bilgisayar3
Renk : Yeşil	Renk : Mavi	Renk : Lacivert
Marka : X	Marka : Y	Marka : Z
İşlemci Modeli : Intel Core i5	İşlemci Modeli : Intel Core i7	İşlemci Modeli : AMD Ryzen3
Ram Miktarı : 8	Ram Miktarı : 16	Ram Miktarı : 32

Resim 5.2. Nesne Türetme ve Değer Atama

Yapılan uygulamada “Nesneleri Türet ve Değerleri Güncelle” butonu yardımıyla nesneler türetilmekte ve her bir nesneye değer ataması yapılmaktadır. Resim 5.2’de gösterildiği üzere her bir bilgisayar nesnesinin değerlerinin farklı olduğuna dikkat ediniz.



Örnek

- Şekil 5.4’de gösterilen örnek uygulamanın "Temizle" fonksiyonu için `lblRenk1.Text = string.Empty;` şeklinde bir kod yazılabilir.



Bireysel Etkinlik

- Bütün nesnelerin özelliklerine verilen değerleri topluca temizlemek için nasıl bir yol izlenmelidir? Değer temizleme işlemini düşünerek örnek uygulamaya ekleyiniz.

Gerçek dünyadaki bir nesne temel olarak iki bölümden oluştuğunu biliyoruz. Bunlar özellikler (attributes) ve davranışlar (behaviors) dır.

Tablo 5.2. Örnek Nesneler Özellikleri ve Davranışları

Nesne	Özellikler	Davranışlar
Bilgisayar	Renk, Marka, İşlemci Modeli, Ram Miktarı vb.	Bilgisayarı açma Bilgisayarı kapatma Ses açma vb.
Kuş	Tüy rengi, Kanat genişliği, Gaga büyüklüğü, vb.	Uçmak, Kanat çırpma, Ötmek, vb.
Uçak	Kanat uzunluğu, Ağırlığı, Kapasitesi, Motor gücü vb.	İniş yapma, Kalkış yapma, Yakıt doldurma vb.



Metotlar, nesnelere işlevsellik sağlayan yapılardır.

Yazılım dünyasında ise nesnelerin modellerine sınıf (class), özelliklere veri/alan (data/field), davranışlara ise metot/fonksiyon (method/function) denilmektedir. Söz konusu bu yapıları daha net anlayabilmek için aşağıda verilen örneği inceleyelim. Otomatik vites bir arabamız olduğunu düşünelim. Şimdi arabamızın çok basit bir modelini tasarlayalım. Bunun için öncelikle bir otomobil sınıfı oluşturalım. Daha sonra sınıfta yer almasını istediğimiz özellik ve metotları belirleyelim.

Sınıfımıza ait özellikler aşağıdaki gibi olsun:

- Hız bilgisi,
- Vites bilgisi,
- Ulaşılabilecek maksimum hız bilgisi.

Sınıfımıza ait metotlarımız ise aşağıdaki gibi olsun:

- Gaza basmak,
- Frene basmak,
- Vites göstermek,
- Hız göstermek.

```

public class Araba //class tanımlama
{
    int vites; //özellik
    int hiz; //özellik
    int maksimumHiz = 100; //özellik

    public void VitesGoster() //metot (fonksiyon)
    {
        MessageBox.Show("Vites: "+vites.ToString());
        //veya herhangi bir labelle yazdırabiliriz
    }
    public void HizGoster() //metot (fonksiyon)
    {
        MessageBox.Show("Hız: "+hiz.ToString());
        //veya herhangi bir labelle yazdırabiliriz
    }
    public void GazaBas() //metot (fonksiyon)
    {
        if (hiz < maksimumHiz)
        {
            hiz = hiz + 10;
            VitesDegistir();
        }
    }
    public void FireneBas() //metot (fonksiyon)
    {
        if (hiz >= 0)
        {
            hiz = hiz - 10;
            VitesDegistir();
        }
    }
    private void VitesDegistir() //metot (fonksiyon)
    {
        if (hiz > 80) { vites = 5; }
        else if (hiz > 60) { vites = 4; }
        else if (hiz > 40) { vites = 3; }
        else if (hiz > 20) { vites = 2; }
        else if (hiz > 0) { vites = 1; }
        else { vites = 0; }
    }
}

```



Fonksiyonları tanımlarken public ve private şeklinde erişim belirleyiciler kullanıldığına dikkat ediniz.

