

# Algoritmalar ve Programlama II – BLM 102

## Hafta 7: Kalıtım (Inheritance)



Fenerbahçe Üniversitesi

# Öğretim Elemanları

Öğretim Üyesi: Dr. Vecdi Emre Levent

Ofis: 311

Email: [emre.levent@fbu.edu.tr](mailto:emre.levent@fbu.edu.tr)

Asistan: Arş. Gör. Uğur Özbalkan

Ofis: 311

Email: [ugur.ozbalkan@fbu.edu.tr](mailto:ugur.ozbalkan@fbu.edu.tr)

# Ders Planı

- Kalıtım (Inheritance)
- Ana Sınıf ve Türetilmiş Sınıf İlişkisi
- Sınıflarda Protected Tanımlaması

# Kalıtım (Inheritance)

- Kalıtım, daha önceden yaratılmış bir sınıfın tüm değişken ve fonksiyonlarını başka bir sınıfta tekrar kullanımını sağlayan bir mekanizmadır.
- Mevcut sınıfa'a base class, yeni sınıfa ise, derived (türetilmiş) class denir.
- Türetilmiş sınıflar, türetildikleri sınıflara göre daha özelleşmiş objeleri temsil ederler.
- Örneğin araçlar sınıfı ve otomobiller sınıfı

# Kalıtım (Inheritance)

- Bir sınıf, başka bir sınıftan doğrudan türetilebileceği gibi, dolaylı da türetilmiş olabilir.

# Kalıtım (Inheritance)

## Örnek 1:

- Araç sınıfından türetilmiş bir otomobil sınıfı olsun, otomobil sınıfından da Volkswagen sınıfı olsun. Volkswagen sınıfı araç sınıfından dolaylı üretilmiş olur.

# Kalıtım (Inheritance)

## Örnek 2:

- Araç, kamyon ve otobüs isminde 3 farklı sınıf yaratılacaktır. Bu sınıfların fonksiyonlarından
  - yakıtMiktariniAl()
  - yolcuKapasitesiniAl()
  - frenYap()

işlevleri aynı olabilir.

# Kalıtım (Inheritance)

## Örnek 2:

- Kalıtım metodolojisini izlemeden, bu sınıfları ayrı ayrı yazıldığında aşağıdaki gibi tüm fonksiyonların kopyaları tüm sınıflarda olacaktır. Bir fonksiyonda değişiklik yapılma ihtiyacı olduğunda, diğerlerinde de değiştirmek gerekecektir.

Araba

```
yakitMiktariniAl()  
yolcuKapasitesiniAl()  
frenYap()
```

Kamyon

```
yakitMiktariniAl()  
yolcuKapasitesiniAl()  
frenYap()
```

Otobus

```
yakitMiktariniAl()  
yolcuKapasitesiniAl()  
frenYap()
```



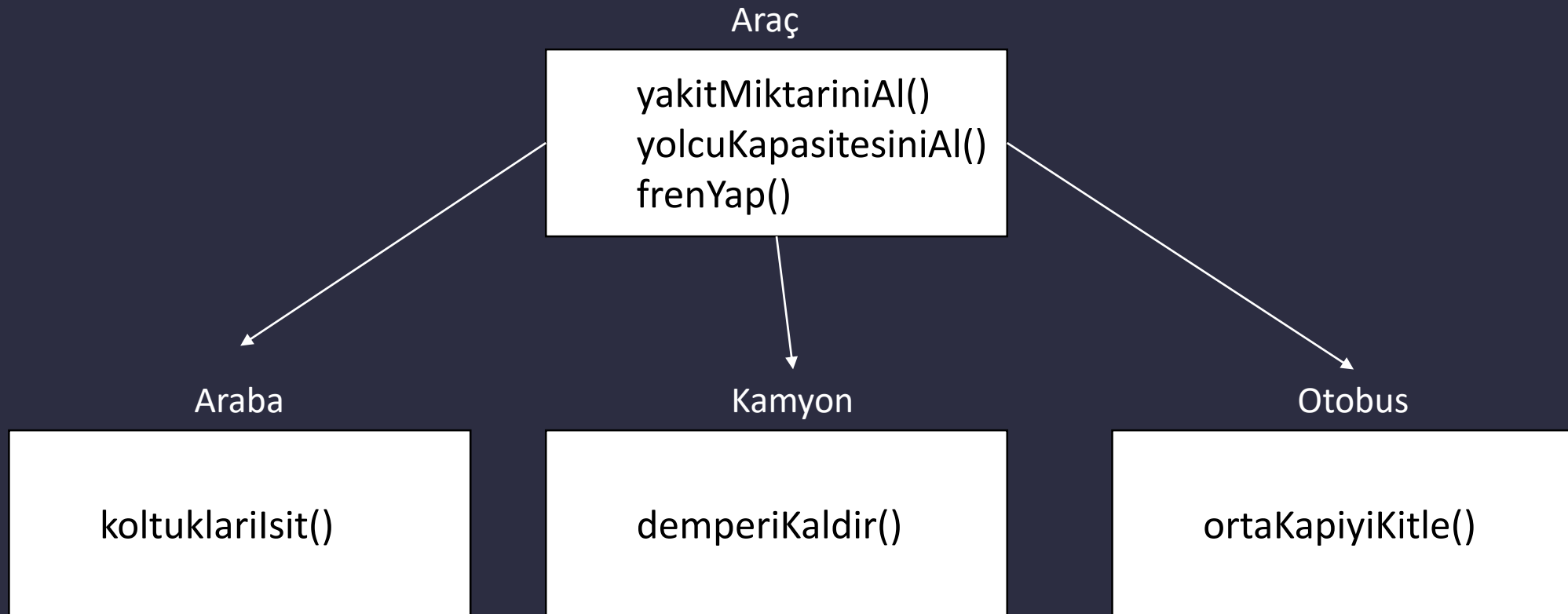
## Kalıtım (Inheritance)

