

MALATYA  
TURGUT ÖZAL ÜNİVERSİTESİ  
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ  
ALGORİTMA VE PROGRAMLAMA 2

**SINIFLAR VE NESNELER-3-UYGULAMA**

# UYGULAMA 1:

- **zaman** adında bir sınıf oluşturunuz.
- Bu sınıfta **int** tipinde **private** olan **saat**, **dakika** ve **saniye** adında 3 değişken bulunsun.
- Sınıfın **zmn** adında bir nesnesi olsun ve bu nesne **sınıfın bitiş parantezinde** tanımlansın.
- Sınıfa **arkadaş** olan **void** tipinde ve **3 adet int parametre** alan **void zaman\_al(int s, int dk, int sn)** adında bir fonksiyon tanımlayın ve bu fonksiyon sınıfa arkadaş olsun.
- **Sınıf içinde** ise **void** tipinde **goster** adında bir fonksiyon oluşturun.
- **Main()** içinde **3 farklı int** değişken oluşturun ve bu değişkenlere **kullanıcıdan saat,dakika,saniye bilgilerini girmesini isteyin.**
- Daha sonra yine **main** içinde girilen bu üç değişkeni **zaman\_al** fonksiyonuna gönderin ve fonksiyon içinde sınıfın saat,dakika,saniye bilgilerine kullanıcının girdiği bu üç değeri atayın.
- Daha sonra yine **main** içinde sınıfın **goster** fonksiyonunu çağırın. Göster fonksiyonu içinde ekrana kullanıcının girdiği bilgileri saat:dakika:saniye şeklinde ekrana yazdırın. Örnek ekran çıktısı aşağıdaki gibi olabilir.

```
saati giriniz:14  
dakika giriniz:28  
saniye giriniz:19  
  
saat=14:28:19
```

# UYGULAMA 1: KODLAR

```
1  #include <iostream>
2  using namespace std;
3
4  class zaman{
5      int saat,dakika, saniye;
6
7      public:
8          friend void zaman_al(int s,int dk,int sn);
9
10         void goster(){
11             cout<<"saat="<<saat<<":"<<dakika<<":"<<saniye;
12         }
13     }zmn;
14
15     void zaman_al(int s, int dk, int sn){
16         zmn.saat=s;
17         zmn.dakika=dk;
18         zmn.saniye=sn;
19     }
20
21     int main() {
22
23         int s,dk,sn;
24         cout<<"saati giriniz:";
25         cin>>s;
26         cout<<"dakika giriniz:";
27         cin>>dk;
28         cout<<"saniye giriniz:";
29         cin>>sn;
30
31         zaman_al(s,dk,sn);
32         cout<<endl<<endl;
33         zmn.goster();
34
35         return 0;
36     }
```

## UYGULAMA 2:

```
1  #include <iostream>
2  using namespace std;
3
4  class dikdortgen{
5      int uzunkenar,kisakenar;
6
7      public:
8          friend void goster(dikdortgen d);
9
10         dikdortgen(){
11             uzunkenar=10;
12             kisakenar=5;
13         }
14
15         dikdortgen(int uzun,int kısa){
16             uzunkenar=uzun;
17             kisakenar=kisa;
18         }
19     };
20
21 void goster(dikdortgen d){
22     cout<<"Dikdortgenin alani="<<d.uzunkenar*d.kisakenar<<endl;
23     cout<<"Dikdortgenin cevresi="<<(d.uzunkenar+d.kisakenar)*2<<endl;
24 }
25
26 int main() {
27     int uzun, kısa;
28     cout<<"uzun kenari girin:";
29     cin>>uzun;
30     cout<<"kisa kenari girin:";
31     cin>>kisa;
32
33     dikdortgen x(uzun,kisa);
34     goster(x);
35
36     cout<<endl<<"Kullaniciдан deger almadan elde edilen sonuc"<<endl;
37     dikdortgen y;
38     goster(y);
39
40     return 0;
41 }
42 }
```

- Bu örnekte dikdortgen adında sınıf oluşturduk. Private uzunkenar ve kisakenar adlı değişkenleri tanımladık. Ayrıca 2 adet yapıcı fonksiyon ekledik.
- goster adında bir fonksiyon oluşturduk ve parametre olarak dikdortgen sınıfı tipinde bir nesne kabul etmesini istedik.
- Daha sonra main içinde kullanıcıdan uzun ve kısa kenarları girmesini istedik.
- X adında sınıfın nesnesini oluşturduk ve daha sonra sınıfın private olan uzunkenar ve kisakenar değişkenlerine erişmek için yapıcı fonksiyona parametre olarak gönderdik.  
*dikdortgen x(uzun,kisa);*
- Goster fonksiyonuna x nesnesini gönderdik ve fonksiyon içinde alan ve çevresi hesaplanıp ekrana yazdırıldı.
- Daha sonra y adında bir nesne oluşturduk. Bu nesne için kenar bilgileri girmedığımız için otomatik olarak dikdortgen() yapıcı fonksiyonu işleyecektir.
- Daha sonra y nesnesinide goster fonksiyonuna gönderdik ve onunda alan ve çevre bilgileri hesaplanıp ekrana yazdırıldı.

```
uzun kenari girin:20
kisa kenari girin:10
Dikdortgenin alani=200
Dikdortgenin cevresi=60
```

```
Kullaniciдан deger almadan elde edilen sonuc
Dikdortgenin alani=50
Dikdortgenin cevresi=30
```

## UYGULAMA 3:

- öğrenci adında bir sınıf oluşturunuz.,
- Bu sınıfın private olarak float tipinde vize ve final olmak üzere 2 değişkeni olsun.
- Sınıfın **ogr1** adında bir nesnesi olsun ve bu nesne **sınıfın bitiş parantezinde** tanımlansın.
- Bu sınıfa arkadaş olan al() isminde bir fonksiyon tanımlansın ve kullanıcının klavyeden vize ve final notlarını girmesi istensin.
- Yine bu sınıfa arkadaş (friend) olan göster() isminde bir fonksiyon oluşturulsun ve öğrencinin ortalama notu hesaplanıp ekrana yazdırılsın.
- Örnek ekran çıktısı aşağıdaki gibi olabilir.

```
C:\Users\monster\Desktop\ders slaytlar2\algor  
Vize Notunu Giriniz:55  
Final Notunu Giriniz:75  
ortalaması=67  
-----  
Process exited after 2.69 seconds with
```

# UYGULAMA 3: ÇÖZÜM

```
1  #include <iostream>
2  using namespace std;
3
4  class ogrenci{
5      float vize,final;
6
7      public:
8          friend void al();
9          friend void goster();
10
11 }ogr1;
12
13 void al(){
14     float vz,fn;
15     cout<<"Vize Notunu Giriniz:";
16     cin>>vz;
17     cout<<"Final Notunu Giriniz:";
18     cin>>fn;
19     ogr1.vize=vz;
20     ogr1.final=fn;
21 }
22
23 void goster(){
24     float ort= ogr1.vize*0.4 + ogr1.final*0.6;
25     cout<<"ortalaması="<<ort<<endl;
26 }
27
28 int main() {
29
30     al();
31     goster();
32
33     return 0;
34 }
```

C:\Users\monster\Desktop\d

Vize Notunu Giriniz:55  
Final Notunu Giriniz:75  
ortalaması=67

-----

Process exited after 1.77  
Press any key to continue