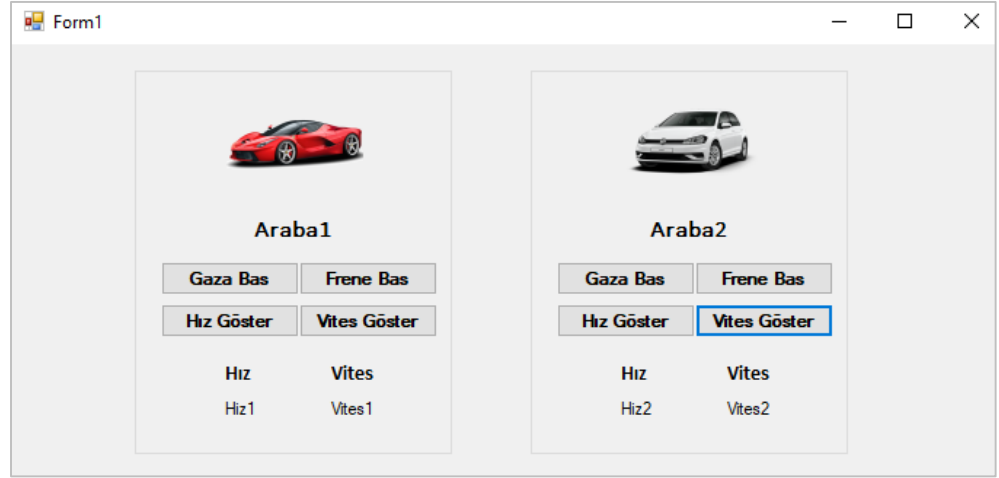
Yukarıdaki kod blokları ile istenildiği gibi bir araba modeli tasarlanmıştır. Oluşturulan araba sınıfından nesne türetmek için ise **new** anahtar sözcüğü kullanılmalıdır. Aşağıda araba sınıfından “araba1” ve “araba2” adında 2 adet nesne türetilmiştir.

Sonrasında oluşturulan araba sınıfından “araba1” ve “araba2” adında 2 adet nesne aşağıdaki gibi türetilmiştir.

```
Araba araba1 = new Araba();
```

```
Araba araba2 = new Araba();
```

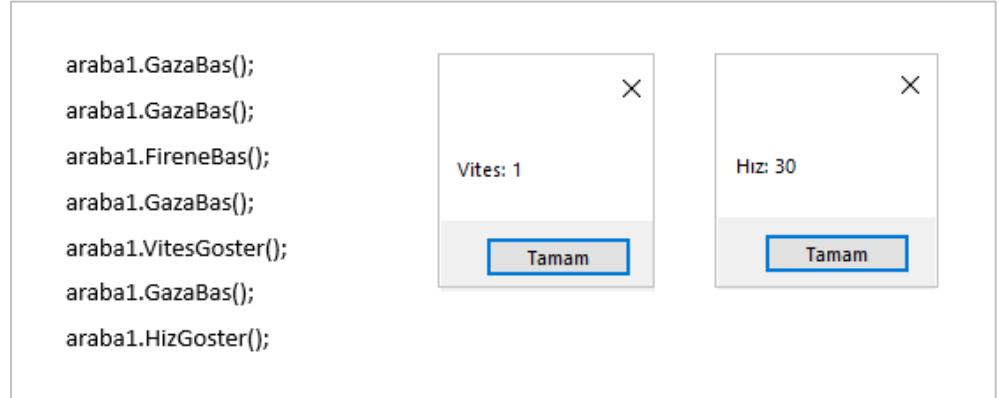
Tasarladığımız formun ekran görüntüsü ise Resim 5.3’de gösterildiği gibidir.



Resim 5.3. Sınıf ve Nesne İlişkisi Örnek Uygulama 2

Daha sonra ise ister form ekranında bulunan butonlar yardımıyla isterse de kod blokları yardımıyla formumuz üzerinde özellik ve metotlarımızı test edebiliriz.

