BM102 Algoritma ve Programlama II

Sinif (Class)

İçerik



- 1. Sınıf Tanımı
- 2. Nesne Tanımı
- 3. İkili Kapsam Çözme Operatörü
- 4. UML ile Sınıfların Gösterimi Çözümlü Sorular

Hedefler



- Verilen tanımlamalardan sınıf ve nesneleri ayırtetme
- Veri üyesi ve üye fonksiyonlarla sınıf tanımı yapma
- public ve private erişim etiketlerinin kullanım farklarını anlatma
- Bir sınıZan nesne yaratma
- Nesnenin özelliklerine ve davranışlarına erişme
- İkili kapsam çözme operatörü ile üye fonksiyonları sınıfı tanımlama
- Sınıfa ait veri üyeleri için erişimci ve değiştirici fonksiyonları yazma
- Bir sınıf tanımını kullanarak UML sınıf diyagramını çizme
- Verilen UML sınıf diyagramını dikkate alarak sınıf tanımını yazma

Sinif

- Sınıf kavramı, nesne tabanlı programlamanın temelini oluşturur.
- Nesne tabanlı programlamanın ilk özelliği olan sarmalama (encapsulation), veri ve bu veriyle ilgili işlemlerin tek bir birimde toplanması gerektiğini belirtir.
- Gerçek hayatta etrafımıza baktığımızda canlı cansız birçok nesneyle karşılaşırız.







Nesne Örnekleri

Sinif...

 Bu nesnelerin her birinin durumu ve davranış özellikleri vardır. Bu örneklerde de görüldüğü gibi bir nesne, durum ve davranış özelliklerini içinde barındırır.

Köpek Nesnesi:

Durum Bilgileri:

İsim: Lola Renk: Sarı

Cins: Golden Retriever

Davranış Bilgileri:

Havlamak Koklamak

Televizyon Nesnesi:

<u>Durum Bilgileri:</u>

Marka: Toshiba Cins: Plazma

Boyut: 120

Davranış Bilgileri:

Açma/Kapama Kanal değiştirme

Çiçek Nesnesi:

<u>Durum Bilgileri:</u>

Cins: Gül

Renk: Kırmızı

<u>Davranış Bilgileri:</u>

Çiçek açma Solma







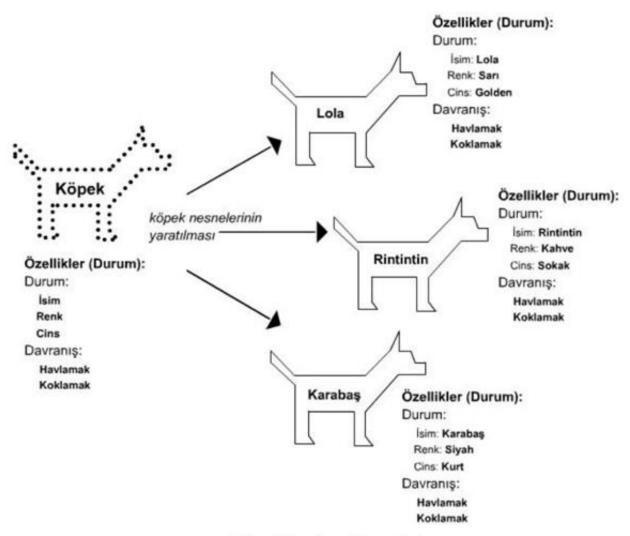
Sinif...

- Nesne tabanlı programlama mantığını kullanırsak, Lola nesnesi, tüm köpek nesnelerini oluşturan sınıfınbir örneğidir.
- Sınıf ise tüm köpek nesnelerini oluşturmak için kullanılan bir modeldir.
- Buradan yola çıkarak sınıf (class), bir nesneyi (object / instance)
 oluşturmak için gereken model veya plan olarak tanımlayabiliriz.

Köpek Sınıfı Durum: İsim Renk Cins Davranış: Havlamak Koklamak

Köpek Sınıfı Modeli

Sınıf...



Köpek Sınıfı ve Nesneleri

1. Sınıf Tanımı

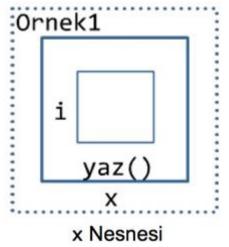
- Sınıf, çözülmesi gereken problemin içeriğine göre yeni bir veri tipi ve bu veri tipi ile yapılacak işlemleri tanımlamamıza olanak sağlar.
- Oluşturulan her nesnenin durum özellikleri veri üyesi (data member), davranış özellikleri ise üye fonksiyonlarla (member function) tanımlanır.

