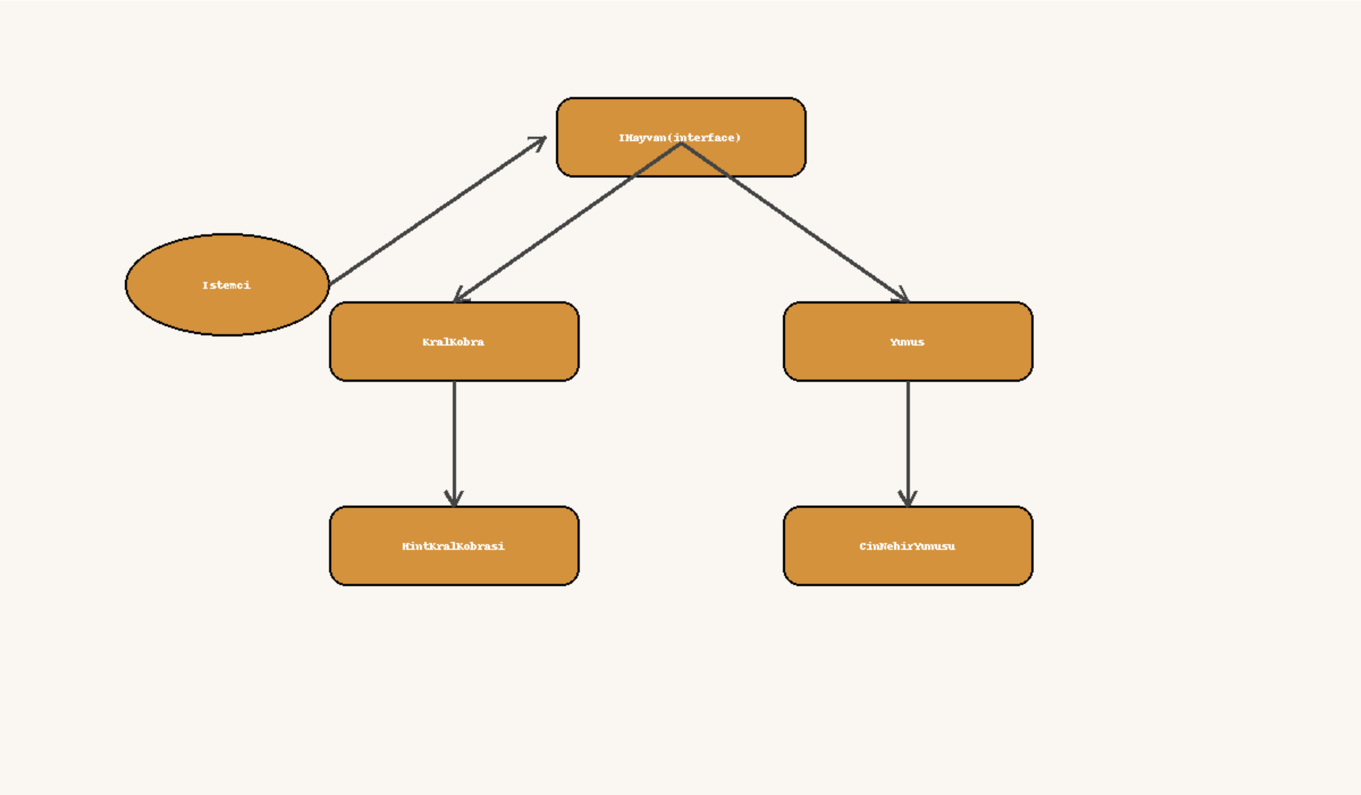
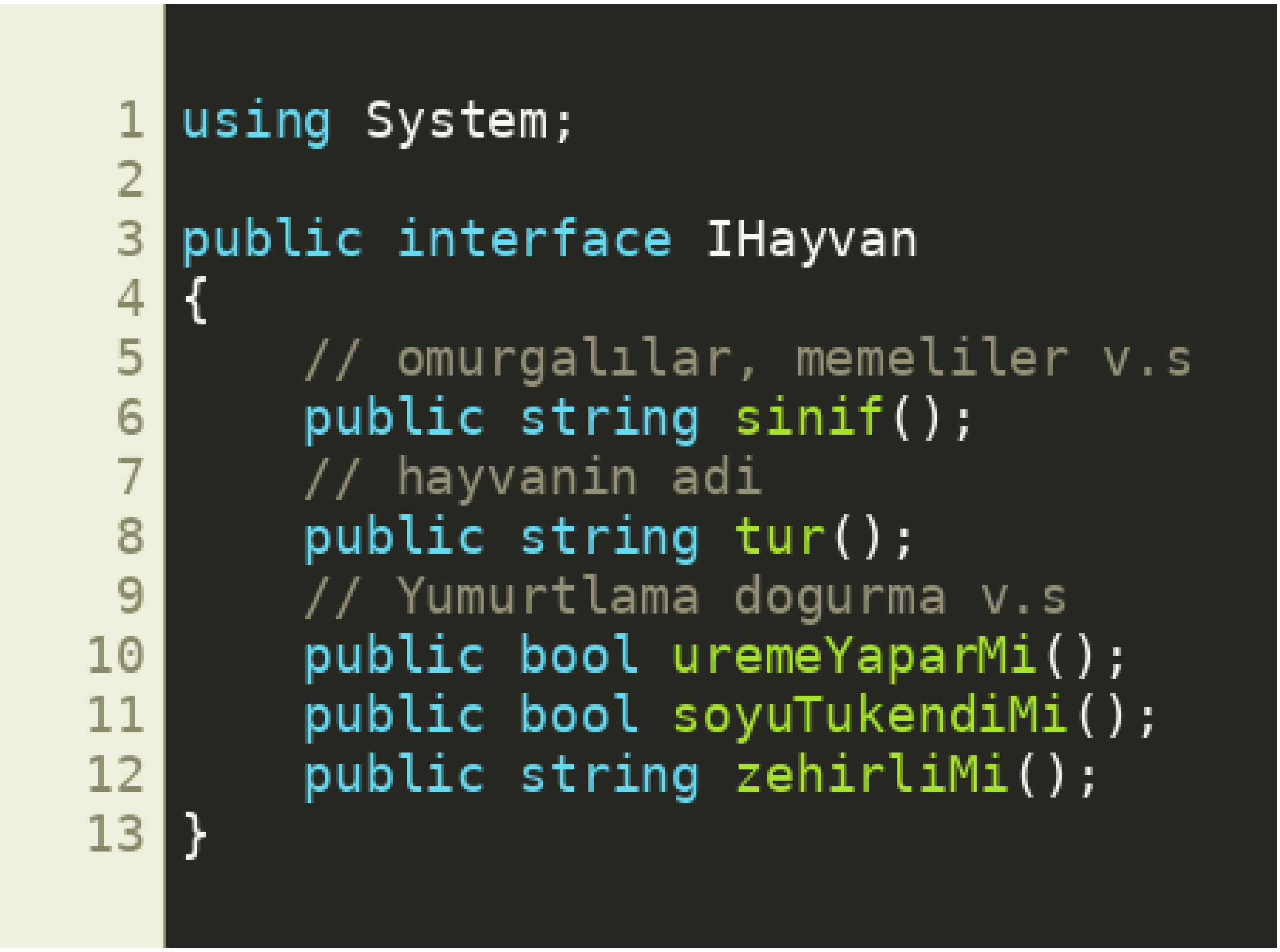
Polymorphism Illustrated - Final Aesthetic

