ANKARA
HACI BAYRAM VELİ ÜNİVERSİTESİ
UZAKTAN EĞİTİM UYGULAMA VE ARAŞTIRMA MERKEZİ

İNTERNET PROGRAMCILIĞI

Dersin Genel Hedefleri

Inheritance ve Polymorphism kavramlarını tanımlamak ve uygulama içinde kullanabilmek.

HAFTA 2. NESNE TABANLI PROGRAMLAMA

Bölüm Hedefleri

- Öğrencilerin Nesne Tabanlı Programlama alıştırmaları yaparak, Inheritance ve Polymorphism unsurlarını kavramalarını sağlamak

1.1. Inheritance (Kalıtım)

Kalıtım ile sınıf yapılarımız birbirinden türetilebilir.

Bir sınıf herhangi bir üst sınıftan türetildiği zaman, türemiş olduğu sınıfın içerisindeki public, default ve protected özellikteki bütün özelliklere sahip olur

Bir bankanın kredi satışı için oluşturmuş olduğu çeşitli sınıf yapıları bulunsun.

Bu sınıf yapılarında çeşitli hesaplamalar, indirimler ve ödeme seçeneklerinin imkanı sağlansın.

Örnekte inheritance (kalıtım) mantığı; FarmerCreditManager sınıfı ile CreditManager sınıfı arasında gerçekleşmektedir.

1.1. Inheritance (Kalıtım)

Kredi işlemlerinin yönetileceği genel bir sınıf yapısı olarak "CreditManager" isminde bir üst sınıf oluşturuldu, içine çeşitli değişkenler tanımlandı. Giriş yapabilmek için constractor yapısı tanımlandı. Daha sonra kullanıcıların kredi çekmesini sağlayacak ve çekmiş olduğu kredinin bilgilerini göreceği iki adet metot tanımlandı

```
class CreditManager // CreditManager sınıfını tanımlar.

{

        public CreditManager() //Parametresiz bir constructor tanımlar.
Classların içi dolu gelebilsin.
        {
                creditName, creditType, creditInterest, creditPaymentMonth
        }
        public double takeCredit(double creditValue) // Kredi alma metodu
(hesaplamalar)
        {
        }

        public void creditInfo() // Kredi bilgilerini ekranda gösteren metot.
        {
        }
}
```

1.1. Inheritance (Kalıtım)

"FarmCreditManager" isminde bir alt sınıf oluşturulur.

Constructor bir yapı oluşturarak değişkenlerin değerlerini düzenlenir.

Çiftçi kredisi için özel indirim yapan metot tanımlanır.