Örneğin aşağıdaki gibi bir kod bloğu yazalım:

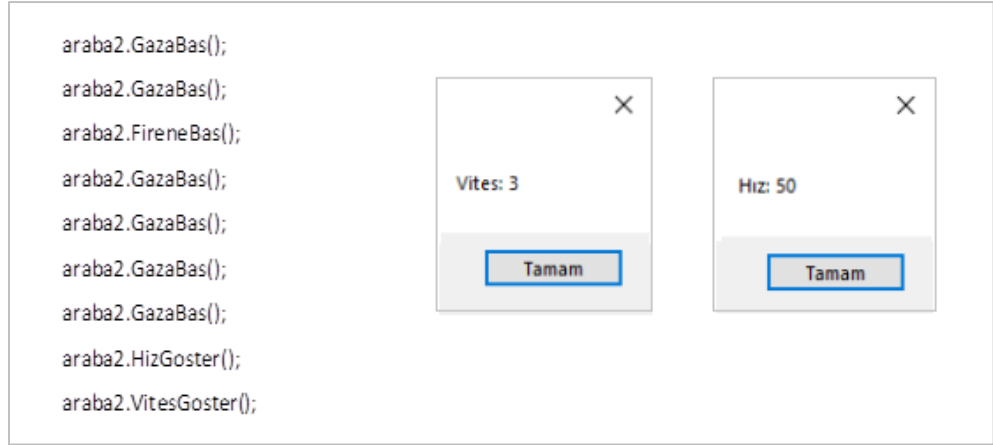


Resim 5.4. Örnek Uygulama 2’ye Ait Örnek Ekran Çıktısı

Resim 5.4’de “araba1” adında araba sınıfından türetilen nesneye ait örnek kod bloğu çalıştırılarak ekran çıktıları gösterilmiştir.



Birden fazla nesnenin aynı metot üzerinden çalışılabildiğine dikkat ediniz.



Resim 5.5. Örnek Uygulama 2'ye Ait Örnek Ekran Çıktısı

Resim 5.5'de "araba2" adında araba sınıfından türetilen nesneye ait örnek kod bloğu çalıştırılarak ekran çıktıları gösterilmiştir.



Örnek

- Resim 5.4 ve 5.5'deki gibi örnek uygulamaları çalıştırmak için form üzerindeki butonlar da kullanılabilir. Örnek bir gaza bas butonu aşağıdaki gibidir:
- ```
private void btnGazaBas_Click(object sender, EventArgs e)
{
 araba1.GazaBas();
}
```



### Bireysel Etkinlik

- Siz de bu ünite de anlatılan sınıf ve nesne ilişkisini kullandığınız örnek uygulamalar yapınız. Daha sonra bu ünite de anlatılanlar sayesinde elde ettiğiniz yeteneklerinizi gözlemleyiniz.



## Özet

### •GİRİŞ

- Nesne tabanlı programlama, önceki ünitelerde de belirtildiği üzere yazılım tasarımını işlevler ve mantık yerine, sınıf ve nesneler etrafında düzenleyen bir programlama yaklaşımıdır. Nesne tabanlı programlama, programcıların işlemleri için gereken mantıktan ziyade işlem yapacağı nesnelere odaklanır. Nesne tabanlı programlamada büyük, karmaşık ve aktif olarak güncellenen veya bakımı yapılan programlar için çok uygundur. Bu yöntem işbirlikçi geliştirmeye de elverişlidir ve projeler böylece gruplandırılabilir.
- Nesne tabanlı programlama yaklaşımı, gerçek hayatta var olan problemi çözmek üzere tasarlanacak modelin, sınıflar yardımıyla oluşturulan nesneler ve bu nesneler arasındaki ilişkilerden faydalanılarak kurulmasını ilke edinmiştir.

