

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

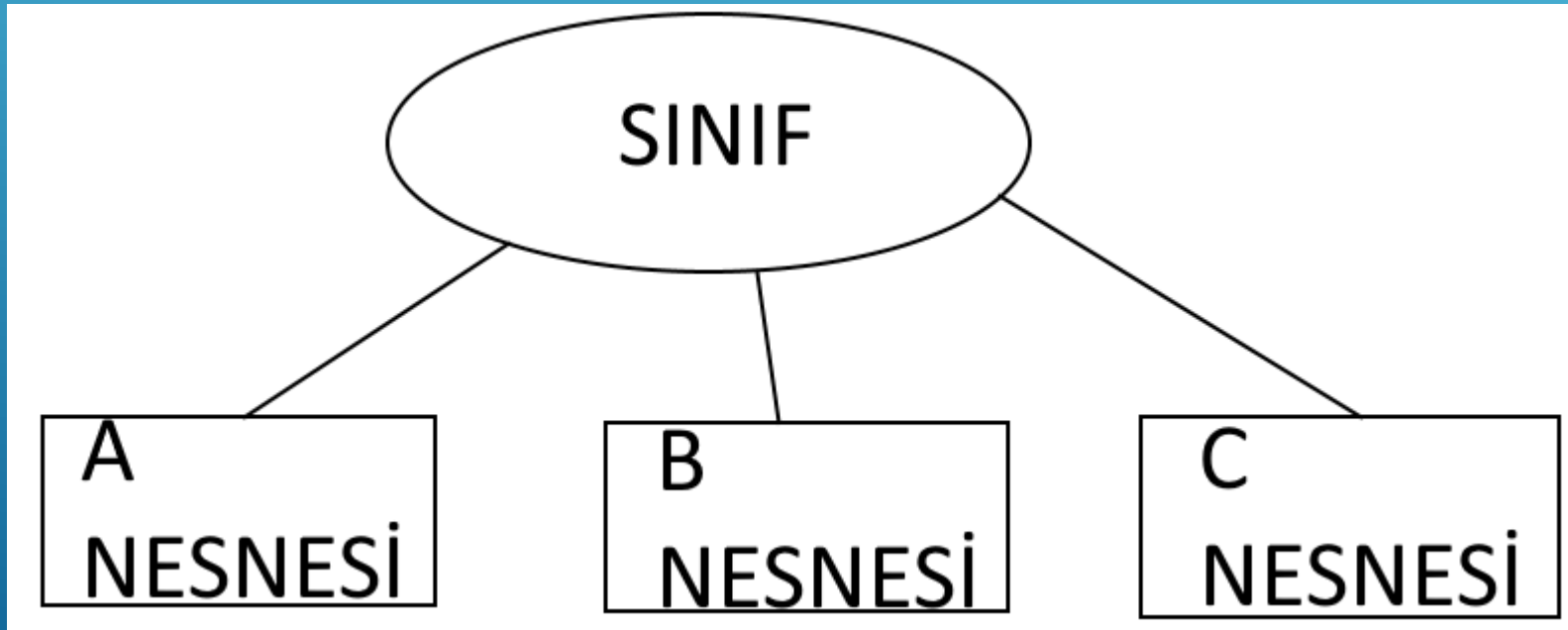
SINIFLAR VE NESNELER

NESNEYE YÖNELİK PROGRAMLAMA

- C++' da şuana kadar yapısal programlama tekniği kullanılarak anlatımlarımız gerçekleştirildi.
- Bu nedenle tüm örneklerimizi bu teknik ile çözdük.
- Çünkü şu ana kadarki çözdüğümüz örneklerin karmaşıklığı ve uzunluğu ileri seviyede değildi.
- Ancak 100-200 satırı geçen programlar yazacağımız zaman işler bir hayli karışacaktır.
- İşte burada imdadımıza nesneye yönelik programlama yetişiyor.
- Güvenli, dayanıklı, uyarlanabilir olmasının yanı sıra geliştirilebilir olması sayesinde programcılar nesneye yönelik programlama tekniği ile daha kaliteli ve karmaşıklığı daha az olan programlar yazabilmektedir.