Polymorphism lets different animals respond differently to the same questions. Keywords:
- virtual: parent says 'kids may change this'.
- override: child says 'I change it!'.
Let's explore our animal family!



IHayvan.cs

Interface: common questions every animal must answer.

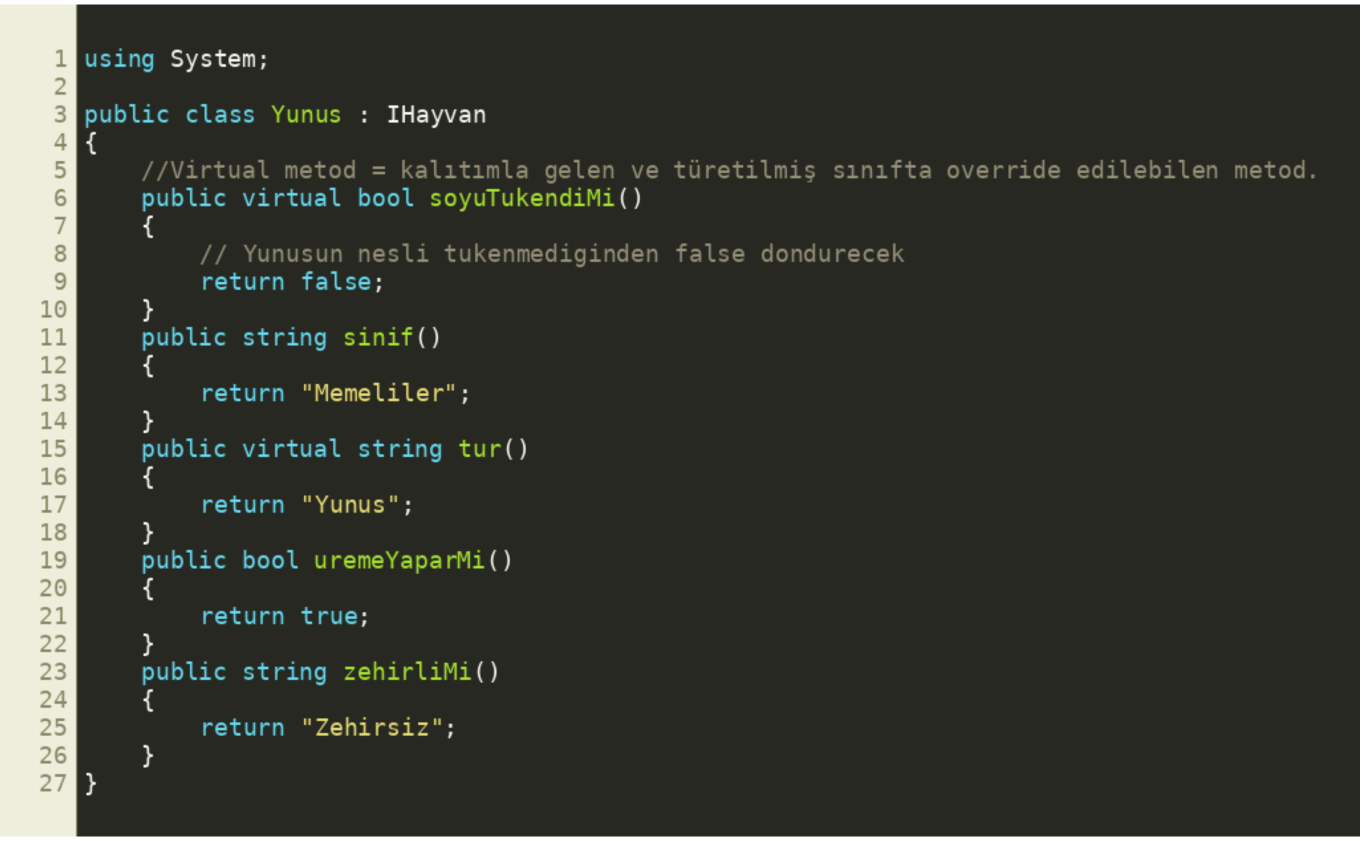


```
using System;

public interface IHayvan
{
    // omurgalılar, memeliler v.s
    public string sinif();
    // hayvanin adi
    public string tur();
    // Yumurtlama dogurma v.s
    public bool uremeYaparMi();
    public bool soyuTukendiMi();
    public string zehirliMi();
}
```

Yunus.cs

Parent dolphin class. Says: virtual answers can be changed by kids.

