




BİL102 Nesne Yönelimli Programlama

Dr. Öğr. Üyesi Yavuz CANBAY
Kahramanmaraş Sütçü İmam Üniversitesi
Bilgisayar Mühendisliği Bölümü



›Konu 4. Kalıtım-devam




```
// C++ program to explain
// Single inheritance
#include <iostream>
using namespace std;

// base class
class Vehicle {
public:
    Vehicle()
    {
        cout << "This is a Vehicle" << endl;
    }
};

// sub class derived from two base classes
class Car: public Vehicle{

};

// main function
int main()
{
    // creating object of sub class will
    // invoke the constructor of base classes
    Car obj;
    return 0;
}
```



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


// first base class
class Vehicle {
public:
    Vehicle()
    {
        cout << "This is a Vehicle" << endl;
    }
};

// second base class
class FourWheeler {
public:
    FourWheeler()
    {
        cout << "This is a 4 wheeler Vehicle" << endl;
    }
};

// sub class derived from two base classes
class Car: public Vehicle, public FourWheeler {

};

// main function
int main()
{
    // creating object of sub class will
    // invoke the constructor of base classes
    Car obj;
    return 0;
}
```



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

// base class
class Vehicle
{
public:
    Vehicle()
    {
        cout << "This is a Vehicle" << endl;
    }
};

// first sub class
class Car: public Vehicle
{
};

// second sub class
class Bus: public Vehicle
{
};

// main function
int main()
{
    // creating object of sub class will
    // invoke the constructor of base class
    Car obj1;
    Bus obj2;
    return 0;
}
```

```
using namespace std;

// Base class
class Shape {
public:
    void setWidth(int w) {
        width = w;
    }
    void setHeight(int h) {
        height = h;
    }

protected:
    int width;
    int height;
};


// Derived class
class Rectangle: public Shape {
public:
    int getArea() {
        return (width * height);
    }
};

int main(void) {
    Rectangle Rect;


    Rect.setWidth(5);
    Rect.setHeight(7);


    // Print the area of the object.
    cout << "Total area: " << Rect.getArea() << endl;

    return 0;
}
```



```
#include <iostream>
using namespace std;
class Animal {
    public:
    void eat() {
        cout<<"Eating..."<<endl;
    }
};
class Dog: public Animal
{
    public:
    void bark(){
        cout<<"Barking...";
    }
};
int main(void) {
    Dog d1;
    d1.eat();
    d1.bark();
    return 0;
}
```





```
#include <iostream>
using namespace std;
class Animal {
public:
void eat() {
    cout<<"Eating..."<<endl;
}
};
class Dog: public Animal
{
public:
void bark(){
    cout<<"Barking..."<<endl;
}
};
class BabyDog: public Dog
{
public:
void weep() {
    cout<<"Weeping...";
}
};
int main(void) {
    BabyDog d1;
    d1.eat();
    d1.bark();
    d1.weep();
    return 0;
}
```



```
#include <iostream>
using namespace std;
class Teacher {
public:
    Teacher(){
        cout<<"Hey Guys, I am a teacher"<<endl;
    }
    string collegeName = "Beginnersbook";
};
//This class inherits Teacher class
class MathTeacher: public Teacher {
public:
    MathTeacher(){
        cout<<"I am a Math Teacher"<<endl;
    }
    string mainSub = "Math";
    string name = "Negan";
};
int main() {
    MathTeacher obj;
    cout<<"Name: "<<obj.name<<endl;
    cout<<"College Name: "<<obj.collegeName<<endl;
    cout<<"Main Subject: "<<obj.mainSub<<endl;
    return 0;
}
```

```

2  #include<iostream>
3  #include<string>
4  using namespace std;
5  class temel {
6  public:
7      string ad,soyad;
8      int yas;
9      void sorgu()
10     {
11         cout << "Adiniz = \n";
12         cin >> ad;
13         cout << "Soyadiniz = " << endl;
14         cin >> soyad;
15         cout << "Yasiniz = " << endl;
16         cin >> yas;
17     }
18 };
19 class turetilmis :public temel
20 {
21 };
22 int main()
23 {
24     turetilmis t;
25     t.sorgu();
26     system("Pause");
27     return 0;
28 }

```

```

:Adiniz = sema
:Soyadiniz = kılıç
Yasiniz = 78
Press any key to continue . . .

