



ANKARA  
HACI BAYRAM VELİ ÜNİVERSİTESİ  
**UZAKTAN EĞİTİM UYGULAMA VE  
ARAŞTIRMA MERKEZİ**

# İNTERNET PROGRAMCILIĞI

Bu bölüm kapsamında C# alıştırmaları yapılmıştır.

## 1. C# ALIŞTIRMALARI

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. İşte detaylar:

1. **CreditManager Sınıfı:** Bu sınıf, kredi işlemlerini yönetmek için temel bir yapı sağlar. creditName, creditType, creditValue, creditInterest, creditPaymentMonth, ve creditLastMoney gibi özellikleri içerir. takeCredit metodu, belirli bir miktar kredi almak için kullanılırken, creditInfo metodu ise kredi bilgilerini göstermek için kullanılır.
2. **FarmerCreditManager Sınıfı:** Bu sınıf, CreditManager sınıfından türetilmiştir, yani CreditManager sınıfının tüm özelliklerini ve metotlarını miras almıştır. FarmerCreditManager, çiftçilere özel kredi işlemlerini yönetmek için özelleştirilmiştir. FarmerCreditManager, CreditManager sınıfının tüm özelliklerine ve metodlarına sahip olduğu için, takeCredit ve creditInfo metodlarını doğrudan kullanabilir.

```
using System;
namespace InheritanceApp
{
    class CreditManager // CreditManager sınıfını tanımlar.
    {
        protected string creditName, creditType, mark = new string('-', 6); // Kredi bilgilerini ve işaretleme karakterini içeren özellikler.
        protected double creditValue, creditInterest, creditPaymentMonth, creditLastMoney = 0; // Kredi değeri, faiz oranı, ödeme süresi ve son kalan kredi miktarı gibi özellikler.

        public CreditManager() // Parametresiz bir constructor tanımlar.
        {
            ///---> Default Value <---\\
            this.creditName = "İhtiyaç Kredisi";
            this.creditType = "Standart";
            this.creditInterest = 0.20;
            this.creditPaymentMonth = 6;
        }

        public double takeCredit(double creditValue) // Kredi alma metodu. Krediyi alırken yapılacak hesaplamalar ve kredi bilgilerini günceller.
        {
            this.creditValue = creditValue;
            creditLastMoney += creditValue;
            double creditPayment = (creditLastMoney / creditPaymentMonth);
            for (int i = 0; i < creditPaymentMonth; i++)
            {
                creditLastMoney += (creditValue * (creditInterest - (i / 1000) + (((creditPayment / 1000) * creditPaymentMonth)) / 1000) - creditPayment * 0.7);
            }
            return creditLastMoney = Math.Round(creditLastMoney, 2);
        }

        public void creditInfo() // Kredi bilgilerini gösteren metot.
        {
            Console.WriteLine("{0}\n < Kredi Bilgileriniz > \n{1}\n-> Kredi İsmi: {2}\n-> Kredi Tipi: {3}\n-> Kredi Çektiğiniz Tutar: {4} TL\n-> Kredi Faiz Oranı: % {5} -> Kredi Vadesi: {6} Ay\n-> Ödenecek Tutar: {7} TL\n{8}"
                , mark, mark, creditName, creditType, creditValue, (creditInterest * 100), creditPaymentMonth, creditLastMoney, mark);
        }
    }

    class FarmerCreditManager : CreditManager // CreditManager sınıfından türetilen FarmerCreditManager sınıfını tanımlar.
    {
        public FarmerCreditManager() // Parametresiz bir constructor tanımlar. Çiftçi kredisi özelinde kredi bilgilerini tanımlar.
        {
            this.creditName = "Çiftçi Kredisi";
            this.creditType = "İndirimli Destek";
            this.creditInterest = 0.12;
        }
    }
}
```

```
this.creditPaymentMonth = 12;
}
public void creditDisc() // Çiftçi kredisi indirimi hesaplama metodu. Çiftçi kredisi indirimlerini hesaplar ve kredi bilgilerini günceller.
{
    creditLastMoney = takeCredit(creditValue) - (creditValue * (creditPaymentMonth / 100) + (creditLastMoney * creditInterest * 3.5));
}
}
class Program // Ana program sınıfını tanımlar.
{
    static void Main(string[] args) // Ana programın giriş noktasını belirler.
    {
        CreditManager manager = new CreditManager(); // CreditManager sınıfından bir nesne oluşturulur ve kredi alım işlemi gerçekleştirilir.
        manager.takeCredit(8000);
        manager.creditInfo(); // Kredi bilgileri ekrana yazdırılır.
        FarmerCreditManager farmer = new FarmerCreditManager(); // FarmerCreditManager sınıfından bir nesne oluşturulur ve çiftçi kredisi alım işlemi gerçekleştirilir.
        farmer.takeCredit(8000);
        farmer.creditDisc(); // Çiftçi kredisi indirimi yapılır ve kredi bilgileri ekrana yazdırılır.
        farmer.creditInfo();
    }
}
}
```

### ÇALIŞMA SORUSU

Sınıf ve Kalıtım kavramları hangi amaçla kullanılır açıklayınız.