**UĞUR ÖREN NESNE TABANLI PROGRAMLAMA FİNAL SORUSU 2. MADDE**

**Interface nedir?**

**Nesne Arayüzleri, nesneye dayalı programlamada vazgeçilmez olan Class’larımıza rehberlik ederler. Bir Class’ın methodlarını belirtmek için kullanılırlar. Bu methodların içeriğine müdahale etmezler. Programcı istediği allgoritma ile metodların içerisini doldurup işlemini tamamlayabilir.**

**Interface kullanımı ve örneği:**

**Toplama ve çarpma örneği yaptım basit bir şekilde.**

**ilk olarak ınterface ile iki metod tanımladım.**

public interface IIslem

{

void topla(int sayi1, int sayi2);

void carp(int sayi1, int sayi2);

}

**Interface'i tanımladıktan sonra;**

public class Hesap : IIslem

{

public void carp(int sayi1, int sayi2)

{

Console.WriteLine(sayi1 + "\*" + sayi2 + "=" + sayi1 \* sayi2);

}

public void topla(int sayi1, int sayi2)

{

Console.WriteLine(sayi1 + "+" + sayi2 + "=" + (sayi1 + sayi2));

}

}

**hesaplama işlemlerini yapmak için hesap adında bir sınıf oluşturup Interface ile implement(bağlamak,uygulamak) ettim.**

**Son olarak;**

**Bir nesne türettim ve hesaplama işlemini yaptım.**

class Program

{

static void Main(string[] args)

{

Hesap hesap = new Hesap();

hesap.topla(5, 10);

hesap.carp(5, 10);

Console.ReadLine();

}

}

}

**Interface'in temel mantığı ise şu şekildedir**

**günlük hayattan örnek verecek olursak evimizdeki prizler bir ınterface örneğidir.**

**Kodlama kısmına dönecek olursak herhangi bir sınıfta metod parametre olarak başka bir sınıfta olucak ve farklı bir metodu çağıracak yaptığım örnekteki topla gibi ve bunu hangi sınıfta tanımladığımızın bir önemi yok prize takacağımız fişin de bir önemi olmaması gibi.**

**Arayüzü tanımladığın her sınıfta kullanabilirsin bir nevi arayüz aracı gibi bir şey oluyor.**