SINIFLARA GİRİŞ

- Sınıflar için nesneye yönelik programlamanın kalbidir diyebiliriz.
- C++'da içerisinde nesneye yönelik programlamayıda barındırmaktadır.
- Bu dil içinde sınıf kavramı özel bir yer teşkil etmektedir.
- Bu kavramı iyi anlayarak ve kendimize özgü nesneler oluşturarak sınıfların avantajlarından yararlanabiliriz.



SINIFLARA GİRİŞ

Kod kısmına geçmeden önce güncel hayattan örnekler vererek konuyu daha anlaşılır hale getirelim.

- Örneğin sahip olduğumuz otomobili kullanırken sadece belli başlı parçalarla ilişkili oluruz.
- Mesela freni arabayı yavaşlatmak ve durdurmak için, gazı harekete geçirmek ve hızlandırmak için kullanırız.
- Peki biz motorun nasıl çalıştığını, gaza veya frene bastığımızda arabanın iç yapısında nasıl bir işleyiş olduğunu biliyormuyuz? Hayır.
- Bu örnekte gaz ve fren bizim programlamada kullandığımız veriler ve fonksiyonlar ile aynı anlamdadır.
- Arabalar tasarlanırken bize sadece bu kısımları kullanma izni veriliyor.
- Bizde bu fonksiyon ve verileri kullanarak (gaz, fren) arabanın iç yapısındaki erişemediğimiz fonksiyonlara ulaşabiliyoruz. Böylece o⁴ fonksiyonlara erişim sağlıyoruz.

SINIF BİLDİRİMİ

C++'da sınıf bildirimi yaparken class anahtar sözcüğü kullanılır.

1. Sınıf bildirimi class anahtar sözcüğü ile başlar ve hemen sonra sınıf adı gelir.
2. Private ve public kısımlarında önce veriler sonra fonksiyonlar yazılır.
3. Sınıfın içindeki verilere ve fonksiyonlara o sınıfın elemanları denilmektedir.
4. Sınıf bildirimi ; noktalı virgül ile bitirilmektedir.
5. Eğer private bildiriminin için veriler ve fonksiyonlar yazarsak bunlar özel tanımlama olur ve programın herhangi bir yerinde türetilen nesneler ve o sınıf içinde erişim hariç private verilere erişim sağlanamaz.
6. Eğer public bildiriminden sonra veriler ve fonksiyonlar yazarsak oradaki yapılar genele hitap eder ve programın herhangi bir yerinden erişilebilir.

NOT1: Sınıf sonundaki nesne listesini main() içindede tanımlayabiliriz.

NOT2: **private** veri ve fonksiyon tanımlamak istiyorsak **private** yazmamıza gerek yoktur.

Sınıfın adından sonra {} içinde direk veri ve fonksiyon yazarsak otomatik olarak **private** olarak kabul edilir.

Yani sınıf içinde **public** dışında yazdığımız herşeyi **private** kabul eder.

```
class sinifadi{  
    private: //Özel  
        veriler;  
        fonksiyonlar;  
    public: // genel  
        veriler;  
        fonksiyonlar;  
}nesneListesi;
```

ÖRNEK 1

```
1  #include <iostream>
2  using namespace std;
3
4  class dikdortgen{
5      public:
6          int kisa_kenar;
7          int uzun_kenar;
8
9          int alan(){
10             return kisa_kenar*uzun_kenar;
11         }
12
13         int cevre(){
14             return 2*(kisa_kenar+uzun_kenar);
15         }
16     }x;
17
18     int main() {
19         cout<<"kisa kenari giriniz:";
20         cin>>x.kisa_kenar;
21         cout<<"uzun kenari giriniz:";
22         cin>>x.uzun_kenar;
23
24         cout<<"Dikdortgenin alani="<<x.alan()<<endl;
25         cout<<"Dikdortgenin cevresi="<<x.cevre()<<endl;
26
27         return 0;
28     }
```

C:\Users\monster\Desktop\di

```
kisa kenari giriniz:5
uzun kenari giriniz:10
Dikdortgenin alani=50
Dikdortgenin cevresi=30

-----
Process exited after 4.62
Press any key to continue
```