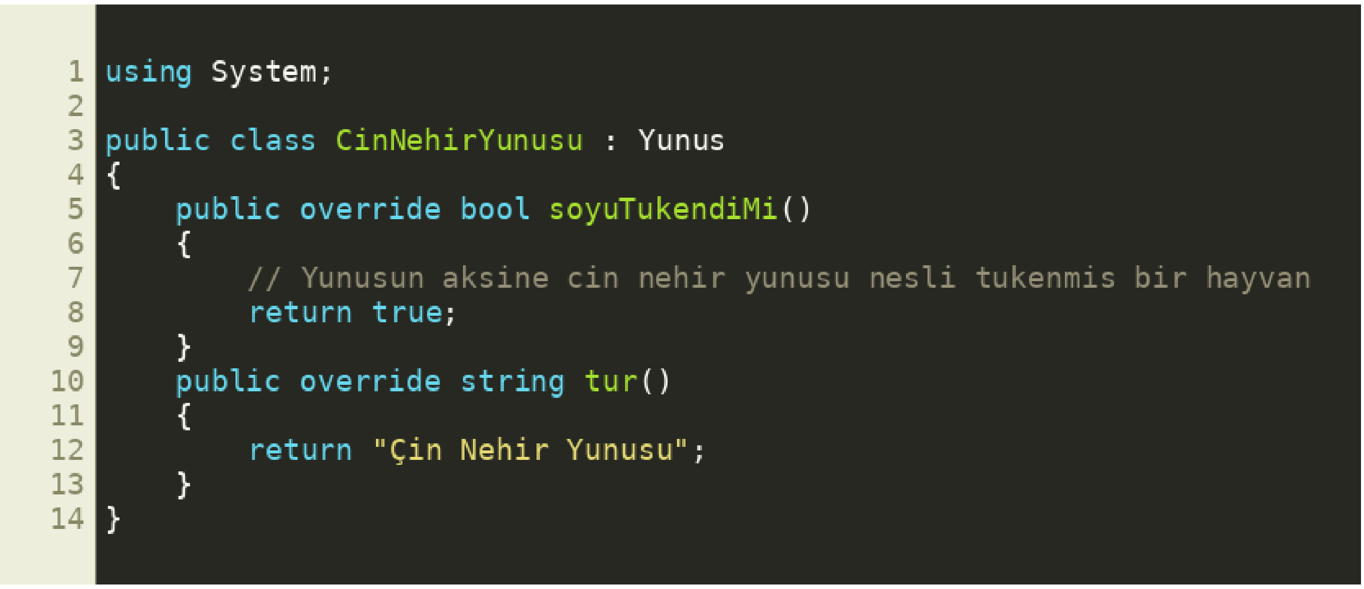


```
using System;

public class Yunus : IHayvan
{
    //Virtual metod = kalıtımla gelen ve türetilmiş sınıfta override edilebilen metod.
    public virtual bool soyuTukendiMi()
    {
        // Yunusun nesli tukenmediginden false dondurecek
        return false;
    }
    public string sinif()
    {
        return "Memeliler";
    }
    public virtual string tur()
    {
        return "Yunus";
    }
    public bool uremeYaparMi()
    {
        return true;
    }
    public string zehirliMi()
    {
        return "Zehirsiz";
    }
}
```

CinNehirYunusu.cs

Child overrides ('override') and changes the answers.

