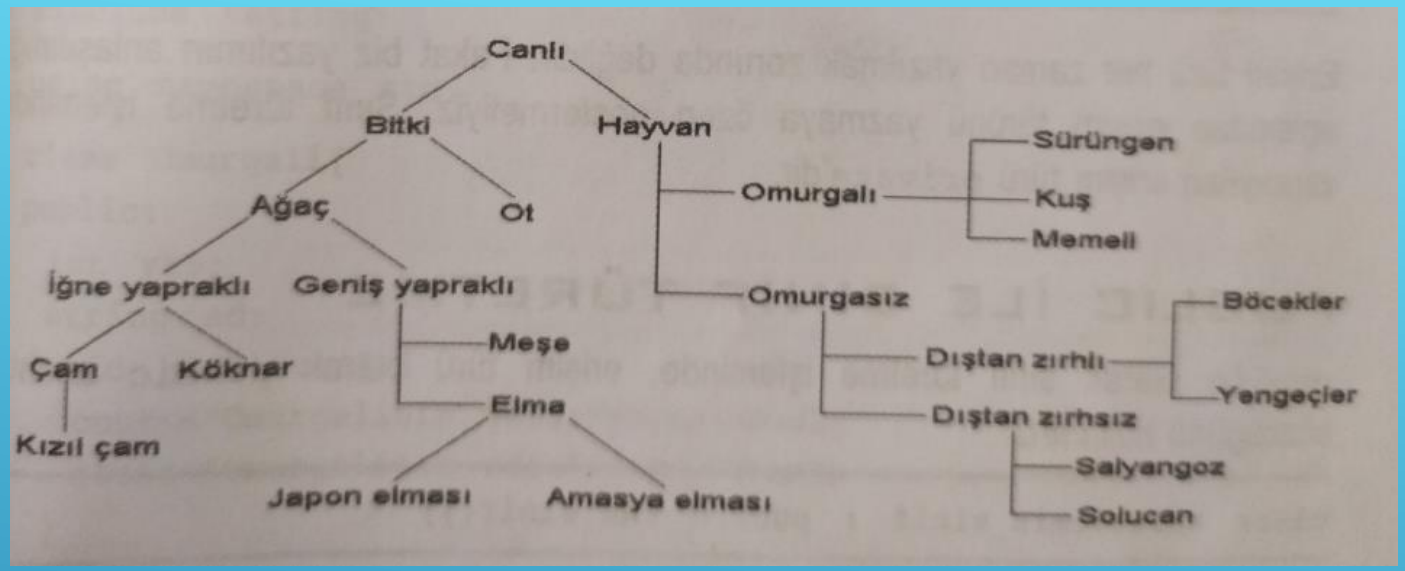
MALATYA
TURGUT ÖZAL ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
ALGORİTMA VE PROGRAMLAMA 2

KALITIM

KALITIM

- Kalıtım, var olan bir ana sınıfın özelliklerinin
  - yeri geldiğinde değiştirilerek
  - yeri geldiğinde ise yeni özellikler eklenilerek bir alt sınıfa aktarılmasıdır.
- Yukarıdan aşağıya doğru hiyerarşik bir yapısı bulunmaktadır.
- Bu nedenle en üst kademedeki ana sınıfa **temel sınıf**,
- temel sınıftan alt sınıftaki sınıflara ise **türetilmiş sınıf** denir.

KALITIM



- En üstteki **Canlı** sınıfı, Şemadaki bütün sınıfların **temel sınıf**ıdır.
- **Bitki** ve **Hayvan** Sınıfları **Canlı** sınıfı için **türetilmiş sınıf**lardır.
- **Bitki** sınıfı; **ağaç** ve **ot** sınıfı için **üst sınıf**tır. Aynı şekilde **omurgalı** ve **omurgasız** sınıfları için de **Hayvan üst sınıf**tır.
- **Canlı** sınıfında bulunan bütün özellikler, **canlı** sınıfından türetilmiş sınıflar olan **bitki** ve **hayvan** sınıfınında özellikleridir. Ancak **bitki** ve **hayvan** sınıfında olan bir özellik, **canlı** sınıfında olmayabilir. Kısacası türetilen sınıflar, temel sınıfların tüm özelliklerini taşırken kendilerine özel özellikleride içerisinde barındırabilirler.
- **Omurgalı** sınıfındaki **Kuş** sınıfı, **Omurgalı** sınıfına göre daha az nesneyi ifade etmektedir. Çünkü **kuş** sınıfında sadece o sınıfla ilgili nesneler oluşturulurken, **omurgalı** sınıfından hem **sürüngen** hem **kuş** hemde **memeli** sınıflarından nesneler oluşturulur.