1. Sınıf Tanımı...

- Sınıf tanımında yer alan üyelere erişimi, farklı erişim etiketleri kullanarak sınırlandırabiliriz:
 - public (genel): bu etiket albnda tanımlanan üyelere programın her yerinden erişilebilir.
 - private (özel): bu etiket albnda tanımlanan üyelere sadece sınıf içerisinden erişilebilir.
- Sınıf tipinde tanımlanan değişkenlere nesne (object) adı verilir.
- Sınıf tanımı sırasında **public** veya **private** erişim etiketi belirtilmezse, tanımlanan sınıf üyelerinin erişim etiketleri **private** olarak varsayılır.

1. Sinif Tanimi...

```
#include <iostream>
using namespace std;
class Ornek1
     public:
          int i;
                                        // Veri üye
                                        // Fonksiyon üye
          void yaz()
               cout<<"i="<<i<<endl;
  };
int main()
    Ornek1 x;
                                        // Ornek1 sınıfı tipinde x nesnesi tanımlanır
     x.i = 5;
                                        // x nesnesinin i üyesine 5 atanır.
                                        // x nesnesinin yaz() fonksiyonu çağırılır.
     x.yaz();
     return 0;
```



Çıktı

i=5

1. Sınıf Tanımı...

 Aynı yapılarda olduğu gibi bir nesnenin veri veya fonksiyon üyelerine erişmek için aşağıdaki sözdiziminde görüldüğü gibi nokta (.) operatörünü kullanmamız gerekir.

nesneAdı.üyeİsmi

Örnek:

```
x.i = 5;
x.yaz();
```

1. Sinif Tanimi...

```
#include <iostream>
using namespace std;
class Ornek2
     public:
          int i;
                                        // Her yerden erişilebilen üyeler
         void yaz()
                                        // Üye fonksiyondan özel y üyesine erişilebilir.
               y = 5.2;
               z = 9;
               cout<<"i= "<<i<<endl;
               cout<<"y= "<<y<<endl;
               cout<<"z= "<<z <<endl;
     private:
                                        // Sadece sınıf içinden erişilebilen üyeler
        float y;
         int z;
int main()
                                        // Ornek2 sınıfı tipinde nsn isimli nesne tanımı
     Ornek2 nsn;
     nsn.i = 5;
                                        // nsn nesnesinin i üyesine 5 atanır.
                                        // nsn nesnesinin yaz() fonksiyonu çağırılır.
     nsn.yaz();
     return 0;
```

Çıktı

```
i= 5
y= 5.2
z= 9
```

1. Sınıf Tanımı...

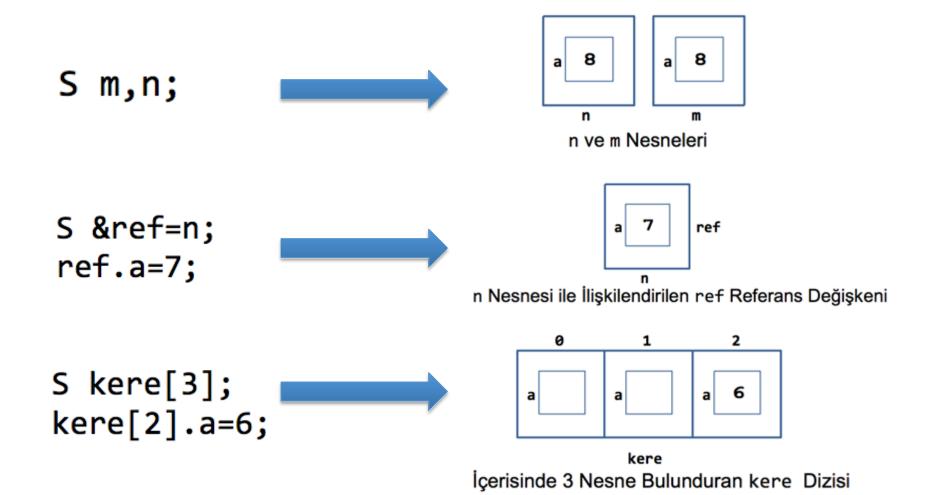
Aşağıda görüldüğü üzere public olarak etiketlenmiş üyesine main()
fonksiyonundan erişebilirken, private olarak etiketlenmiş y ve
z üyelerine erişmemiz mümkün olmamıştır..