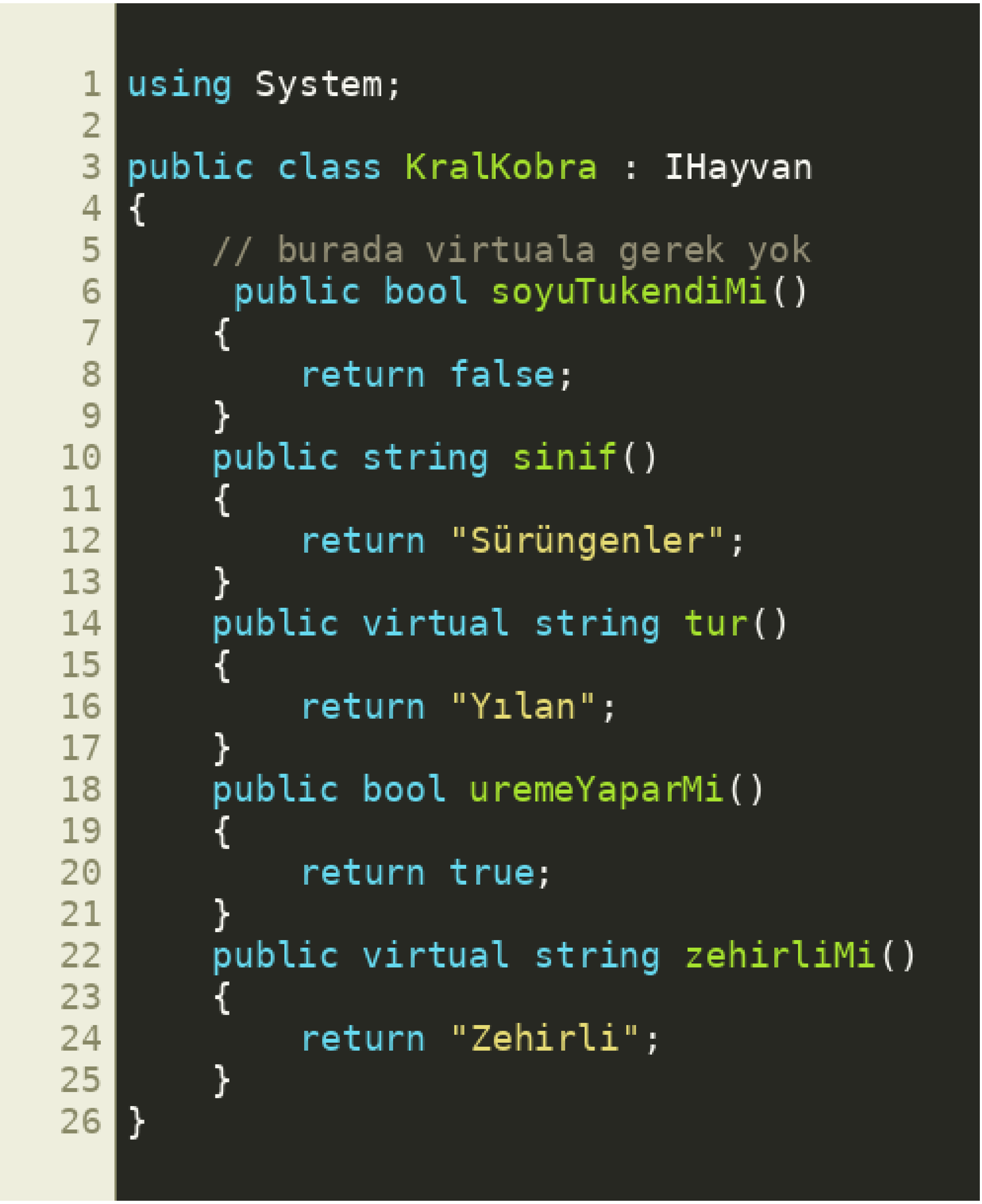


```
using System;

public class CinNehirYunusu : Yunus
{
    public override bool soyuTukendiMi()
    {
        // Yunusun aksine cin nehir yunusu nesli tukenmis bir hayvan
        return true;
    }
    public override string tur()
    {
        return "Çin Nehir Yunusu";
    }
}
```

KralKobra.cs

Base snake. Marks venom answer virtual.