### Örnek 2:

- Bu durumdan kaçınılmak için kalıtım (inheritance) kullanılmalıdır.
- Tüm sınıfları kapsayacak araç isminde bir sınıf yapıp, bu ortak kullanılacak fonksiyonları bu sınıfa yazıp, diğer sınıfların (araba, otobüs ve kamyon) bu sınıftan kalıtım yapmalarını sağlatılmalıdır.

# Kalıtım (Inheritance)

## Örnek 2:



# Kalıtım (Inheritance)

C++ dilindeki syntax'i

```
class sinifAdi : erisimModu baseSinifAdi
{
    //Sinif İçeriği
};
```

# Kalıtım (Inheritance)

## C++ dilindeki syntax'ı

```
class sinifAdi : erisimModu baseSinifAdi  
{  
    //Sinif içeriği  
};
```

- Erişim Modları: 3 tür bulunmaktadır.
  - Public: Base sınıfında tanımlanmış olan public ve protected değişken ve fonksiyonlara erişebilir. Base'de private tanımlanmış olanlara erişemez.
  - Protected: Base sınıfında bu ifade ile tanımlanmış değişken ve fonksiyonlar, sadece kalıtım yapıldıkları türetilmiş sınıflar tarafından erişilebilirler
  - Private: Türetilmiş sınıfta, base sınıftaki tüm değişken ve fonksiyonlar private tanımlanmış olur.

Not: Base sınıfındaki private tanımlanmış olan değişken ve fonksiyonların tamamı, hiçbir modda erişilebilir değildir.

# Kalıtım (Inheritance)

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

```
class Aile
{
public:
string soyadi;
};
```

```
class Birey : public Aile
{
public:
string adi;
};
```

```
int main()
{
Birey obj1;

obj1.soyadi = "levent";
obj1.adi = "emre";
cout << "Birey ismi " << obj1.adi <<
obj1.soyadi << endl;
```

```
return 0;
}
```

Çıktı:  
Birey ismi emrelevent

# Kalıtım (Inheritance)

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

class A
{
public:
int x;
protected:
int y;
private:
int z;
};

class B : public A
{
// x public
// y protected
// z, B sınıfından erişilebilir değildir
};
```

```
class C : protected A
{
// x is protected
// y is protected
// z, C sınıfından erişilebilir değildir
};

class D : private A
{
// x is private
// y is private
// z, D sınıfından erişilebilir değildir
};