```

```

1 #include "stdafx.h"
2 #include<iostream>
3 #include<string>
4 using namespace std;
5 class canlilar {
6 public:
7     canlilar();
8     string tur;
9     string cins;
10 };
11 canlilar::canlilar()
12 {
13     cout << "\t\t-----Canlilar Alemine Hosgeldiniz-----" << endl;
14 }
15 }
16 class hayvanlar :public canlilar
17 {
18 public:
19     hayvanlar();
20     int ayaksayisi, elsayisi;
21 };
22 hayvanlar::hayvanlar()
23 {
24     cout << "Cinsini Giriniz = ";
25     cin >> cins;
26     cout << "Ayaksayisini Giriniz = ";
27     cin >> ayaksayisi;
28     cout << "Elsayisini Giriniz = ";
29     cin >> elsayisi;
30     cout << cins << "\n" << ayaksayisi << "\n" << elsayisi << endl;
31 }
32 class bitkiler :public canlilar
33 {
34 public:
35     bitkiler();
36 }
37 };

```

```

38 bitkiler::bitkiler()
39 {
40     cout << "Cinsini yaziniz =";
41     cin >> cins;
42     cout << "Turunu giriniz = ";
43     cin >> tur;
44 }
45 }
46 }
47 int main()
48 {
49     canlilar c;
50     hayvanlar h;
51     bitkiler b;
52     system("Pause");
53     return 0;
54 }
55 }

```

```

-----Canlilar Alemine Hosgeldiniz-----
-----Canlilar Alemine Hosgeldiniz-----
Cinsini Giriniz = hayvan
Ayaksayisini Giriniz = 2
Elsayisini Giriniz = 2
hayvan
2
2
-----Canlilar Alemine Hosgeldiniz-----
Cinsini yaziniz =bitki
Turunu giriniz = çam
bitki
çam
Press any key to continue . . .

```

```
1 #include "stdafx.h"
2 #include<iostream>
3 using namespace std;
4 class arac {
5 public:
6     arac();
7 };
8 arac::arac()
9 {
10     cout << "Bu bir arac" << endl;
11 }
12 class tekerlek
13 {
14 public:
15     tekerlek();
16 };
17 tekerlek::tekerlek()
18 {
19     cout << "Bu arac 4 tekerlekli" << endl;
20 }
21 class araba :public arac, public tekerlek
22 {
23 };
24
25
26
27 int main()
28 {
29     araba obj;
30     system("Pause");
31     return 0;
32 }
33
```

```
Bu bir arac
Bu arac 4 tekerlekli
Press any key to continue . . .
```

```
#include <iostream>
#include <conio.h>
using namespace std;
```

```
class sekil
{
public:
int alan;
string birim;
string sekilturu;
void alanyazdir()
{
cout<<sekilturu<<" "<<alan<<" "<<birim<<" "<<endl;
}
sekil()
{
birim="cm2";
}
};

class ucgen:public sekil
{
public:
int yukseklik,taban;
ucgen()
{
sekilturu="ucgen";
}
};
```

```
class dikdortgen:public sekil
{
public:
int en,boy;
dikdortgen()
{
sekilturu="dikdortgen";
}
};

class kare:public sekil
{
public:
int kenar;
kare()
{
sekilturu="kare";
}
};

class daire:public sekil
{
public:
int yaricap;
daire()
{
sekilturu="daire";
}
};
```

```
int main()
{
kare k1;
k1.kenar=10;
k1.alan=k1.kenar*k1.kenar;
k1.alanyazdir();

daire d1;
d1.yaricap=10;
d1.alan=3.14*d1.yaricap*d1.yaricap;
d1.alanyazdir();

}
```

```

1 #include <iostream>
2
3 using namespace std;
4
5 // Hayvan isminde bir sınıf olusturduk.
6 class Hayvan{
7     private:
8         int agirlik;
9         string isim;
10    public:
11        void agirligiBelirle(int kg)
12        {
13            agirlik = kg;
14        }
15        void ismiBelirle(string ismi)
16        {
17            isim = ismi;
18        }
19        int agirligiGetir()
20        {
21            return agirlik;
22        }
23        string ismiGetir()
24        {
25            return isim;
26        }
27        Hayvan();
28        ~Hayvan();

