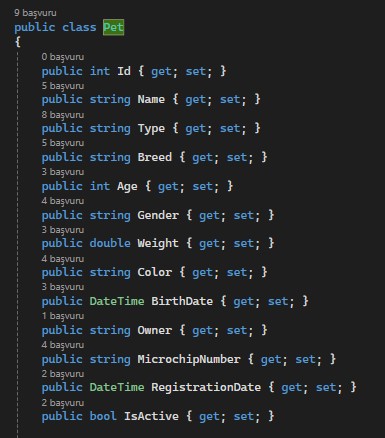
**Evcil Hayvan Takip Sistemi (Aşı ve bakım planlayıcı) - OOP Proje Raporu**

PetTakipp Uygulaması C# ve SQL Server kullanılarak geliştirilmiştir. Evcil hayvanların kayıt, bakım, tedavi ve aşı süreçlerini kolayca yönetmeye olanak tanıyan bu uygulama sayesinde kullanıcılar; hangi bakımın ne zaman yapıldığını, uygulanan aşıları, teşhisleri, tedavileri, cerrahi müdahaleleri ve kullanılan ilaçları detaylı şekilde görüntüleyebilir, böylece dostlarının sağlık geçmişini eksiksiz bir şekilde takip edebilirler.

**ENCAPSULATION (KAPSÜLLEME) PRENSİBİ - Pet Sınıfında Kapsülleme yapılmıştır.**

Property, veriye erişimi kontrol eden bir yapıdır. Get ve set ile birlikte kullanılır. Get ve set kullanımı ile Veri erişimi kontrollü bir şekilde sağlanıyor. Bu sayede Sınıf içerisindeki verilere dışarıdan doğrudan erişilemez. Veri güvenliği sağlanır.

**Kalıtım Hiyerarşisi**

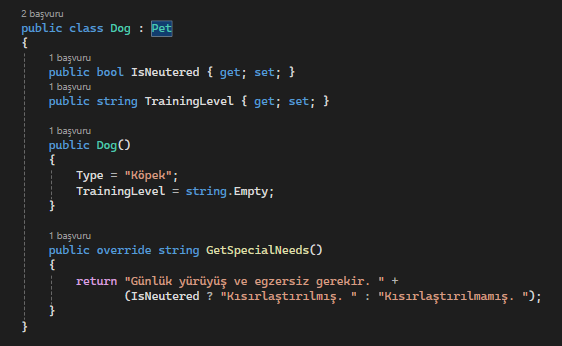
Pet (Base Class) Temel Sınıf

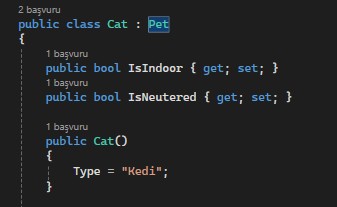
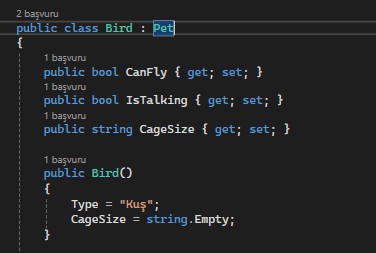
├── Dog Türetilmiş Sınıf

├── Cat Türetilmiş Sınıf

└── Bird Türetilmiş Sınıf

**Türetilmiş Sınıflar**



**Kalıtım Sayesinde**

Ortak özellikler Pet sınıfında tanımlandığından kod tekrarları azalır. Yeni hayvan türleri kolayca eklenir. Tip Güvenliği vardır.