ÖRNEK 1: nesne listesini main içindede tanımlayabilirdik.

```
1  #include <iostream>
2  using namespace std;
3
4  class dikdortgen{
5      public:
6          int kisa_kenar;
7          int uzun_kenar;
8
9          int alan(){
10             return kisa_kenar*uzun_kenar;
11         }
12
13         int cevre(){
14             return 2*(kisa_kenar+uzun_kenar);
15         }
16     };
17
18     int main() {
19
20         dikdortgen x; ←
21
22         cout<<"kisa kenari giriniz:";
23         cin>>x.kisa_kenar;
24         cout<<"uzun kenari giriniz:";
25         cin>>x.uzun_kenar;
26
27         cout<<"Dikdortgenin alani="<<x.alan()<<endl;
28         cout<<"Dikdortgenin cevresi="<<x.cevre()<<endl;
29
30         return 0;
31     }
```


C:\Users\monster\Desktop\der

```
kisa kenari giriniz:10
uzun kenari giriniz:15
Dikdortgenin alani=150
Dikdortgenin cevresi=50
```

```
-----
Process exited after 6.145
Press any key to continue
```


ÖRNEK 2: Peki veriler veya fonksiyonlar **private** olsaydı ne yapacaktık?

```
1  #include <iostream>
2  using namespace std;
3
4  class goster{
5
6      int sayi;
7
8      public:
9
10     void atama_yap(int a){
11         sayi=a;
12     }
13
14     int ekrana_yaz(){
15         return sayi;
16     }
17 } x, y ;
18
19 int main() {
20     x.atama_yap(50);
21     y.atama_yap(100);
22
23     cout<<"x = "<<x.ekrana_yaz()<<endl;
24     cout<<"y = "<<y.ekrana_yaz()<<endl;
25
26     return 0;
27 }
```

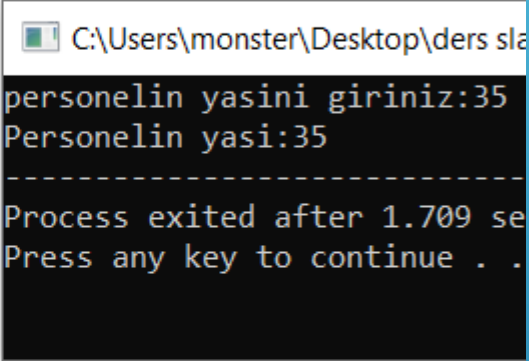


The screenshot shows a terminal window with the output of the program. The first line is 'x = 50' and the second line is 'y = 100'. Below these lines, there is a separator line of dashes, followed by the text 'Process exited' and 'Press any key to continue'.

- Bu örnekte **goster** adında bir sınıf oluşturduk.
- **sayi** adında bir tamsayı değişken oluşturduk ve bu değişken public dışında olduğu için otomatik **private** olarak algılandı.
- public: ' in içine **atama_yap** ve **ekrana_yaz** adında fonksiyonlar oluşturduk.
- **atama_yap** fonksiyonu ile sayi değişkenine ulaşmaya çalıştık. Çünkü private verilere başka bir şekilde ulaşamıyoruz. Yani sınıfın public olan fonksiyonuna eriştik o fonksiyon ise bizi private veriye ulaştırdı.
- Nesne olarak **x ve y** tanımladık.
- Main içinde **atama_yap** fonk. İle **x ve y** ye ulaşp 50 ve 100 değerlerini girdik.
- Bu değerleri **ekrana_yaz** fonksiyonu ile ekrana yazınca nesneye sabitlediğimiz 50 ve 100 sayılarını gördük.

ÖRNEK 3:

```
1  #include <iostream>
2  using namespace std;
3
4  class personel{
5
6      int yas;
7
8      public:
9
10         void gir(int y){
11             yas=y;
12         }
13
14         int yaz(){
15             cout<<yas;
16         }
17     };
18
19     int main() {
20         personel p;
21         int x;
22
23         cout<<"personelin yasini giriniz:";
24         cin>>x;
25         p.gir(x);
26
27         cout<<"Personelin yasi:";
28         p.yaz();
29
30         return 0;
31     }
```



- Bu örnekte yine bir private yas değişkenini tanımladık.
- Ancak main() içinde bir önceki örnekte olduğu gibi direkt sayıyı eklemedik.
- Sayıyı klavyeden kullanıcının girmesini istedik.
- Daha sonra p nesnemizi kullanarak fonksiyonlara ulaştık ve personelin yaşını ekrana yazdırdık.

