

Class(Sınıf) ve Object(Nesne)

- Class soyut bir veri tipidir. Object ise onun somutlaşan bir cismidir.
- Sınıf kavramı belli bir veya daha çok ortak özellikleri olan varlıkları tanımlamaya yarayan soyut bir kavramdır. Nesne ise sınıfa ait bir varlıktır.
- Classlar oluşturulurken sadece özellikler değil metotlar da tanımlanabilir.
- Object ise class yapısını kullanarak üretilen bir varlıktır.

Sınıf ve Nesne Oluşturmak

- `class sinifAdi {`

`}`

```
sinifAdi nesneAdi=new sinifAdi();  
sinifAdi nesneAdi2=new sinifAdi();
```

Öğelere Erişim

- `class sinifAdi {`

`}`

`sinifAdi nesneAdi=new sinifAdi();`

`nesneAdi.ozellik=değer;`

`nesneAdi2.ozellik=değer2;`

Sınıf ve Nesne Oluşturmak

- ```
class Ev {
 string cadde;
 string sokak;
 int kapiNo;
}
```

```
Ev alininEvi=new Ev();
alininEvi.cadde=«Menekşe»;
alininEvi.sokak=«Gül»;
alininEvi.kapiNo=15;
Ev fatmaninEvi=new Ev();
```



# Kapsülleme (Encapsulation )

- Sınıfa ait öğelere (özellik, metot) dışardan yetkisiz erişimlerin engellenmesi için sınıfın dış dünyaya kapatılmasıdır.
- Public erişilmek isteneni genel yapar ve dış dünyaya açar.

# Static ve Dynamic

- Static öğeler bir sınıfa ait nesne oluşturulmadan erişilebilen öğelerdir. Böylece onlara programın her yerinden erişilebilir (public gibi..)
- Static öğeler için bellekte tek yer ayrılır ve nesnelerin oluşturulmasından, yok edilmesinden etkilenmezler.
- Static olmayan öğelere dinamik öğeler denir ve her biri için bellekte yer ayrılır.
- Dinamik öğeler için nesne oluşturmak gerekirken, static öğeler doğrudan çağrılabilir.

# Access Modifiers

- C# OOP bir dil olduğu için herşey sınıflar altında tanımlanır. Sınıflara ve sınıf öğelerine erişim kısıtlanabilir veya belli düzeyde erişime izin verilebilir. Bu yetki tanımlayan anahtarlara access modifiers denir.
  - public: Erişim kısıtı yoktur, her yerden erişilebilir.
  - protected: Ait olduğu sınıftan veya o sınıftan türetilen sınıflardan erişilebilir.
  - internal: Bulunduğu assembly'nin (Dll veya Exe dosyası) içinde erişilebilirdir. Dll veya Exe dosyasının içerisinde erişim için kısıtlama yoktur, ama dışarıdan erişilemez.
  - protected internal: Etkin projeye ait sınıflardan ve türevlerinden erişilebilir.
  - private: Yalnızca bulunduğu sınıf içinden erişim sağlanabilir.
- Namespace erişim belirteci almaz, daima publictir.
- Sınıflar public veya internal olabilirler ama protected ve private olamazlar.

# Metotlar

- Matematikteki fonksiyon kavramına karşılık gelmektedir.
  - $F : x \rightarrow y, \{x \rightarrow y\}$
- Belirli görevleri olan ve bu görevler sonucuna göre akışı belirleyen program parçalarıdır.

```
public int kareBul(int x) {
 int kare =x*x;
 return kare;
}
public void kareYaz(int kare) {
 Console.WriteLine(kare);
}
```



# Uygulama

1. öğrenci sınıfı oluşturun ve bu sınıf altında no, ad, soyad ve vize özelliklerini oluşturun. Bu sınıfı kullanarak 3 öğrenci nesnesi oluşturun ve özelliklerine atama yapın ve ekranda yazdırın. Vize sınavı için sınıf ortalamasını bulun.
2. Stok sınıfı oluşturun ve bu sınıf altında ürün adı, ürün türü, ürün sayısı özelliklerini oluşturun. Bu sınıfı kullanarak 3 nesne oluşturun, özelliklerine atama yapın ve ürün sayısını ekranda yazdırın. En az ürün sayısı 30 olarak düşünüp ürün sayılarının durumunu (stok eksik, stok yeterli gibi) yazdırın.
3. Girilen iki sayı arasında toplama, çıkarma, çarpma ve bölme yaptıran yöntemleri yazıp ekranda yazdırın.
4. static öğeler kullanarak 4 işlem yapan class ve class a ait yöntemleri yazınız.
5. Ürün fiyatı ve kdv oranı verildiğinde vergili fiyatını bulan class ve yöntemleri access modifier kullanarak yazınız.

# Kaynak

- V. Aktaş, *Her Yönüyle C#*. İstanbul: Kodlab, 2017.
- T. Karaçay ve A. Karaçay, *Hiç Bilmeyenler İçin C# ile Programlamaya Giriş*, 2. Baskı. Ankara: Seçkin Yayıncılık, 2016.
- Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.