2. Nesne Tanımı

```
#include <iostream>
 using namespace std;
                                        // S isimli sınıf tanımı
 class S
      public:
                                                                           Çıktı
           int a;
 };
int main()
                // S tipinde m ve n isminde iki nesne tanımlanır.
    S m,n;
    n.a=8; // n nesnesinin a üye'sine 8 atanır.
    cout<<" n nesnesinin a uyesi: "<<n.a<<endl;</pre>
                     // n nesnesi m nesnesine kopyalanır.
    m=n;
    cout<<" m nesnesinin a uyesi: "<<m.a<<endl;</pre>
    S &ref=n; // n nesnesini gösteren referans değişkeni ref tanımlanır.
    cout<<" ref referansinin a uyesi: "<<ref.a<<endl;</pre>
    ref.a =7; //refin referans olduğu nesnenin a üyesine 7 atanır.
    cout<<" ref referansinin a uyesi: "<<ref.a<<endl;</pre>
    S kere[3]; // S tipinde 3 nesneyi içeren bir dizi tanımlanır
    kere[2].a=6; // kere dizisinin 2.indeksli nesnesinin a uyesine 6 atanır.
    cout<<" kere dizisinin 2. nesnesinin a uyesi: "<<kere[2].a<<endl;</pre>
    return 0:
```

```
n nesnesinin a uyesi: 8
m nesnesinin a uyesi: 8
ref referansinin a uyesi: 8
ref referansinin a uyesi: 7
kere dizisinin 2. nesnesinin a uyesi: 6
```

2. Nesne Tanımı...



3. İkili Kapsam Çözme Operatörü(::)

• Sınıfın dışında tanımlanan fonksiyonun hangi sınıfın üye fonksiyonu olduğunu belirtmek amacıyla *ikili kapsam çözme operatörü* (::) (binary scope resolution operator) kullanılır.

```
döndürülenVeriTipi SınıfAdı :: fonksiyonAdı (parametre listesi){
...
}
```

3. İkili Kapsam Çözme Operatörü (::)...

```
#include <iostream>
using namespace std;
class Ornek4
    public:
        int i;
        void yaz();
                            // Fonksiyon protipi
};
void Ornek4::yaz()
                            // Fonksiyon tanımı
    cout<<"Merhaba"<<endl;
    cout<<"i degeri = "<<i<<endl;
int main()
    Ornek4 x;
    x.i=5;
    x.yaz();
    return 0;
```

Çıktı

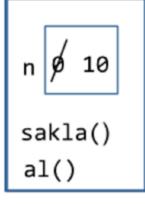
Merhaba i degeri = 5

3. İkili Kapsam Çözme Operatörü (::)...

```
#include <iostream>
using namespace std;
class Sayi
    int n;
                                          // n üyesinin private olduğu varsayılır
    public:
                                         // Fonksiyon prototipi ve varsayılan parametre
         void sakla (int=0);
         int al();
                                         // Fonksiyon prototipi
void Sayi::sakla (int x)
                                         // Değiştirici fonksiyon tanımı
    n=x;
                                         // Erişimci fonksiyon tanımı
int Sayi::al()
    return n;
int main()
    Sayi no;
    no.sakla();
    cout<<"n = "<<no.al()<<endl;
    no.sakla(10);
    cout<<"n = "<<no.al()<<endl;
    return 0;
```

Çıktı

```
n = 0
n = 10
```



no no Nesnesi

3. İkili Kapsam Çözme Operatörü (::)...

- Bir sınıf üyesinin değerini değiştiren fonksiyonlara değiştirici fonksiyon (mutator function) adı verilir.
- Sınıf üyesini değiştirmeyen, sadece bu üyeleri döndüren fonksiyonlara *erişimci fonksiyon* (accessor function) ismi verilir.