### •SINIF VE NESNE KAVRAMLARI

#### •Sınıf Kavramı

- Sınıf (Class), nesne yönelimli programlamanın en önemli ögesidir. Sınıflar sayesinde program birkaç bölüme ayrılır ve karmaşıklıkları azalır (İTÜ/BİDB, 2021). Oluşturulan yöntemler ve özellikler bir sınıftadır ve bir sınıf birden çok kez kullanılabilir. Bir sınıf, içinde oluşturulacak nesnenin bir dizi üyesini içermelidir. Bu üyelere örnek olarak özellikler, metotlar, yapıcılar, olaylar sıralanabilir. Sınıf aynı zamanda nesne için bir şablon görevi görür. Bir başka ifadeyle sınıf, nesnenin durumuyla ilgili işlemleri ve nitelikleri tanımlar. Ortak niteliklere sahip nesnelere ait veriler ve işlemler bir sınıfta toplanır. Program içindeki nesneleri tanımlamak için bu tür bir sınıf yapısı sıklıkla kullanılır.

#### •Nesne Kavramı

- Nesne tabanlı programlamada nesneler sınıflardan türetilir (Hüsem, 2018). Bu nedenle nesneler, sınıfların aksine canlıdır ve kimlikleri vardır. Aynı sınıftan üretilen iki nesnenin özellikleri benzerdir ancak bu özelliklerin değerleri farklı olabilir. Bir ifadeyi nesne yapmak için bellekte bir konum belirtmelidir. Sınıflar kullanılarak uygulamadaki nesneleri tanımlanır ve bu nesnelere mesaj gönderilir. Gönderilen mesaja göre yeni bir nesneye ihtiyaç duyulduğunda bu nesne oluşturulur. Yanıt mesajı sınıfın işidir. Bir nesne, kendisi ve gerçekleştirebileceği işlemler hakkında bilgi depolar. Programın gerektirdiği tüm veriler nesne tarafından kaydedilir. Gerçekleştirilen işlemin türüne bağlı olarak, nesne farklı bilgiler içerir..

### •SINIF VE NESNE İLİŞKİSİ

- Sınıf, soyut bir veri tipidir. Nesne ise sınıftan türetilen somutlaşan bir varlıktır. Sınıflar, nesnelerde olması gereken özelliklerden durum ve davranışları içerir. Bu sınıftan oluşturulan örneğe (instance) de nesne denildiği önceki nesne konusunda bahsedilmişti. Dolayısıyla bir sınıftan birçok nesne üretilebilmektedir.
- Sınıf ve nesne arasındaki ilişkide en önemli noktalardan birisi sadece aynı sınıftan oluşturulan nesnelerin tipleri aynı olabilmektedir. Örneğin bir bilgisayar sınıfından üretilen nesnenin özellikleri ile bir ev sınıfından üretilen nesnenin özellikleri birbirinden farklıdır.
- Bu kısımda dikkat edilmesi gereken önemli bir nokta ise, aynı sınıftan üretilen nesneler aynı ortak özelliklere sahip olabilirler ancak bu özelliklerin değerleri farklıdır. Örnek olarak bir bilgisayar sınıfından üretilecek renk, marka, işlemci modeli veya ram miktarı özellikleri aynı olsa da bu özelliklerin değerleri birbirinden farklıdır
- Sınıf ve Nesne İlişkisine Örnekler
- Buraya kadar öğrendiklerimizi pekiştirmek için uygulamalar geliştirerek ilerleyelim. Örnek uygulamaları geliştirmek için Visual Studio 2019 ortamında Windows Forms proje türü seçilerek C# programlama dili kullanılmıştır.

## DEĞERLENDİRME SORULARI

1. Aşağıdakilerden hangisi bir sınıftan nesne oluşturma işlemine denilmektedir?
  - a) Örnekleme
  - b) Özellik
  - c) Yöntem
  - d) Sınıf
  - e) Nesne
2. Nesne tabanlı programlamada bir sınıf oluşturmak için aşağıdakilerden hangisi kullanılmalıdır?
  - a) new
  - b) class
  - c) this
  - d) object
  - e) attributes
3. Nesne tabanlı programlamada bir nesne oluşturmak için aşağıda verilen hangi kod bloğu kullanılmalıdır?
  - a) new nesneAdı()
  - b) class sınıfAdı
  - c) sınıfAdı nesneAdı = this sınıfAdı();
  - d) sınıfAdı nesneAdı = new sınıfAdı();
  - e) nesneAdı sınıfAdı=new nesneAdı();

“Nesne için bir şablon görevi görür. Ortak niteliklere sahip nesnelere ait veriler ve işlemler burada toplanır.”