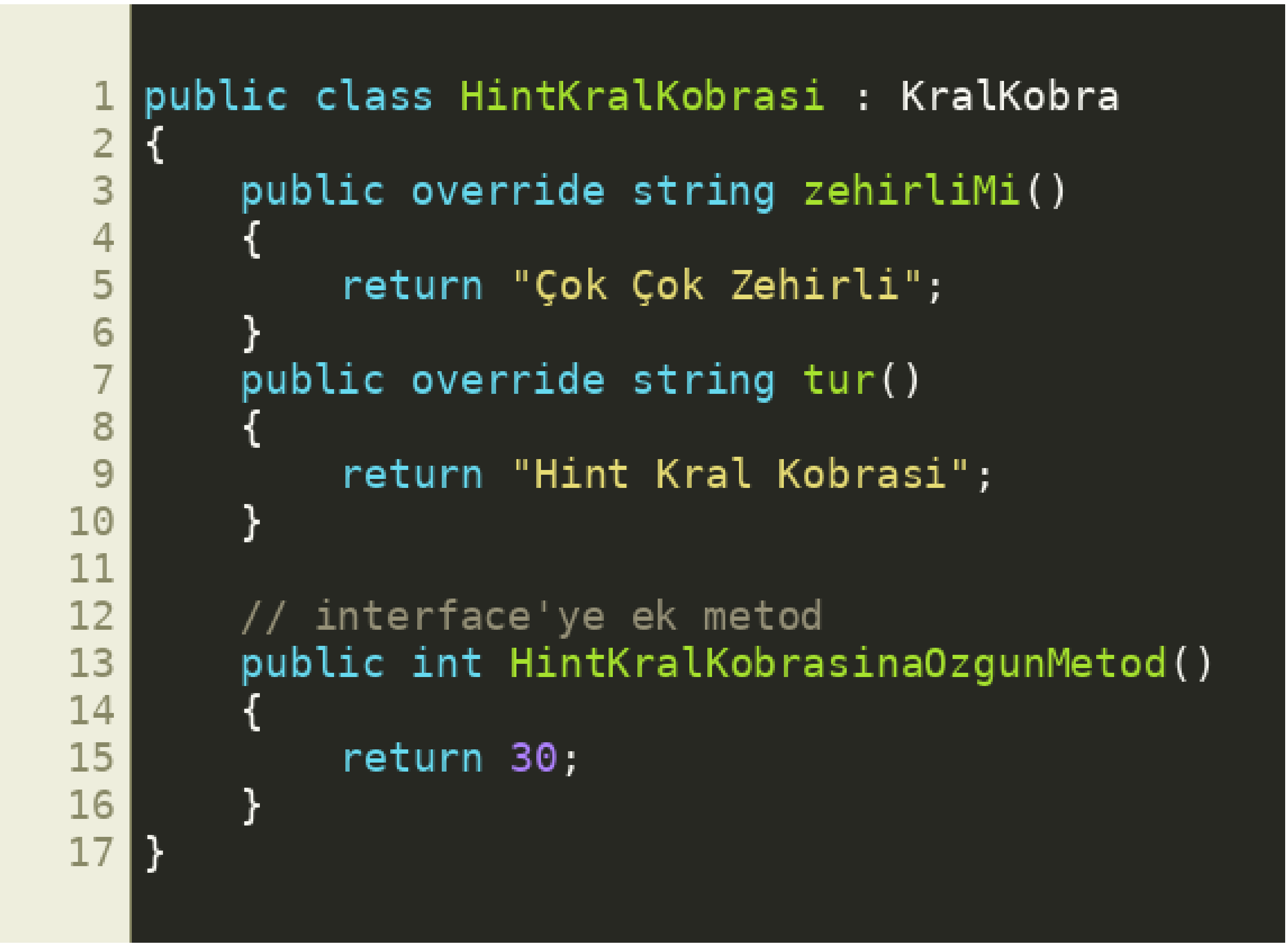


```csharp
using System;

public class KralKobra : IHayvan
{
    // burada virtuala gerek yok
     public bool soyuTukendiMi()
    {
        return false;
    }
    public string sinif()
    {
        return "Sürüngenler";
    }
    public virtual string tur()
    {
        return "Yılan";
    }
    public bool uremeYaparMi()
    {
        return true;
    }
    public virtual string zehirliMi()
    {
        return "Zehirli";
    }
}
```

HintKralKobrasi.cs

Child snake overrides to 'Very very venomous'.