3. İkili Kapsam Çözme Operatörü(::)...

```
#include <iostream>
using namespace std;
#define SINIF MEVCUDU 50
class Ogrenci
                                                 // private veri üyeleri
    int no, vize, final;
    public:
                                                 // Erişimci fonksiyonlar
         int noAl(){return no;}
        int vizeAl(){return vize;}
        int finalAl(){return final;}
        void noAta(int no){no= no;}
                                                 // Değiştirici fonksiyonlar
        void vizeAta(int _vize){vize=_vize;}
        void finalAta(int final){final= final;}
};
```

```
int main()
                                                   // 50 öğrencilik dizi tanımı
    Ogrenci sinif[SINIF MEVCUDU];
    int no, vize, final;
    float ortalama;
    for (int i=0;i<SINIF MEVCUDU;i++){</pre>
         cout<<endl<<"Ogrenci numarasi, vize ve finali giriniz: ";</pre>
         cin>>no>>vize>>final;
         Ogrenci ogrenci;
                                            // ogrenci nesnesine deger aktarımı
         ogrenci.noAta(no);
         ogrenci.vizeAta(vize);
         ogrenci.finalAta(final);
                                                   // sinif dizisine eleman ekleme
         sinif[i] = ogrenci;
    for (int i=0;i<SINIF MEVCUDU;i++){</pre>
                                                   // ortalama hesabi
         ortalama = 0.4*sinif[i].vizeAl() + 0.6*sinif[i].finalAl();
         cout<<sinif[i].noAl()<<" numarali ogrencinin ortalamasi:"</pre>
                                 <<ortalama<<endl;
    return 0;
```

Çıktı

```
Ogrenci numarasi, vize ve finali giriniz: 123 60 70 Ogrenci numarasi, vize ve finali giriniz: 524 80 90 ...
123 numarali ogrencinin ortalamasi:66 524 numarali ogrencinin ortalamasi:86 ...
```

4. UML ile Sınıfların Gösterimi

- UML (Unified Modeling Language) (Tümleşik Modelleme Dili), nesne tabanlı analiz ve tasarım sürecinde sınıflar ve sınıflar arası ilişkiyi göstermek amacı ile kullanılan bir modelleme aracıdır.
- UML, ihtiyaca yönelik birçok farklı gösterim modelini içerir.
- Sınıf diyagramları bu gösterim modellerinden biridir.
 Sınıf diyagramları ile tasarlanan sınıfın durumu (veri üyeleri) ve davranışları (fonksiyonlar) görsel olarak tanımlanır.

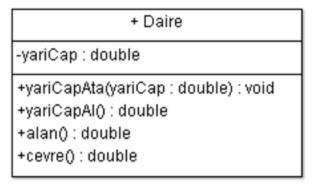
4. UML ile Sınıfların Gösterimi...



Sınıf Diyagramı Şablonu

Bu diyagramda kullanılan '+' işareti, söz konusu üyenin public olduğunu, '-' işareti ise üyenin private olduğunu göstermektedir.

Örnek



Daire Sınıf Diyagramı

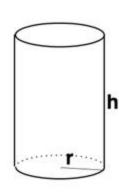
4. UML ile Sınıfların Gösterimi...

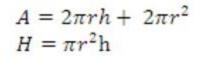
```
#include <iostream>
using namespace std;
class Daire
                                                                                       + Daire
                                                                          -yariCap : double
    double yariCap;
    public:
                                                                          +yariCapAta(yariCap : double) : void
         void yariCapAta(double yariCap){yariCap= yariCap;}
                                                                          +yariCapAl(): double
         double yariCapAl(){ return yariCap; }
                                                                          +alan(): double
         double alan (){ return 3.14*yariCap*yariCap; }
                                                                          +cevre(): double
         double cevre(){ return 2*3.14*yariCap;
 };
                                                                               Daire Sınıf Diyagramı
int main()
    Daire d;
                                    // Daire tipinde d isimli bir nesne yaratır.
    double r;
                                                                           Çıktı
    cout<<endl<<"Daire yaricapi: ";</pre>
    cin>>r:
                                                                           Daire yaricapi: 4
    d.yariCapAta(r);
                                    // Okunan değer yarı çapa atanır
                                                                           r=4 alan=50.24 cevre=25.12
    double alan = d.alan();
                                    // Alan hesaplanır
    double cevre = d.cevre();
                                    // Cevre hesaplanır
    cout<<"r="<<d.yariCapAl() <<" alan="<<alan<<" cevre="<<cevre<<endl;</pre>
    return 0;
```

Soru

Silindir sınıfına ait sınıf diyagramı ve bu sınıfa ait fonksiyonların gerçekleştirilmesinde kullanılacak formüller aşağıda verilmiştir. Program çalıştırıldığında aşağıdaki örnek çıktı elde edilecek şekilde Silindir sınıfını yazınız.