4. Yukarıdaki bilgide bahsedilen kavram aşağıdakilerden hangisidir?
  - a) Nesne
  - b) Sınıf
  - c) Özellik
  - d) Davranış
  - e) Alan

“Nesne tabanlı programlamada sınıflardan türetilirler. Sınıfların aksine canlıdır ve kimlikleri vardır.”

5. Yukarıdaki bilgide bahsedilen kavram aşağıdakilerden hangisidir?
  - a) Fonksiyon
  - b) Özellik
  - c) Nesne
  - d) Erişim belirleyici
  - e) Üye değişkenler

6. Aşağıdakilerden hangisi sınıflar için yanlıştır?
- a) Sınıflar sayesinde program birkaç bölüme ayrılır.
  - b) Bir sınıfın oluşumu hafıza ayırmaz.
  - c) Bir kez kodlama yapılarak her zaman kullanılabilecek nesneler oluşturma olanağı sağlar.
  - d) Yeni (new) anahtar kelime tarafından çağrılır.
  - e) Program içindeki nesneleri tanımlamak sıklıkla kullanılır.
7. Aşağıdakilerden hangisi nesneler için yanlıştır?
- a) Bir sınıfın örneği olarak bilinmektedir.
  - b) Gerçekleştirilen işlemin türüne bağlı olarak, farklı bilgiler içerir.
  - c) Bir nesnenin oluşumu hafıza ayırmamaktadır.
  - d) Yeni (new) anahtar kelime tarafından çağrılmaktadır.
  - e) Herhangi bir uygulamada tekrar tekrar kullanılabilir.
8. Sınıf ve nesne ilişkisi için aşağıdakilerden hangisi doğrudur?
- a) Sınıfın varlık türü fiziksel iken nesneninki mantıksaldır.
  - b) Her ikisinin de oluşumu için hafıza ayrılmaktadır.
  - c) Sınıf bir tür iken nesne bir değişkendir.
  - d) Her sınıftan sadece bir adet nesne üretilebilir.
  - e) Sınıflar nesneden türetilirler.

“Yazılım dünyasında ise nesnelerin modellerine ....., özelliklere ....., davranışlara ise ..... denilmektedir.”

9. Yukarıda boşluk bırakılan yerlere sırasıyla aşağıdakilerden hangisi gelmelidir?
- a) sınıf - veri/alan - metot/fonksiyon
  - b) sınıf - nesne - metot/fonksiyon
  - c) nesne - sınıf - veri/alan
  - d) nesne - veri/alan - sınıf
  - e) örnekleme - alan - belirleyici
10. Aşağıdakilerden hangisi örnek bir araba sınıfının özelliği olamaz?
- a) Hız bilgisi
  - b) Yakıt bilgisi
  - c) Ulaşılabilecek maksimum hız
  - d) Gaza basmak
  - e) Üretim yılı

**Cevap Anahtarı**

1.a, 2.b, 3.d, 4.b, 5.c, 6.d, 7.c, 8.c, 9.a, 10.d

## YARARLANILAN KAYNAKLAR

- Hüsem, H. (2018). Sınıf ve nesne. 27 Eylül 2021 tarihinde [https://teknikakil.com/c-sharp/nesne-tabanli-programlama/sinif-ve-nesne-kavrami/adresinden erişildi](https://teknikakil.com/c-sharp/nesne-tabanli-programlama/sinif-ve-nesne-kavrami/adresinden-erisildi).
- İTÜ/BİDB. (2021). Sınıf ve nesne kavramları. 27 Eylül 2021 tarihinde [https://bidb.itu.edu.tr/sevir-defteri/blog/2013/09/08/s%c4%b1n%c4%b1f-\(class\)-ve-nesne-\(object\)-kavramlar%c4%b1](https://bidb.itu.edu.tr/sevir-defteri/blog/2013/09/08/s%c4%b1n%c4%b1f-(class)-ve-nesne-(object)-kavramlar%c4%b1) adresinden erişildi.
- Uyanık, D. (2017). Sınıf (Class) mantığı. 25 Eylül 2021 tarihinde <http://docplayer.biz.tr/45158887-1-siniflar-sinif-class-mantigi.html> adresinden erişildi.