```

```

29 }; // noktali virgule dikkat etmeyi unutmayın.
30
31 Hayvan::Hayvan()
32 {
33     cout << "Bir hayvan nesnesi olusturuldu." << endl;
34 }
35
36 Hayvan::~~Hayvan()
37 {
38     cout << this->isim << " nesnesi yok edildi." << endl;
39 }
40
41 // Hayvan sınıfından kedi ve kopek olmak uzere iki sinif turettik.
42
43 class Kedi : public Hayvan
44 {
45     private:
46         string ses = "miyav";
47     public:
48         string sesiGetir()
49         {
50             return ses;
51         }
52         Kedi();
53         ~Kedi();
54 };
55

```

```

57 Kedi::Kedi(){
58     cout << "Bir kedi nesnesi olusturuldu." << endl;
59 }
60 Kedi::~~Kedi(){
61     cout << "Bir kedi nesnesi yok edildi." << endl;
62 }
63
64 class Kopek : public Hayvan
65 {
66 private:
67     string ses = "havhav";
68 public:
69     string sesiGetir()
70     {
71         return ses;
72     }
73     Kopek();
74     ~Kopek();
75 };
76
77 //Kopek sınıfı için constructor ve deconstructor
78 Kopek::Kopek(){
79     cout << "Bir kopek nesnesi olusturuldu." << endl;
80 }
81 Kopek::~~Kopek(){
82     cout << "Bir kopek nesnesi yok edildi." << endl;
83 }
84

```

29.03.2022

```

86 int main()
87 {
88     Kedi Tekir;
89     Kopek Karabas;
90
91     Tekir.ismiBelirle("Tekir");
92     Tekir.agirligiBelirle(5);
93
94     Karabas.ismiBelirle("Karabaş");
95     Karabas.agirligiBelirle(10);
96
97     // Hayvan sinifindan miras alinan ozellikleri cagirdik.
98     cout << Tekir.ismiGetir() << " isimli ";
99     cout << Tekir.agirligiGetir() << " agirligindaki ";
100
101     // Kedi sinifina ozel ozelligilimiz olan sesi cagirdik.
102     cout << "nesnemiz " << Tekir.sesiGetir() << " sesini cikarir." << endl;
103
104
105     // Hayvan sinifindan miras alinan ozellikleri cagirdik.
106     cout << Karabas.ismiGetir() << " isimli ";
107     cout << Karabas.agirligiGetir() << " agirligindaki ";
108
109     // Kedi sinifina ozel ozelligilimiz olan sesi cagirdik.
110     cout << "nesnemiz " << Karabas.sesiGetir() << " sesini cikarir." << endl;
111
112     return 0;
113 }

```

HAFTA 5

15

```

1  #include<iostream>
2  #include<cstdlib>
3  #include<string>
4  using namespace std;
5  //Basit bir inheritance(kalıtım,miras) örneği vermeye çalışacağım
6  class calisan{//kalıtım sınıfımız
7  public:
8      string ad;
9      string soyad;
10     string departman;
11     char tcno[11];
12
13     void bilgial(){
14         cout<<"Ad:"; cin>>ad;
15
16         cout<<"Soyad:"; cin>>soyad;
17
18         cout<<"Departman:"; cin>>departman;
19
20         cout<<"Tc No:"; cin>>tcno;
21     }
22 };
23
24 class muhendis:public calisan{//calisan sınıfının public değişken ve üye fonksiyonlarına kalıtım ile erişiyoruz
25     public://muhendis sınıfının public verileri
26     char calistigiProje[10];
27     void muhbilgi(){
28         cout<<"Calistigi Proje:"; cin>>calistigiProje;
29         cout<<endl;
30     }
31 };
32
33
34 class takimLideri:public calisan{
35     public://takimLideri sınıfının public verileri
36     string yonettigiTakim;
37     void liderbilgi(){
38         cout<<"Yonettigi Takim:"; cin>>yonettigiTakim;
39     }
40 };
41