**POLYMORPHISM (ÇOK BİÇİMLİLİK)**

private Pet CreatePetFromForm()

{

string type = cmbType.SelectedItem?.ToString();

Pet pet;

switch (type)

{

case "Köpek":

pet = new Dog();

break;

case "Kedi":

pet = new Cat();

break;

case "Kuş":

pet = new Bird();

break;

default:

pet = new Pet();

break;

}

pet.Name = txtName.Text;

pet.Type = type;

pet.Breed = txtBreed.Text;

pet.Age = (int)numAge.Value;

pet.Gender = cmbGender.SelectedItem.ToString();

pet.Weight = (double)numWeight.Value;

pet.Color = txtColor.Text;

pet.BirthDate = dtpBirthDate.Value;

pet.Owner = txtOwner.Text;

pet.MicrochipNumber = txtMicrochip.Text;

pet.RegistrationDate = DateTime.Now;

pet.IsActive = true;

return pet;

}

**Method Overriding ile Polimorfizm**

public virtual string GetSpecialNeeds()

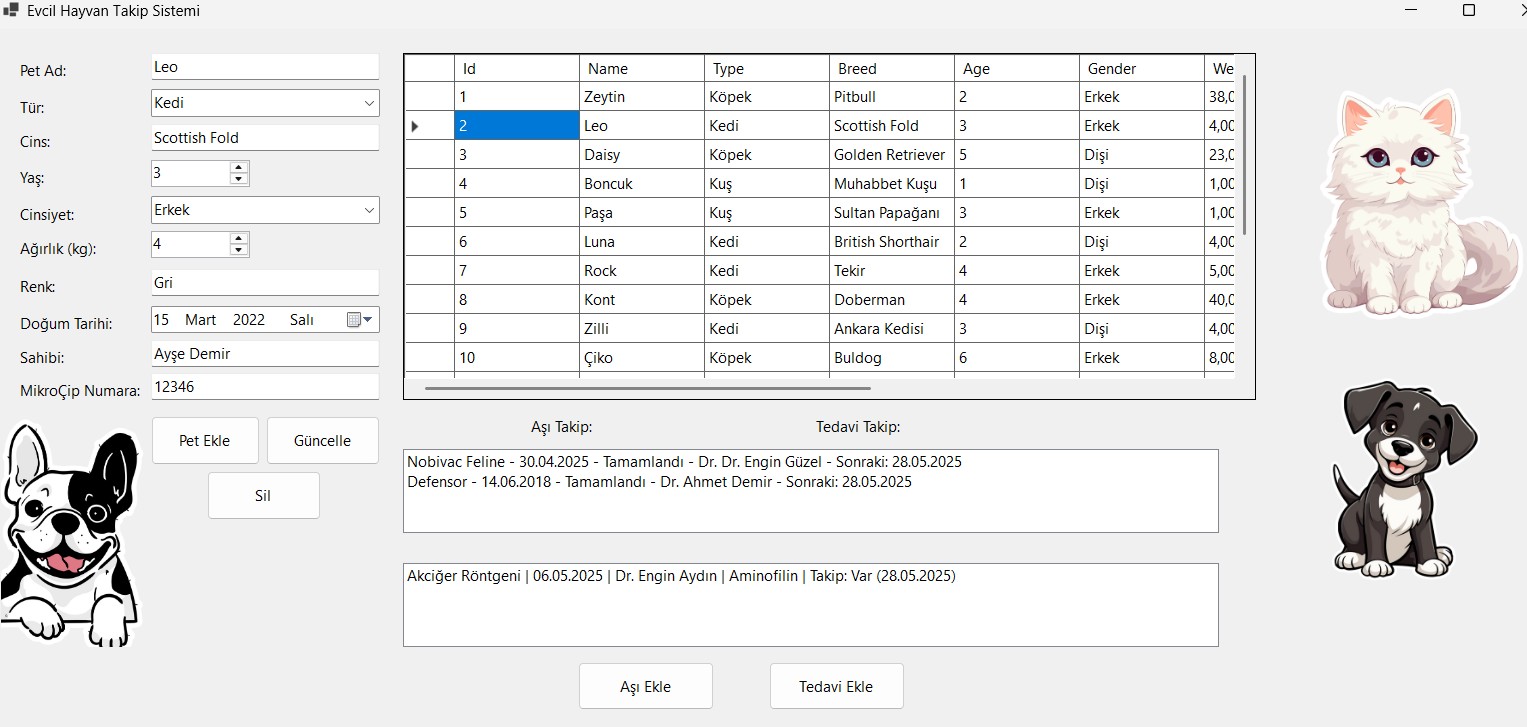
{

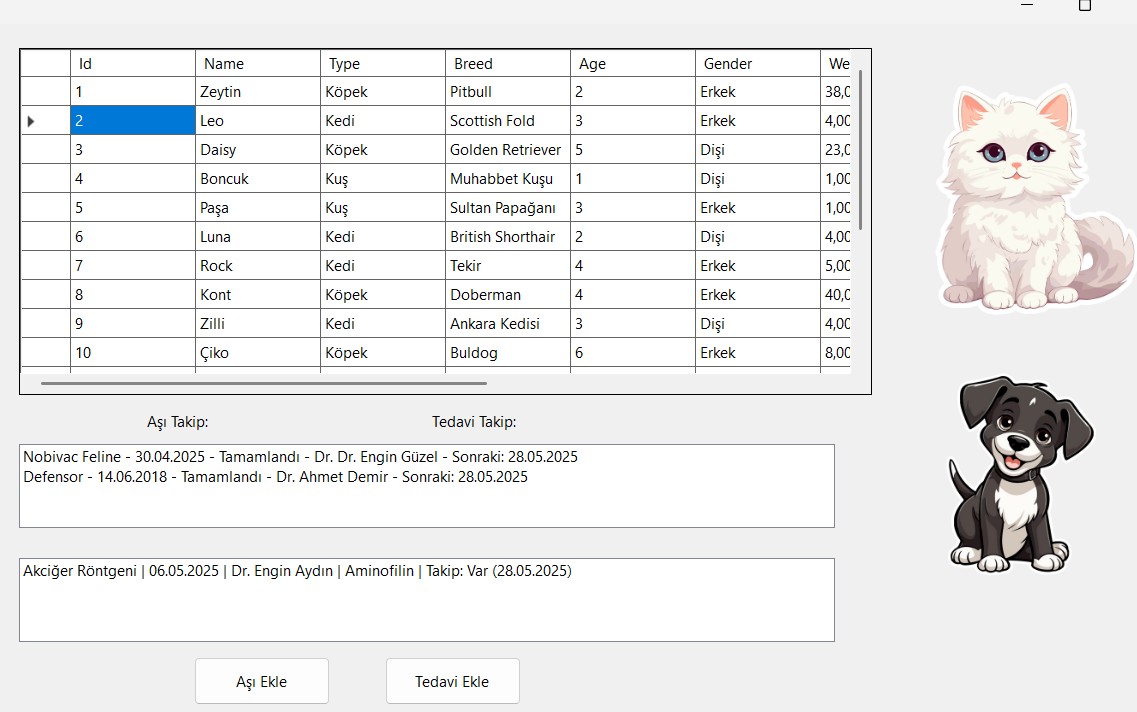
return "Genel bakım ihtiyaçları.";

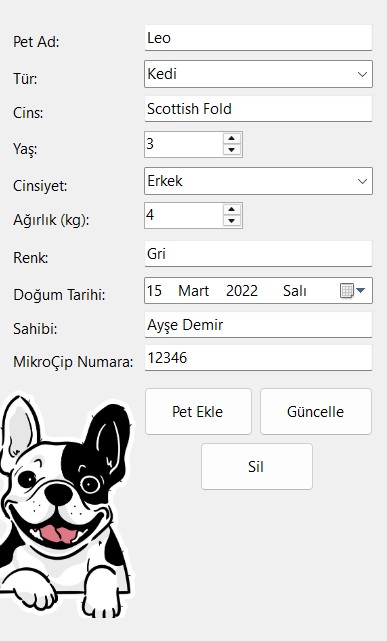
}

**Polimorfizm Sayesinde**

Aynı kod farklı hayvan türleri için çalışabiliyor ve tüm hayvan türleri aynı interface üzerinden kullanılıyor. Çalışma zamanında hangi metodun çağrılacağına karar verilir.





Reusability: Sınıfların tekrar tekrar kullanılabilir olmasına ve Genişletilebilirlik yani yeni hayvan türlerinin kolayca eklenebilir olmasına dikkat edilmiştir. Aşağıda Tedavi ve Aşı ekleme Form Sayfaları gözükmektedir.