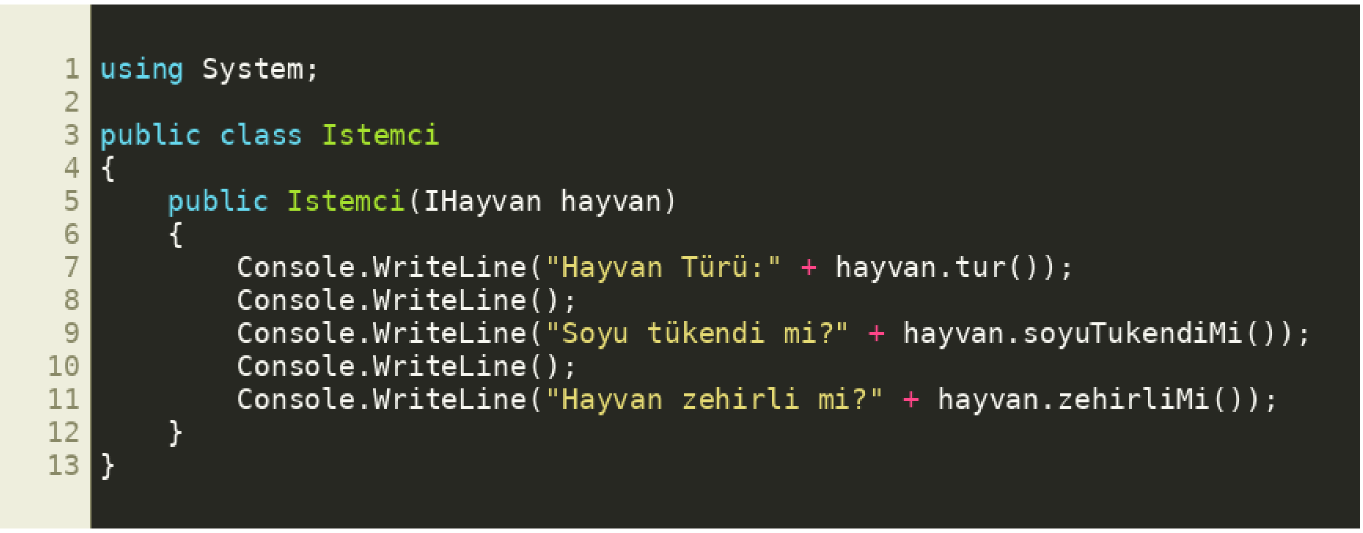


```
public class HintKralKobrasi : KralKobra
{
    public override string zehirliMi()
    {
        return "Çok Çok Zehirli";
    }
    public override string tur()
    {
        return "Hint Kral Kobrasi";
    }

    // interface'ye ek metod
    public int HintKralKobrasinaOzgunMetod()
    {
        return 30;
    }
}
```

Istemci.cs

Client knows only IHayvan, asks questions, prints answers.