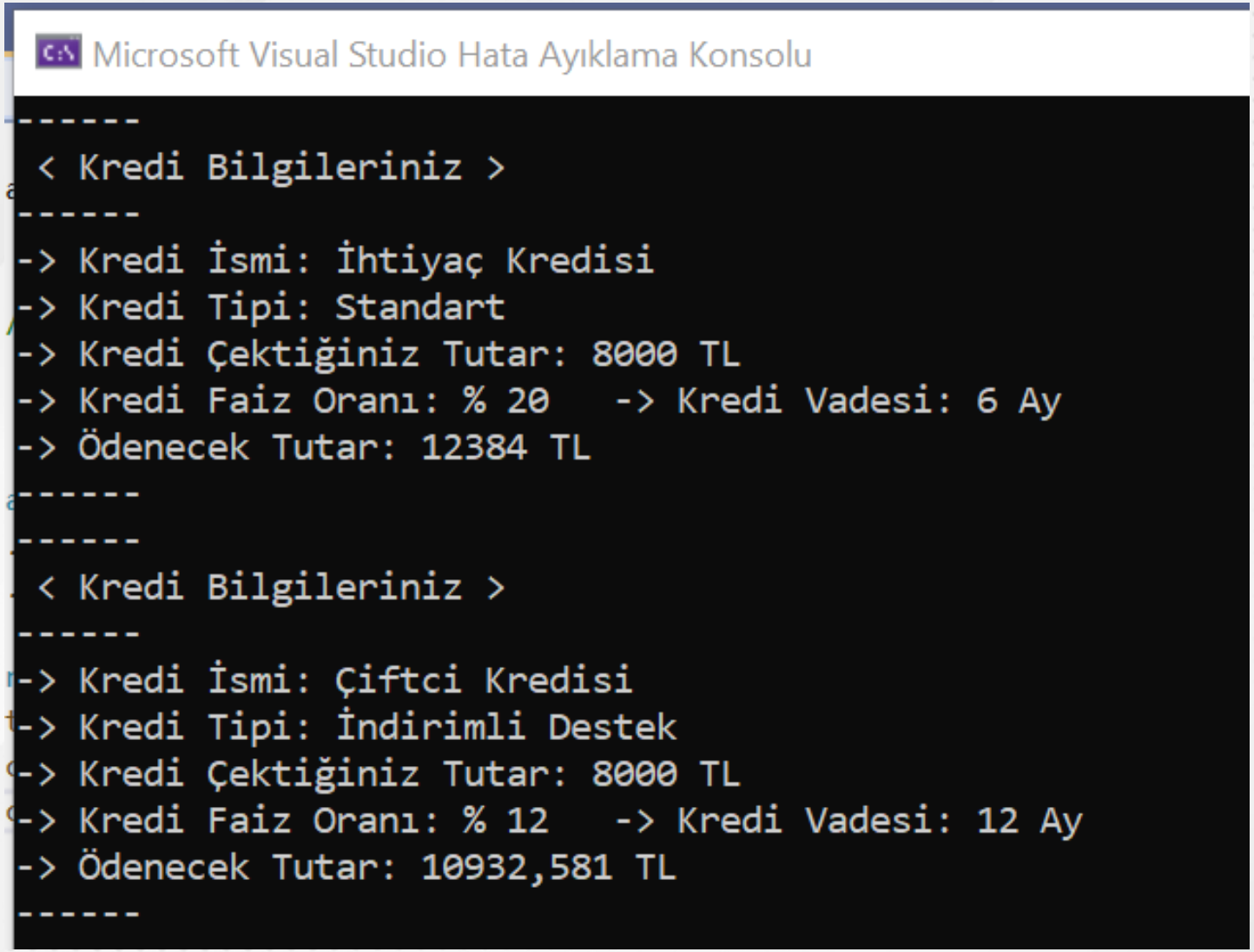
```
class FarmerCreditManager : CreditManager // CreditManager sınıfından türetilen FarmerCreditManager sınıfını tanımlar.
{
        public FarmerCreditManager() // Parametresiz bir constructor tanımlar.
        {
                creditName, creditType, creditInterest, creditPaymentMonth
        }
        public void creditDisc() // Çiftçi kredisinde ek farklı bir indirim.
        {
                takeCredit
        }
}
```

1.1. Inheritance (Kalıtım)

```
class Program // Ana program sınıfını tanımlar.
{
        static void Main(string[] args) // Ana programın giriş noktasını belirler.
        {
                CreditManager
                manager.takeCredit();
                manager.creditInfo();
                FarmerCreditManager
                farmer.takeCredit();
                farmer.creditDisc();
                farmer.creditInfo();
        }

}
```

1.1. Inheritance (Kalıtım)



1.2. Polymorphism (Çok Biçimlilik)

Programımızda oluşan bir nesne yapısının birbirinden farklı nesneler şeklinde davranmasını sağlayan yapı.

- kurmuş olduğumuz yapılara esneklik sağlar
- okunabilirlik oranında artış
- Hata çözümlerini kolaylaştırır

Bir firmanın çalışan elemanlarının bilgilerini kayıt altına alarak bazı hesaplamalar yürütebilmesine imkan veren bir otomasyon sistemi olduğu varsayılsın.

Bu otomasyon sisteminde eklenen çalışanlar

- mühendisler (mühendis, kıdemli mühendis, uzman mühendis, stajyer mühendis, stajyer) ve
- diğer çalışanlar (vasıflı, vasıfsız, kıdemli, uzman, stajyer)

olarak iki alana ayırarak çeşitli özel işlemlere tabii tutulsun.

1.2. Polymorphism (Çok Biçimlilik)

İki kısımdan oluşacak bu kayıt sistemimizin ilk ve genel sınıfı olacak **"Employe"** isminde bir sınıf oluşturuldu.