TEMEL SINIF VE TÜRETİLMİŞ SINIF TANIMLAMA

- Ana sınıftan bir türetilmiş sınıf tanımlarken öncelikle türetilmiş sınıfın adı yazılır ve devamında: erişim türü ve ana sınıfın adı yazılır. Erişim türü public ve private olabilir.

- class türetilen_sinif_adi : erişim_türü ana_sınıf{};

- Erişim türü yazmak zorunlu değildir. Eğer yazılmaz ise varsayılan olarak **private** kabul edilir.

PUBLİC İLE SINIF TÜRETME

- Erişim türüolarak public anahtar sözcüğü kullanılır.

  class türetilen_sinif_adi : **public** ana_sınıf{};
- Bir sınıf başka bir sınıftan **public** olarak türetilmişse; türetildiği ana sınıfın **public** olan bütün elemanları, türetilmiş sınıfında **public** elemanlarıdır.
- Türetilmiş sınıfın bütün nesneleri ana sınıfın **public** elemanlarına erişebilir.
- Burada en önemli nokta türetilmiş sınıf nesnelerinin ana sınıfın **private** elemanlarına erişimlerinin mümkün olmamasıdır.

```
class Canli{
    private:
        veriler;
        fonksiyonlar;
    public:
        veriler;
        fonksiyonlar;
};

class Bitki:public Canli{};
class Hayvan:public Canli{};
```

ÖRNEK -1

```cpp
#include <iostream>
#include <string>
using namespace std;

class Omurgali{
    public:
        int yas;
        string ad;

        void goruntule(){
            cout<<"Omurgalinin yasi:"<<yas<<endl;
            cout<<"Omurgalinin adi:"<<ad<<endl<<endl;
        }
};

class Surungen:public Omurgali{
    public:
        Surungen(int y, string a){
            yas=y;
            ad=a;
        }
};

class Kus:public Omurgali{
    public:
        Kus(int y, string a){
            yas=y;
            ad=a;
        }
};
```

```cpp
int main() {
    Surungen nesne1(3,"yilan");
    Kus nesne2(5,"kanarya");

    nesne1.goruntule();
    nesne2.goruntule();


    return 0;
}
```

```
C:\Users\monster\Desktop\ders slaytla
Omurgalinin yasi:3
Omurgalinin adi:yilan

Omurgalinin yasi:5
Omurgalinin adi:kanarya
```

Örnek 1- Açıklama

- Omurgali sınıfı oluşturduk ve her omurgalinin adı ve yaşı olabileceği için ad ve yas değişkenlerini public tanımladık, ve görüntüle() fonksiyonu ile ekrana yazdırmak istedik.
- Daha sonra class Kus: public Omurgali, class Surunge: public Omurgali; tanımlamaları ile Omurgali sınıfından kus ve sürüngen adında iki sınıf türettik. Sınıflar içerisinde yapıcı fonksiyonlar ile omurgali sınıfının değişkenlerine atama yaptık.
- Main için Kus ve Surungen sınıflarından nesneler tanımladık ve girdiğimiz degerleride o nesnenin gorutule fonksiyonu ile ekrana getirdik.

Bu örnekte, dikkat edilmesi gereken en önemli kısım, sınıftaki elemanların tanımlama şeklidir. Kus ve sürüngen sınıfları, public Omurgali şeklinde türedi. Böyle bir durumda kus ve sürüngen sınıfı içerisinde, Omurgalı temel sınıfı içerisindeki bütün public veriler türetilmiş sınıflar içinde public olmuş oldu.
Yani ana sınıftan türetilmiş sınıf public ile türetildi ise; ana sınıftaki bütün public tanımlamalar, türetilen sınıf içerisinden erişilebilir hale geldi.

PRIVATE İLE SINIF TÜRETME

- Ana sınıftan **public** olarak bir sınıf türetebildiğimiz gibi **private** olarakta bir sınıf türetebiliriz.
- **Public** türetmede ana sınıfın **public** üyeleri türetilmiş sınıfında **public** üyeleri oluyordu.
- **Private** sınıf türetmede ise ana sınıfın **public** üyeleri türetilmiş sınıfın **private** üyeleri haline gelir.
- Burada dikkat etmemiz gereken nokta; Sınıf türetme işlem **public** te olsa **privatede** olsa ana sınıfın **private** üyelerine türetilmiş sınıf nesneleri erişemez.