+ Silindir
-yukseklik : int
-tabanYaricapi : int
+yuzeyAlaniHesapla() : double
+hacimHesapla() : double







Örnek Çıktı

Yukseklik (h):5 Taban yaricapı (r) :3

Yuzey alani (A):150.796

Hacmi (H):141.372

Cevap

```
#include <iostream>
#define PI 3.14159265
using namespace std;
class Silindir
    public:
        int r;
        int h;
        float yuzelselAlaniHesapla();
        float hacimHesapla();
};
float Silindir::yuzelselAlaniHesapla()
    return 2*PI*r*h+2*PI*r*r;
float Silindir::hacimHesapla()
    return PI*r*r*h;
int main()
    Silindir silindir;
    cout<<"Yukseklik (h):";
    cin>>silindir.h;
    cout<<"Taban yaricapi (r):";
    cin>>silindir.r;
    cout<<"----"<<endl;
    cout<<"Yuzey alani (A):"<<silindir.yuzelselAlaniHesapla()<<endl;</pre>
    cout<<"Hacim (H):"<<silindir.hacimHesapla()<<endl;</pre>
    return 0:
```

Soru

Bir kola makinesi için program yazalım. Programda KolaMakinesi isimli aşağıdaki bileşenlerden oluşan bir sınıf tanımlayın:

- Veri Üyeleri:
 - o Bir kolanın fiyatı
 - Makinada kalan kolaların sayısı
- · Üye fonksiyonlar:
 - ilkYukleme(): fiyat ve kola sayısına parametre olarak gelen değerleri atar. Varsayılan fiyat değeri 2.5 lira, sayı değeri ise 100 olarak alınır.
 - kolaVerme(): kullanıcıya fiyatı gösterir. Kullanıcı yeterli para girdiyse ve makinede kola varsa kullanıcıya bir kola verir ve kullanıcının para üstünü hesaplar.
 - o goster (): Kola fiyatını ve makinede kalan kola sayısını yazar.

main() fonksiyonunda okulunuz için bir kola makinesi nesnesi yaratın, bir kullanıcının kola alması durumunda makinenizi örnek çıktıda olduğu gibi güncelleyin.

Örnek Çıktı

Kolanin fiyati: 2.5 lira.
Odemeniz: 3
Kolaniz veriliyor.
Para ustunuz: 50 kurus
Son durum:
 Kolanin fiyati: 2.5 lira.
 Kalan Kola sayisi : 99



Cevap

```
#include <iostream>
                                                  void KolaMakinesi::ilkYukleme (float f,int s){
using namespace std;
                                                       fiyat=f;
class KolaMakinesi
                                                       sayi=s;
                                                   }
    float fiyat;
                                                  void KolaMakinesi::kolaVerme()
    int sayi;
    public:
                                                       float ode, fark;
        void ilkYukleme (float f=2.5,int s=100);
       void kolaVerme();
                                                       cout<<"Kolanin fiyati: "<<fiyat<<" lira."<<endl;</pre>
       void goster();
                                                       cout<<"Odemeniz: ";
};
                                                       cin>>ode;
                                                       if ((ode>=fiyat)&&(sayi>0)){
                                                           cout<<"Kolaniz veriliyor."<<endl;</pre>
                                                           sayi--;
                                                           fark=ode-fiyat;
                                                           if (fark>0){
                                                                cout<<"Para ustunuz: ";
                                                                if (fark>1.0)
                                                                    cout<<fark<<" lira"<<endl;
                                                                else
                                                                    cout<<fark*100<<" kurus"<<endl;
                                                       }else
                                                           cout<<"Kola verilemiyor";</pre>
                                                  }
```

Cevap

```
void KolaMakinesi::goster()
    cout<<"Son durum:"<<endl;
    cout<<"Kolanin fiyati: "<<fiyat<<endl;</pre>
    cout<<"Kalan Kola sayisi : "<<sayi<<endl;</pre>
int main()
    KolaMakinesi univ;
    univ.ilkYukleme();
    univ.kolaVerme();
    univ.goster ();
    return 0;
```

Soru

A sınıfında tanımlı değişken ve fonksiyonların sınıf dışından erişilebilirliklerini gösteren tablo aşağıda verilmiştir. Bu tabloya uygun olarak A sınıfını değişken ve fonksiyonları ile tanımlayınız.

	X	Y	f1()	f2()
A sınıf içinden erişim	1	1	1	/
A sınıfı dışından erişim		1		V