Sınıf yapısının içerisine çalışan kişilerin bilgilerinin not alınabileceği, işlenebileceğim

bazı değişkenler (`salary, hday, raise, job, name, surname`) ve

metot yapıları (`pSalary, pHday, pJob`) tanımlandı.

```
class Employe //// Employe adında bir sınıf tanımlanır.
{
        public int pSalary // Maaş değişkenine erişmek için property tanımlar.
        {
                get{} set{}
        }
        public int pHday // İzin günü değişkenine erişmek için property tanımlar
        {
                get{} set{}
        }
        public string pJob // Görev değişkenine erişmek için property tanımlar.
        {
                get{} set{}
        }
…
```

1.2. Polymorphism (Çok Biçimlilik)

sınıf yapısı kullanarak bir çalışanın eklenmesini sağlayacak **Constructor** metotları tanımlandı.

Construct or yapıları

```
….
        public Employe() // Parametresiz bir constructor tanımlar.
        {
        }
        public Employe(string name, string surname) // İsim ve soyisim
alarak bir constructor tanımlar.
        {
        }

        public Employe(string name, string surname, int salary) // İsim,
soyisim ve maaş alarak bir constructor tanımlar.
        {
        }
}
```

1.2. Polymorphism (Çok Biçimlilik)

createJob, setHday, makeRaise, infoBox, raiseBox metotları tanımlandı.

```
public string createJob(int salary) // Maaşa göre görev oluşturan metod. Kod satırı içindeki koşullara göre çalışanın görevini
belirler ve döndürür..

{

}
public void setHday()  // Çalışanın görevine ve maaşına göre izin gününü ayarlar
{

}

public void makeRaise(int value) // Çalışanın görevine göre zam yapar ve ilgili metotları çağırır. Mühendislere ayrı zam.
{
      createJob(); setHday();
}
public void infoBox() //  Bilgi metodu: Çalışanın bilgilerini ekrana yazdırır
{

}
public void raiseBox() // Zam metodu: Yapılan zamın bilgilerini ekrana yazdırır
{

}
            …
```

1.2. Polymorphism (Çok Biçimlilik)

mühendislerin kayıt olmasını sağlayacak **yeni bir sınıf «engineer»** oluşturulur.

kayıt işlemini gerçekleştirecek **Constructor** yapıları kurulur.

```
class Engineer : Employe // Mühendis sınıfını tanımlar ve Employe sınıfından
türetilir. // İçinde Mühendis sınıfının constructor ve diğer metotları
tanımlanır
{
    public Engineer(string name, string surname)
    {
    }
    public Engineer(string name, string surname, int salary)
    {
    }
}
```

Constructor

1.2. Polymorphism (Çok Biçimlilik)

“Main” sınıfıma giderek çalışanların kayıtlarını oluşturulur.

```
Program //Ana program sınıfı
{
        static void Main(string[] args) // Programın ana kod bloğu
}
```

1.2. Polymorphism (Çok Biçimlilik)

“Main” sınıfıma giderek çalışanların kayıtlarını oluşturulur.

```
Program //Ana program sınıfı
{
        static void Main(string[] args) // Programın ana kod bloğu
}
```

1.2. Polymorphism (Çok Biçimlilik)

Microsoft Visual Studio Hata Ayıklama Konsolu

```
------
< Vasıfsız Çalışan Bilgileri >
------
-> Personel İsmi: Mert
-> Personel Soyismi: Çağlayan
-> Görevi: Vasıfsız Çalışan
-> Maaşı: 3500 TL
-> İzin Günü: 14 Gün
------

------
< Mühendis Bilgileri >
------
-> Personel İsmi: Ömer Faruk
-> Personel Soyismi: Yıldız
-> Görevi: Mühendis
-> Maaşı: 5500 TL
-> İzin Günü: 21 Gün
------
```

```
------
 < Vasıflı Çalışan Zam Uygulaması >
------
-> Personelin Eski Maaşı: 3500 TL
-> Yeni Verilecek Maaş: 4000 TL
------
------
 < Kıdemli Mühendis Zam Uygulaması >
------
-> Personelin Eski Maaşı: 5500 TL
-> Yeni Verilecek Maaş: 6820 TL
------
```

Bölüm Özeti

- Inheritance (Kalıtım) ile sınıf yapılarımız birbirinden türetilebilir.
- Polymorphism (Çok Biçimlilik) programımızda oluşan bir nesne yapısının birbirinden farklı nesneler şeklinde davranmasını sağlayan yapıdır.

Değerlendirme Soruları

BÖLÜM SORULARI

- 1) Inheritance (Kalıtım) kavramının işlevini açıklayınız.
- 2) Polymorphism (Çok Biçimlilik) kavramının işlevini açıklayınız.