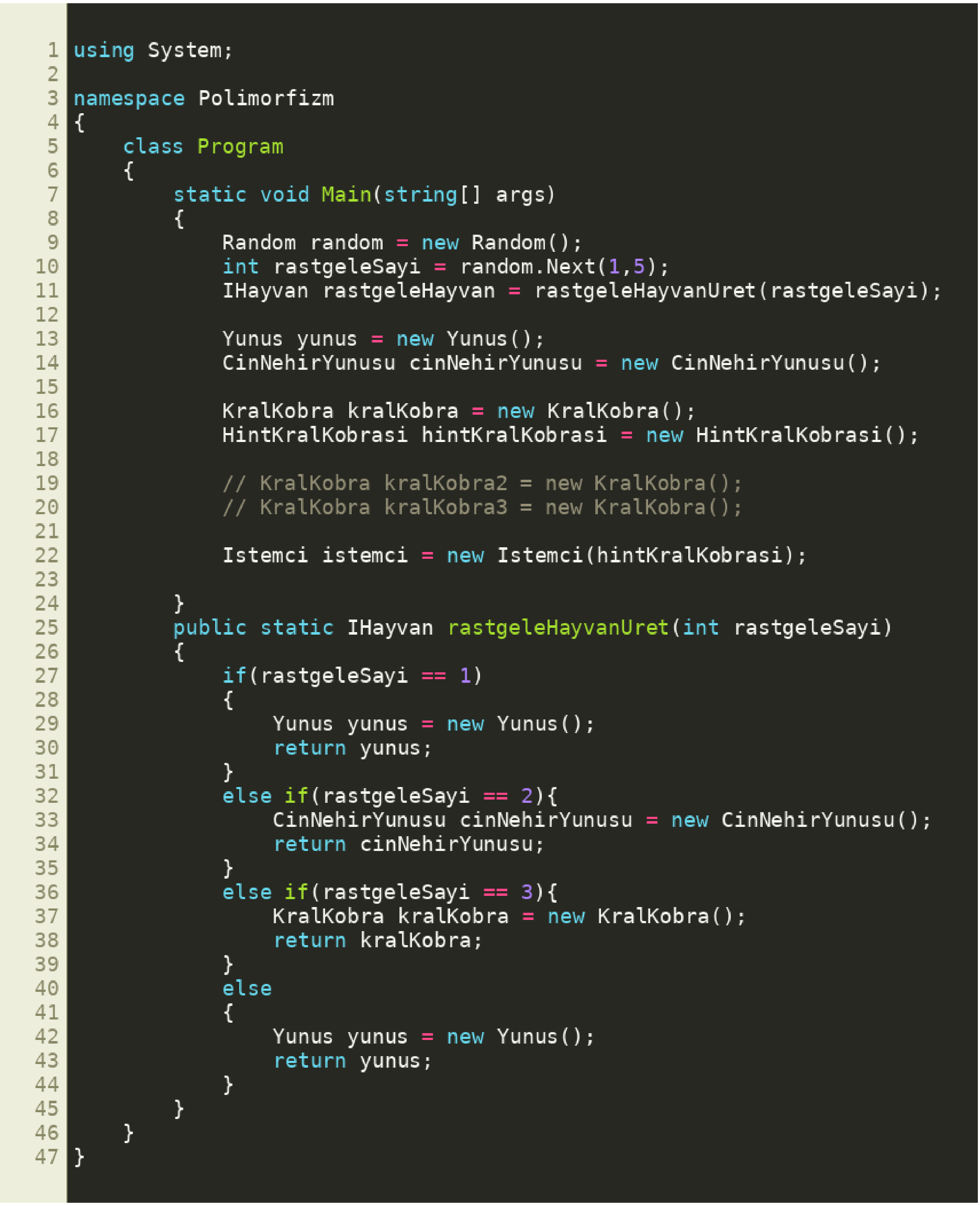


```
using System;

public class Istemci
{
    public Istemci(IHayvan hayvan)
    {
        Console.WriteLine("Hayvan Türü:" + hayvan.tur());
        Console.WriteLine();
        Console.WriteLine("Soyu tükendi mi?" + hayvan.soyuTukendiMi());
        Console.WriteLine();
        Console.WriteLine("Hayvan zehirli mi?" + hayvan.zehirliMi());
    }
}
```

Program.cs

Main program picks random animal and shows polymorphism.



```
using System;

namespace Polimorfizm
{
    class Program
    {
        static void Main(string[] args)
        {
            Random random = new Random();
            int rastgeleSayi = random.Next(1,5);
            IHayvan rastgeleHayvan = rastgeleHayvanUret(rastgeleSayi);

            Yunus yunus = new Yunus();
            CinNehirYunusu cinNehirYunusu = new CinNehirYunusu();

            KralKobra kralKobra = new KralKobra();
            HintKralKobrasi hintKralKobrasi = new HintKralKobrasi();

            // KralKobra kralKobra2 = new KralKobra();
            // KralKobra kralKobra3 = new KralKobra();

            Istemci istemci = new Istemci(hintKralKobrasi);

        }
        public static IHayvan rastgeleHayvanUret(int rastgeleSayi)
        {
            if(rastgeleSayi == 1)
            {
                Yunus yunus = new Yunus();
                return yunus;
            }
            else if(rastgeleSayi == 2){
                CinNehirYunusu cinNehirYunusu = new CinNehirYunusu();
                return cinNehirYunusu;
            }
            else if(rastgeleSayi == 3){
                KralKobra kralKobra = new KralKobra();
                return kralKobra;
            }
            else
            {
                Yunus yunus = new Yunus();
                return yunus;
            }
        }
    }
}
```

Key Takeaways

- Interface == questions list
- virtual == parent gives default but flexible answer
- override == child customises answer
- Client talks only to interface, actual object decides answers at runtime.