ÖRNEK 4:

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  class personeller{
6
7      char personel_adi[20],personel_soyadi[20];
8      int yas;
9
10     public:
11         void arshiv(char *padi,char *psoyadi,int sayi){
12             strcpy(personel_adi,padi);
13             strcpy(personel_soyadi,psoyadi);
14             yas=sayi;
15         }
16
17         void goruntule(){
18             cout<<"Personel Adi:"<<personel_adi<<endl;
19             cout<<"Personel SoyAdi:"<<personel_soyadi<<endl;
20             cout<<"Personel Yasi:"<<yas<<endl<<endl;
21         }
22     }pers1,pers2,pers3;
23
24     int main() {
25
26         pers1.arshiv("Ali","AK",22);
27         pers2.arshiv("Veli", "DEMIR", 25);
28         pers3.arshiv("Tahir","VURAL",20);
29
30         pers1.goruntule();
31         pers2.goruntule();
32         pers3.goruntule();
33
34         return 0;
35     }
```

C:\Users\monster\Desktop\c

```
Personel Adi:Ali
Personel SoyAdi:AK
Personel Yasi:22

Personel Adi:Veli
Personel SoyAdi:DEMIR
Personel Yasi:25

Personel Adi:Tahir
Personel SoyAdi:VURAL
Personel Yasi:20

-----
Process exited after 0.01
Press any key to continue
```

ÖRNEK 4 (Açıklama)

- Bu örneğimizde işleri biraz karıştırdık.
- Personeller sınıfı oluşturduk ve içine private olarak personel_adi ve personel_soyadi olmak üzer 2 adet char dizisi, yas adında bir adet int değişken oluşturduk.
- Ardından public olarak arshiv ve goruntule adında 2 fonksiyon tanımladık.
- Arshiv de 2 adet char pointer seklinde argüman ve 1 adet int argüman tanımladık.
- Gelen sayiyi yas değişkenine, pointer olan gelen metinleri ise personel_adi ve personel_soyadi değişkenlerine atadık.
- Main() içinde nesnelerimizi kullanarak arşiv() fonksiyonu ile ad,soyad yas girdik, goruntule() ilede ekranda görüntüledik.

ÖRNEK 5: Bir önceki örnekteki benzer bir örnek yapalım. Fakat bu örnekte pointerlar ve dizileri kullanmayalım onun yerine string kullanalım.

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  class arabalar{
6
7      string marka, model;
8      int yil;
9
10     public:
11         void arabaoku(string mrk,string mdl,int yl){
12             marka=mrk;
13             model=mdl;
14             yil=yl;
15         }
16
17         void arabayaz(){
18             cout<<"arabanin markasi:"<<marka<<endl;
19             cout<<"arabanin modeli:"<<model<<endl;
20             cout<<"Arabanin Yili:"<<yil<<endl<<endl;
21         }
22     };
23
24     int main() {
25
26         arabalar a1,a2,a3;
27
28         a1.arabaoku("Opel","Astra", 2015);
29         a2.arabaoku("Fiat", "Egea", 2019);
30         a3.arabaoku("Citroen","C5", 2018);
31
32         a1.arabayaz();
33         a2.arabayaz();
34         a3.arabayaz();
35
36         return 0;
37     }
```

```
C:\Users\monster\Desktop\de
arabanin markasi:Opel
arabanin modeli:Astra
Arabanin Yili:2015

arabanin markasi:Fiat
arabanin modeli:Egea
Arabanin Yili:2019

arabanin markasi:Citroen
arabanin modeli:C5
Arabanin Yili:2018

-----
Process exited after 0.041
Press any key to continue
```

ÖRNEK 6:

- Kullanıcıdan 2 işçinin adını, soyadını, aylık kaç saat çalıştığını, saatlik ücretinin ne kadar olduğu bilgilerini alalım.
- Daha sonra bu bilgilere göre aylık maaşını hesaplayalım.
- Maaşı eğer 2800 TL den az ise Asgeri ücretten yüksek, fazla ise asgari ücretten yüksek yazsın.