ÖRNEK -2

```cpp
#include <iostream>
#include <string>
using namespace std;

class personeller{
    public:
        string ad,soyad;

        personeller(){
        }
};

class sekreter:private personeller{
    public:
        sekreter(string a, string b){
            ad=a;
            soyad=b;
        }

        void goruntule(){
            cout<<"ad:"<<ad<<endl;
            cout<<"soyad:"<<soyad<<endl<<endl;
        }
};

int main() {
    sekreter kisi("Aylin","Demir");
    kisi.goruntule();  //doğru kullanım

    // cout<<kisi.ad;  //hatalı kullanım olurdu

    return 0;
}
```

```
Seç C:\Users\monster\Desktop\
ad:Aylin
soyad:Demir
```

- Örneğimizde ana sınıfımız olan personeller sınıfından private olarak bir sekreter sınıfı türettik.
- Personeller sınıfının public elemanları olan ad ve soyad değişkenleri, sekreter sınıfının private elemanları haline geldi.
- Main içinde kisi adında bir nesne oluşturduk ve yapıcı fonksiyona parametre olarak 2 string verdik.
- Yapıcı fonksiyon bu parametreleri personeller sınıfından private olarak aldığı ad ve soyad değişkenlerine atadı.
- Kisi.goruntule() fonksiyonunu çağırarak ad ve soyad elemanlarını ekrana yazdırdık.

- Eğer ekrana yazdırmak için cout<<kisi.ad; gibi bir ifade kullansaydık hatalı olacaktı.
- Çünkü kisi nesnesinin sahip olduğu sınıfın private olan ad ve soyad değişkenlerine doğrudan erişemiyorduk.

ÖRNEK -3

```cpp
#include <iostream>
#include <string>
using namespace std;

class dortgen{
    public:
        int kenar1,kenar2;

        Dortgen(){
        }

        void alan(){
            cout<<"Dortgenin alani="<<kenar1*kenar2<<endl;
        }
};

class kare:private dortgen{
    public:
        kare(int x){
            kenar1=x;
            kenar2=x;
        }

        void alancagir(){
            alan();
        }
};

int main() {
    kare nesne(4);
    nesne.alancagir(); //doğru kullanım

    //nesne.alan(); Hatalı olacaktı!!!

    return 0;
}
```

```
C:\Users\monster\Desktop\ders
Dortgenin alani=16

--------------------------------
Process exited after 0.0446
```

- Bu örneğimiz kalıtım işlemini daha iyi anlatan bir program. Çünkü kare bir dörtgendir yani dörtgenin bir alt sınıfıdır.
- Dortgen sınıfını tanımladık ve kare sınıfını dörtgen sınıfından private olarak türettik.
- Private olarak türettiğimiz için, dörtgen sınıfının public olan kenar1 ve kenar2 elemanları kare sınıfın private elemanları oldu.
- Main içinde tanımladığımız nesneye parametre olarak verdiğimiz sayı, kalıtım işleminden kare sınıfına private eleman olarak gelen kenar1 ve kenar2 elemanlarına atandı.
- nesne.alancagir() fonksiyonu ile private bir eleman olan alan() fonksiyonuna eriştik ve ekran çıktısı elde ettik.

- nesne.alan() şeklinde kullanım hata verecekti.Çünkü alan() fonksiyonu dörtgen sınıfında public ama kare sınıfında private olarak olarak aktarıldı. Bu nedenle kare sınıfındaki oluşturulan alancagir() isimli public bir fonksiyon ile alan() fonksiyonuna erişebildik.

PROTECTED ELEMANLAR

- Önceki örneklerde türetilmiş sınıfın ana sınıftaki private bir üyeye erişiminin mümkün olmadığını söylemiştik.
- Fakat programlarımızda türetilmiş sınıfın ana sınıfın private elemanlarınada erişmek istediğimiz zamanlar olacak.
- İşte bu noktada protected elemanlar devreye giriyor.
- Ana sınıfta protected olarak tanımlanan bir eleman, public bir türetmede türetilmiş sınıfında protected elemanıdır. Private türetmede ise ana sınıfın protected üyeleri türetilmiş sınıfın private üyeleridir.
- Her iki durumda da ana sınıfın protected üyesine türetilmiş sınıfın eleman fonksiyonları ile erişilebilir.