```



```
51
52 int main()
53     muhendis muhIsimler;//muhendisten muhIsimler isminde nesne oluşturduk.
54     takimLideri tlidIsimler;//takimLiderinden tlidIsimler isminde nesne oluşturduk.
55
56     cout<<"Muhendis Bilgilerini Doldurunuz"<<endl<<endl;
57     muhIsimler.bilgial();
58     muhIsimler.muhbilgi();
59
60     cout<<"Takim Lideri Bilgilerini Doldurunuz"<<endl;
61     tlidIsimler.bilgial();
62     tlidIsimler.liderbilgi();
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
```

Muhendis Bilgilerini Doldurunuz

Ad:TOLGA

Soyad:DIRMIKCI

Departman:YAZILIM

Tc No:111111111111

Calistigi Proje:MOBIL

Takim Lideri Bilgilerini Doldurunuz

Ad:XXXXXXXX

Soyad:YYYYYYYY

Departman:YONETIM

Tc No:000000000000

Yönettiği Takim:ARGE_

Erişim İzinleri

Erişim	Sınıf İçerisinden Erişim	Türetilmiş Sınıftan Erişim	Dışarıdan Erişim
public	<i>evet</i>	<i>evet</i>	<i>evet</i>
protected	<i>evet</i>	<i>evet</i>	<i>hayır</i>
private	<i>evet</i>	<i>hayır</i>	<i>hayır</i>

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

class PublicDerived: public Base {
    // x is public
    // y is protected
    // z is not accessible from PublicDerived
};

class ProtectedDerived: protected Base {
    // x is protected
    // y is protected
    // z is not accessible from ProtectedDerived
};


class PrivateDerived: private Base {
    // x is private
    // y is private
    // z is not accessible from PrivateDerived
}
```

```
class Base
{
public:
    int m_public;
private:
    int m_private;
protected:
    int m_protected;

};

class Pub: public Base
{
public:
    Pub()
    {
        m_public = 1; // doğru: m_public public olarak kalıtılmıştır
        m_private = 2; // yanlış: m_private türetilmiş sınıftan erişilemez
        m_protected = 3; // doğru: m_protected protected olarak kalıtılmıştır
    }

};
```



```
int main()
{
    Base base;
    base.m_public = 1; // doğru
    base.m_private = 2; // yanlış: m_private erişilemez
    base.m_protected = 3; // yanlış: m_protected doğrudan erişilemez

    Pub pub;
    pub.m_public = 1; // doğru
    pub.m_private = 2; // yanlış: m_private Pub tarafından erişilemez
    pub.m_protected = 3; // yanlış: m_protected doğrudan erişilemez
```

Örnek: Private Inheritance

```
class Base
{
public:
    int m_public;
private:
    int m_private;
protected:
    int m_protected;

};


class Pri: private Base
{

public:

    Pri()
    {
        m_public = 1; // doğru
        m_private = 2; // yanlış: türetilmiş sınıflar ana sınıftaki private'a erişemez
        m_protected = 3; // doğru

    }

};
```




```
int main()
{

    Base base;

    base.m_public = 1; // doğru
    base.m_private = 2; // yanlış: m_private Base sınıfında private'dır
    base.m_protected = 3; // yanlış: m_protected Base sınıfta protected'dır

    Pri pri;
    pri.m_public = 1; // yanlış: m_public artık private'dır
    pri.m_private = 2; // yanlış: m_private artık erişilemezdir
    pri.m_protected = 3; // yanlış: m_protected artık private'dır

    return 0;
}
```

› https://www.youtube.com/watch?v=kopoLzvh5jY&ab_channel=OpenAI