

Polymorphism - One Page per Concept

Polymorphism: different objects answer the same questions in their own way. virtual = parent says 'kids may change'. override = child actually changes.

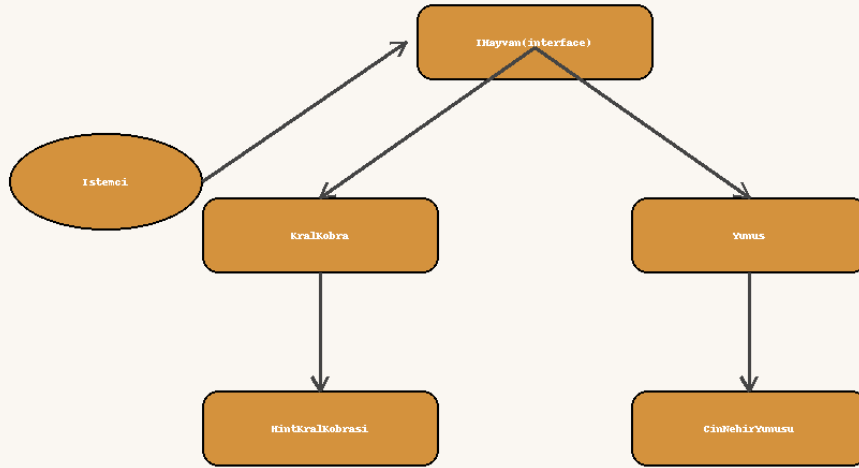


Figure: Family tree of our animal classes and how `Istemci` talks to the interface.

IHayvan.cs

Interface: list of questions every animal must answer.

```
1 using System;
2
3 public interface IHayvan
4 {
5     // omurgalılar, memeliler v.s
6     public string sinif();
7     // hayvanın adı
8     public string tur();
9     // Yumurtlama doğurma v.s
10    public bool uremeYaparMi();
11    public bool soyuTukendiMi();
12    public string zehirliMi();
13 }
```

Yunus.cs

Parent dolphin uses 'virtual' to allow kids to change answers.

```
1 using System;
2
3 public class Yunus : IHayvan
4 {
5     //Virtual metod = kalıtımla gelen ve türetilmiş sınıfta override edilebilen metod.
6     public virtual bool soyuTukendiMi()
7     {
8         // Yunusun nesli tukenmediginden false dondurecek
9         return false;
10    }
11    public string sinif()
12    {
13        return "Memeliler";
14    }
15    public virtual string tur()
16    {
17        return "Yunus";
18    }
19    public bool uremeYaparMi()
20    {
21        return true;
22    }
23    public string zehirliMi()
24    {
25        return "Zehirsiz";
26    }
27 }
```

CinNehirYunusu.cs

Child overrides with its own answers.

```
1 using System;
2
3 public class CinNehirYunusu : Yunus
4 {
5     public override bool soyuTukendiMi()
6     {
7         // Yunusun aksine cin nehir yunusu nesli tukenmis bir hayvan
8         return true;
9     }
10    public override string tur()
11    {
12        return "Çin Nehir Yunusu";
13    }
14 }
```

KralKobra.cs

Parent snake, venom info marked virtual.

```
1 using System;
2
3 public class KralKobra : IHayvan
4 {
5     // burada virtuala gerek yok
6     public bool soyuTukendiMi()
7     {
8         return false;
9     }
10    public string sinif()
11    {
12        return "Sürüngeler";
13    }
14    public virtual string tur()
15    {
16        return "Yılan";
17    }
18    public bool uremeYaparMi()
19    {
20        return true;
21    }
22    public virtual string zehirliMi()
23    {
24        return "Zehirli";
25    }
26 }
```

HintKralKobrası.cs

Child snake overrides venom and species name.

```
1 public class HintKralKobrası : KralKobra
2 {
3     public override string zehirliMi()
4     {
5         return "Çok Çok Zehirli";
6     }
7     public override string tur()
8     {
9         return "Hint Kral Kobrası";
10    }
11
12    // interface'ye ek metod
13    public int HintKralKobrasınaOzgunMetod()
14    {
15        return 30;
16    }
17 }
```

Istemci.cs

Client knows only IHayvan and prints answers.

```
1 using System;
2
3 public class Istemci
4 {
5     public Istemci(IHayvan hayvan)
6     {
7         Console.WriteLine("Hayvan Türü:" + hayvan.tur());
8         Console.WriteLine();
9         Console.WriteLine("Soyu tükendi mi?" + hayvan.soyuTukendiMi());
10        Console.WriteLine();
11        Console.WriteLine("Hayvan zehirli mi?" + hayvan.zehirliMi());
12    }
13 }
```

Program.cs

Main program picks random animal; shows polymorphism.

```
1 using System;
2
3 namespace Polimorfizm
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             Random random = new Random();
10            int rastgeleSayi = random.Next(1,5);
11            IHayvan rastgeleHayvan = rastgeleHayvanUret(rastgeleSayi);
12
13            Yunus yunus = new Yunus();
14            CinNehirYunusu cinNehirYunusu = new CinNehirYunusu();
15
16            KralKobra kralKobra = new KralKobra();
17            HintKralKobrasi hintKralKobrasi = new HintKralKobrasi();
18
19            // KralKobra kralKobra2 = new KralKobra();
20            // KralKobra kralKobra3 = new KralKobra();
21
22            Istemci istemci = new Istemci(hintKralKobrasi);
23
24        }
25        public static IHayvan rastgeleHayvanUret(int rastgeleSayi)
26        {
27            if(rastgeleSayi == 1)
28            {
29                Yunus yunus = new Yunus();
30                return yunus;
31            }
32            else if(rastgeleSayi == 2){
33                CinNehirYunusu cinNehirYunusu = new CinNehirYunusu();
34                return cinNehirYunusu;
35            }
36            else if(rastgeleSayi == 3){
37                KralKobra kralKobra = new KralKobra();
38                return kralKobra;
39            }
40            else
41            {
42                Yunus yunus = new Yunus();
43                return yunus;
44            }
45        }
46    }
47 }
```

Feature Glossary & Key Concepts

Interface

A list of questions every class must answer. It guarantees consistency.

virtual

Parent keyword meaning 'my children are allowed to replace this answer'.

override

Child keyword saying 'I am replacing the parent's answer with my own'.

Polymorphism

Client code talks to the interface; real object decides which override runs at runtime.

Random

Program picks a random animal to prove runtime flexibility.

Open/Closed Principle

System is open for extension (add new animals) but closed for modification (client stays unchanged).

DRY

We avoid repeating printing logic by centralising it in Istemci.

Runtime Binding

CLR chooses the right method implementation after we press Run, not while writing code.