Erişim	Sınıf İçerisinden Erişim	Türetilmiş Sınıftan Erişim	Dışarıdan Erişim
public	*evet*	*evet*	*evet*
protected	*evet*	*evet*	*hayır*
private	*evet*	*hayır*	*hayır*

ÖRNEK -4

```cpp
#include <iostream>
#include <string>
using namespace std;

class okul{
    protected:
        string ad;
    public:
        okul(){
        } //varsayılan yapıcı

        okul(string x){
            ad=x;
        }

        string ad_dondur(){
            return ad;
        }
};
```

```cpp
class ogrenci:public okul{
    string soyad;
    int numara;
    public:
        ogrenci(){}

        ogrenci(string a,string s,int no){
        ad=a;
        soyad=s;
        numara=no;
        }

        void goster(){
            cout<<"Ad="<<ad<<endl;
            cout<<"Soyad="<<soyad<<endl;
            cout<<"Numara="<<numara<<endl;
            cout<<"------------"<<endl;
        }
};
```

```cpp
int main() {
    okul okl("MTU");
    ogrenci ogr1("Ahmet","Demir",1001);
    ogrenci ogr2("Filiz","Vural",1002);
    ogr1.goster();
    ogr2.goster();
    cout<<okl.ad_dondur()<<endl;


    return 0;
}
```

```
Ad=Ahmet
Soyad=Demir
Numara=1001
------------
Ad=Filiz
Soyad=Vural
Numara=1002
------------
MTU
```

- Anasınıfımız olan okul sınıfında protected olan ad isminde bir string tanımladık.
- Okul sınıfında public olarak öğrenci adında sınıf türettik.
- Ogrenci sınıfı public olarak türetildiği için ana sınıfı olan okul sınıfının public elemanları öğrenci sınıfınında public elemanları oldu.
- Ana sınıfının protected elemanları ise öğrenci sınıfının protected elemanı oldu.
- Daha sonra öğrenci sınıfında private olan soyad ve numara değişkenlerini tanımladık.
- Main içinde ise uygun yapıcı fonksiyonlar ile birlikte nesneleri oluşturduk ve elemanlarıda değer atadık.
- Goster ve ad_dondur fonksiyonları ilede değerleri ekranda gösterdik.

PROTECTED İLE SINIF TÜRETME

- Önceki örneklerde public veya private olarak sınıf türetmiştik. Ana sınıftan protected olarakte sınıflar türetilebilir.
- Public sınıf türetmede ana sınıfın public elemanları, türetilmiş sınıfında public elemanları haline, ana sınıfın protected elemanları türetilmiş sınıfında protected elemanları haline geldiğinden bahsetmiştik.
- Aynı şekilde private sınıf türetmede, ana sınıfın public veya protected elemanlarının türetilmiş sınıfın private elemanları haline geldiğini söylemiştik.

- Protected sınıf türetmede ise; ana sınıfın public ve protected elemanları türetilmiş sınıfın protected elemanları haline gelir.

ÖRNEK 5

```cpp
#include <iostream>
#include <string>
using namespace std;

class tasit{
    protected:
        string marka_adi,model_adi;
    public:
        tasit(){} //varsayılan yapıcı

        void isim(string a, string s){
            marka_adi=a;
            model_adi=s;
        }
};

class otomobil:protected tasit{
    int hiz;

    public:
        otomobil(){}

        void otomobil_hiz(int m){
            hiz=m;
        }

        void erisim(string a,string b){
            isim(a,b);
        }

        void goster(){
            cout<<"Marka Adi="<<marka_adi<<endl;
            cout<<"Model Adi="<<model_adi<<endl;
            cout<<"Maksimum hizi="<<hiz<<endl;
            cout<<"-------------"<<endl;
        }
};
```

```cpp
int main() {
    otomobil oto[3];
    oto[0].erisim("Audi","A3");
    oto[1].erisim("Toyota","Corolla");
    oto[2].erisim("Opel","Astra");

    oto[0].otomobil_hiz(250);
    oto[1].otomobil_hiz(230);
    oto[2].otomobil_hiz(210);

    int i;
    for(i=0;i<3;i++){
        oto[i].goster();
    }

    return 0;
}
```

```
Marka Adi=Audi
Model Adi=A3
Maksimum hizi=250
-------------
Marka Adi=Toyota
Model Adi=Corolla
Maksimum hizi=230
-------------
Marka Adi=Opel
Model Adi=Astra
Maksimum hizi=210
-------------
```