Programı Oluştururken **isciler** adında sınıf oluşturalım. Sınıfta şu elemanlar olsun.

- Gerekli değişkenler **private** olsun.
- Klavyeden bilgi almak için -> **bilgi_gir** fonksiyonu
- Aylık maasi hesaplamak için -> **maas** fonksiyonu
- Girilen işçilerin bilgilerini ve hesaplanan maasini ekranda göstermek için -> **goster** fonksiyonu
- Ve askeri ücretten yüksek olup olmadığını mesajını yazan-> **yorum** fonksiyonu

ÖRNEK 6:

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  class isciler{
5      private:
6          string ad, soyad;
7          int saatlik_ucret, calistigi_saat;
8          int aylık_maas;
9      public:
10     void bilgi_gir(){
11         cout<<"iscinin adini giriniz:";
12         cin>>ad;
13         cout<<"iscinin soyadini giriniz:";
14         cin>>soyad;
15         cout<<"iscinin 1 saatlik ucretini giriniz:";
16         cin>>saatlik_ucret;
17         cout<<"iscinin 1 ayda kac saat calistigini giriniz:";
18         cin>>calistigi_saat;
19     }
20
21     int maas(){
22         return aylık_maas=saatlik_ucret*calistigi_saat;
23     }
24
25     void goster(){
26         cout<<"iscinin adi:"<<ad<<endl;
27         cout<<"iscinin soyadi:"<<soyad<<endl;
28         cout<<"iscinin maasi:"<<maas()<<endl;
29         yorum(maas());
30     }
31
32     void yorum(int aylık){
33         if (aylık>2800)
34             cout<<"Maas Asgari Ucretten Yuksek"<<endl;
35         else
36             cout<<"Maas Asgari Ucretten Dusuk"<<endl;
37     }
38 };
```

```
int main() {

    isciler isci1,isci2;

    isci1.bilgi_gir();
    cout<<"-----"<<endl;

    isci2.bilgi_gir();
    cout<<"-----"<<endl;

    cout<<"Iscilerin Bilgileri"<<endl<<endl;
    isci1.goster();
    cout<<"-----"<<endl;
    isci2.goster();

    return 0;
}
```

```
iscinin adini giriniz:Asiye
iscinin soyadini giriniz:Duman
iscinin 1 saatlik ucretini giriniz:60
iscinin 1 ayda kac saat calistigini giriniz:75
-----
iscinin adini giriniz:Mehmet
iscinin soyadini giriniz:BAKIR
iscinin 1 saatlik ucretini giriniz:45
iscinin 1 ayda kac saat calistigini giriniz:50
-----
Iscilerin Bilgileri

iscinin adi:Asiye
iscinin soyadi:Duman
iscinin maasi:4500
Maas Asgari Ucretten Yuksek
-----
iscinin adi:Mehmet
iscinin soyadi:BAKIR
iscinin maasi:2250
Maas Asgari Ucretten Dusuk
```


ÖRNEK 6 (Açıklaması)

1. Öncelikle private olarak ad, soyad, saatlik_ucret, calistigi_saat, aylık_maas değişkenlerimizi tanımladık.
2. Public kısmında void türünde bilgi_gir() adında bir fonksiyon tanımadık ve private olarak tanımladığımız değişkenler klavyeden bilgi okuduk.
3. İnt türünde maas() fonksiyonu tanımladık, saat ücreti ve aylık çalışma saatini çarparak aylık maaş hesaplanıp döndürüldü.
4. Goster fonksiyonunda klavyeden aldığımız ad,soyad bilgilerini, maas() fonksiyonunu çağırıp fonksiyondan gelen maaş bilgisini ve yorum(maas()) fonksiyonu ile maas() fonksiyonundan gelen bilgiyi yorum fonksiyonuna gönderip yorumu ekrana yazdırdık.
5. Main içinde isci1,isci2 adında 2 nesne tanımladık, her nesne için bilgileri aldık ve ekranda gösterdik.