int main()
{
A obj1;
B obj2;
C obj3;
D obj4;
return 0;
}
```

# Kalıtım (Inheritance)

## C++ dilindeki syntax'ı

- A sınıfında tanımlanmış olan 3 değişken bulunmaktadır.
- Main'de A sınıfından türetmiş olan obj1'in sadece x değişkenine erişim sağlanabilir.

# Kalıtım (Inheritance)

## C++ dilindeki syntax'ı

- B sınıfı kendi içerisinde bir değişken tanımlaması barındırmıyor. Ancak A sınıfından public kalıtım yapılmış.
- Yani;
- B sınıfının içerisinde, A sınıfının, x ve y değişkenlerine erişilebilir. Ve bunlar B sınıfının içerisinde x public, y ise protected tanımlanmış olur.
- Main'de B sınıfından türetmiş olan obj2'in sadece x değişkenine erişim sağlanabilir.



# Kalıtım (Inheritance)

## C++ dilindeki syntax'ı

- C sınıfı kendi içerisinde bir değişken tanımlaması barındırmıyor. Ancak A sınıfından protected kalıtım yapılmış.
- Yani;
- C sınıfının içerisinde, A sınıfının, x ve y değişkenlerine erişilebilir. Ve bunlar C sınıfının içerisinde x protected, y ise protected tanımlanmış olur.
- Main'de C sınıfından türetmiş olan obj3'ün hiçbir elemanına ulaşamaz.

# Kalıtım (Inheritance)

## C++ dilindeki syntax'ı

- D sınıfı kendi içerisinde bir değişken tanımlaması barındırmıyor. Ancak A sınıfından private kalıtım yapılmış.
- Yani;
- D sınıfının içerisinde, A sınıfının, x ve y değişkenlerine erişilebilir. Ve bunlar D sınıfının içerisinde x private, y ise private tanımlanmış olur.
- Main'de D sınıfından türetilmiş olan obj4'ün hiçbir elemanına ulaşamaz.

# Kalıtım (Inheritance)

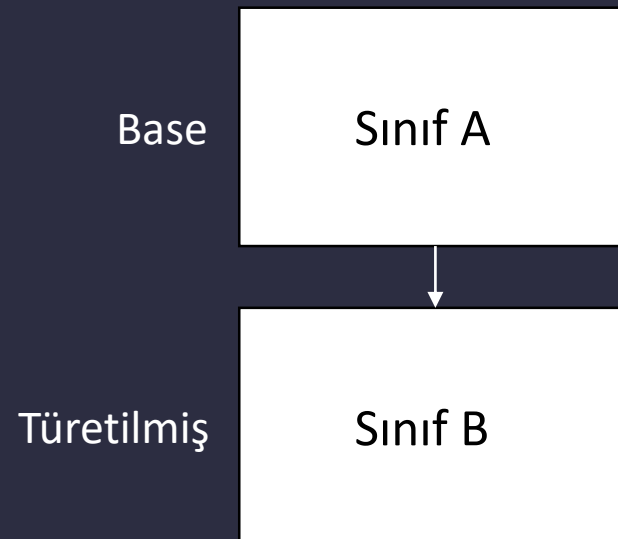
<b>Base Sınıf Üyeleri / Türetilmiş Sınıf Üyeleri</b>	<b>Public</b>	<b>Protected</b>	<b>Private</b>
<b>Public</b>	Public	Protected	Private
<b>Protected</b>	Protected	Protected	Private
<b>Private</b>	Erişilemez	Erişilemez	Erişilemez

Kalıtım Modu Tablosu

# Kalıtım (Inheritance)

## Kalıtım Türleri

Tek kalıtım: Bir sınıfın sadece bir başka sınıf tarafından kalıtım yapılmasıdır.



# Kalıtım (Inheritance)

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

class Arac {
public:
    Arac()
    {
        cout << "Bu bir aractir" << endl;
    }
};

class Araba : public Arac {

};

int main()
{
    Araba obj;
    return 0;
}
```

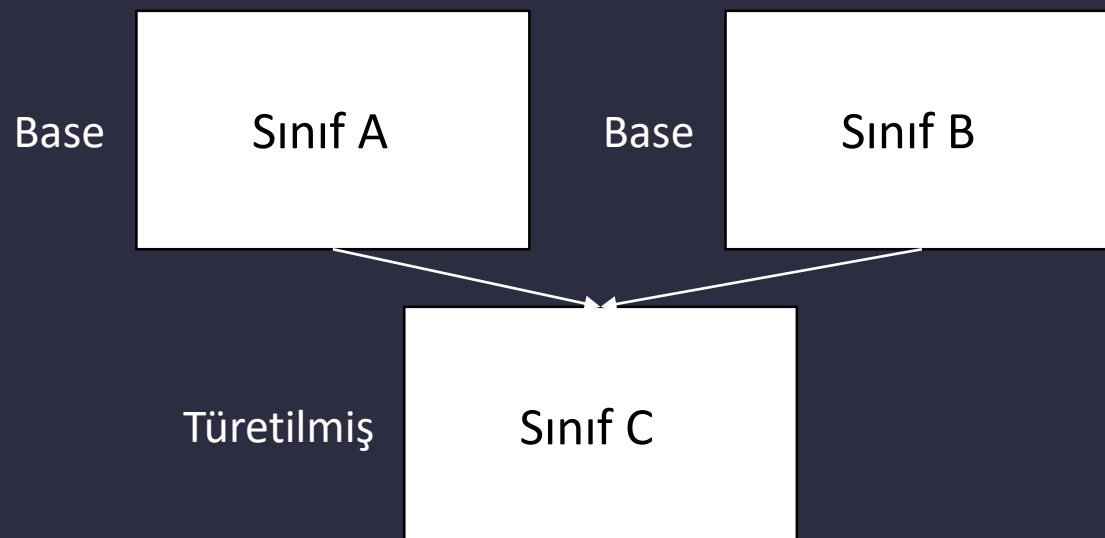
Çıktı:

Bu bir aractir.

# Kalıtım (Inheritance)

## Kalıtım Türleri

Çoklu kalıtım: Türetilen sınıfın birden çok base sınıfının olması durumudur.



# Kalıtım (Inheritance)

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

class Arac {
public:
Arac()
{
cout << "Bu bir aractir" << endl;
}

};

class dortTekerli {
public:
dortTekerli()
{
cout << "Bu bir 4 tekerlidir" << endl;
}
};
```

```
class Araba : public Arac, public dortTekerli {

};

int main()
{
Araba obj;
return 0;
}
```

Çıktı:

Bu bir aractir.

Bu bir 4 tekerlidir