

# 8. Rozhraní/Interface

- u dědičnosti jsme závislí na třídách
  - příklad Zvíře, Vozidlo, které mají sdílejí funkce
- my ale nechceme být závislí na třídě, chceme být závislí na implementaci (funkci)

př1.

Třídy mohou reprezentovat lidi a my v naší restauraci  
nechceme být závislí na Pepovi jako kucháři. My  
chceme být závislí na někom, kdo umí vařit  
s výhradou.  
A je jedno jestli to je Pepa, Lucha, Libor nebo někdo jiný.

př2.

Obchod nechce být závislý na Monete, KB, Paypalu on  
chce být závislý na transakci resp. na platební metodě.  
Zadáme částku a implementace od něho odečte peníze a  
zašle na účet obchodu.

- jmenná konvence

- interface začíná vždy **I**

pr. **IMovable**

**IFlyable**

- vše co **interface** předepisuje, je vždy public  
- nepředepisuje privátní proměnné!!!

- nemá implementaci

- nemá toto

- zápis

**public interface IVehicle {**

**String Brand { get; };** - pokud je pouze get  
hodnota je pouze READ ONLY

**String Color { get; set; };** - hodnotu lze i měnit mimo třídu

**void Drive(int kilometers);**

**}**

Počet odkazů: 0

```
public interface IVehicle
```

```
{
```

Počet odkazů: 0

```
    string Brand { get; } // jedná se o READONLY vlastnost
```

Počet odkazů: 0

```
    string Color { get; set; } // vlastnost, kterou lze i měnit (má setter)
```

Počet odkazů: 0

```
    void Drive(int kilometers); // metoda
```

```
}
```

# Použití

- stejně jako u dědičnosti, jinak se vyskytuje
- auto **implementuje** **IVehicle**

```
internal class Car : IVehicle
{
}
```

interface UKOL\_Sprava\_vozoveho\_parku.IVehicle

CS0535: Car neimplementuje člen rozhraní IVehicle.Brand.

CS0535: Car neimplementuje člen rozhraní IVehicle.Color.

CS0535: Car neimplementuje člen rozhraní IVehicle.Drive(int).

- IDE nám samo řekne co nám chybí **doimplementovat**

Počet odkazů: 4

```
internal class Car : IVehicle
{
}
```

Implementujte rozhraní.  
Implementovat všechny členy explicitně

CS0535 Car neimplementuje člen rozhraní IVehicle.Brand.

Rádky 10 až 11

```
{
+ public string Brand => throw new NotImplementedException();
+ public string Color { get => throw new NotImplementedException(); set => throw new NotImplementedException();
+ public void Drive(int kilometers)
+ {
+     throw new NotImplementedException();
+ }
```

Zobrazit náhled změn  
Opravit všechny výskyty v: Dokument | Projekt | Řešení | Obsahující typ

- pomocí šroubováku nebo zárodky "implementovat rozhraní"

```
internal class Car : IVehicle
{
    Počet odkazů: 1
    public string Brand => throw new NotImplementedException();

    Počet odkazů: 1
    public string Color { get => throw new NotImplementedException(); set => throw new NotImplementedException(); }

    Počet odkazů: 2
    public void Drive(int kilometers)
    {
        throw new NotImplementedException();
    }
}
```

- následně upravíme a máme hotovo

```
internal class Car : IVehicle
{
    Počet odkazů: 2
    public string Brand { get; }

    Počet odkazů: 2
    public string Color { get; set; }

    Počet odkazů: 2
    public Car(string brand, string color)
    {
        this.Brand = brand;
        this.Color = color;
    }

    Počet odkazů: 2
    public void Drive(int kilometers)
    {
        Console.WriteLine($"Drive with steering wheel {kilometers}km.");
    }
}
```

```
internal class Motorcycle : IVehicle
{
    Počet odkazů: 2
    public string Brand { get; }

    Počet odkazů: 2
    public string Color { get; set; }

    Počet odkazů: 2
    public Motorcycle(string brand, string color)
    {
        this.Brand = brand;
        this.Color = color;
    }

    Počet odkazů: 2
    public void Drive(int kilometers)
    {
        Console.WriteLine($"Drive with handlebars {kilometers}km.");
    }
}
```

- my nechceme být závislí na autě nebo motorce ale  
na něčem, co umí ujet počet kilometrů v domě  
příkladě na vozíku, respektive *IVehicle*. Je jedno jak, ale  
odveze nás.