



Национална програма  
"Обучение за ИТ умения и кариера"  
<https://it-kariera.mon.bg>

Министерството на  
образованието и науката  
<https://www.mon.bg>



# Презареждане и презаписване на методи

Overloading & Overriding

# Съдържание

1. Презареждане
2. Презаписване

# Презареждане на методи

- Презареждане (overloading) – методи с едно и също име, но различни сигнатури.
- При компилиране, според подадените параметри компилатора определя кой метод точно ще изпълни

# 3agaya: MathOperation

## MathOperation

```
+Add(int, int): int  
+Add(double, double, double): double  
+Add(decimal, decimal, decimal): decimal
```

```
MathOperations mo = new MathOperations();  
Console.WriteLine(mo.Add(2, 3));  
Console.WriteLine(mo.Add(2.2, 3.3, 5.5));  
Console.WriteLine(mo.Add(2.2m, 3.3m, 4.4m));
```

# Peweehue: MathOperation

```
public int Add(int a, int b)
{
    return a + b;
}
public double Add(double a, double b, double c)
{
    return a + b + c;
}
public decimal Add(decimal a, decimal b, decimal c)
{
    return a + b + c;
}
```

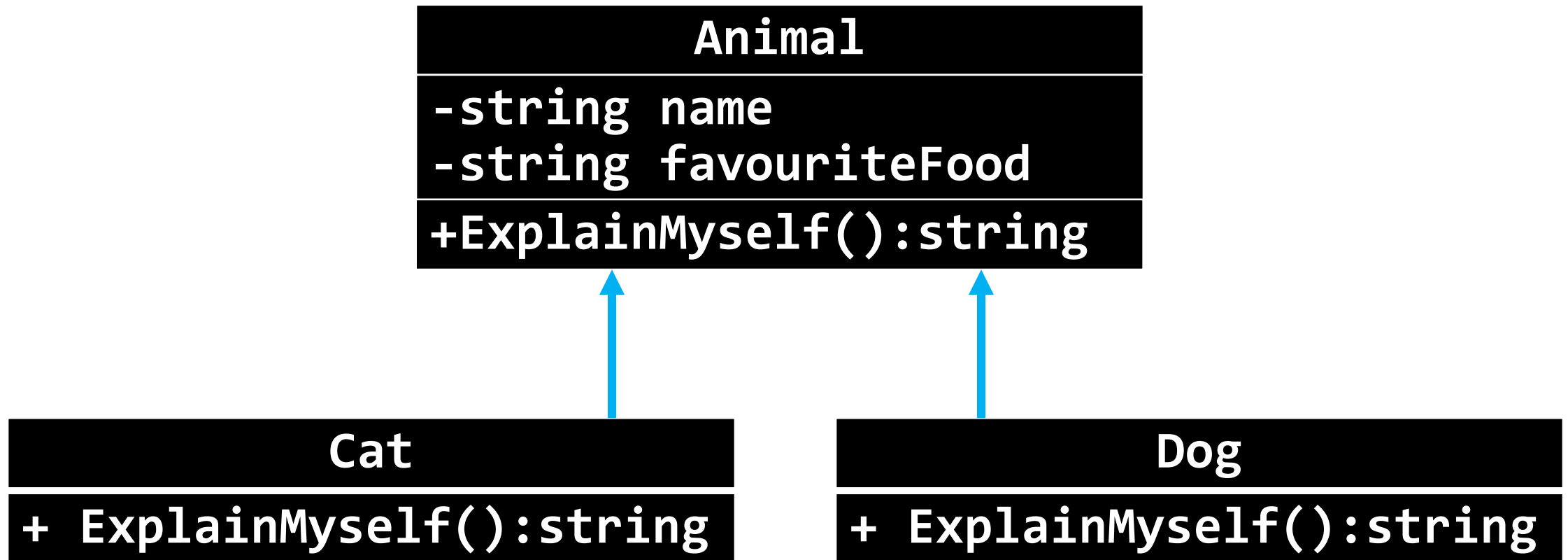
# Правила за презареждане на методи

- Презареждането може да се случи в същия клас или в него подклас.
- Конструкторите могат да бъдат презаредени
- Презаредените методи трябва да имат различни един от друг параметри.
- Презаредените методи винаги трябва да бъдат част от същия клас (подклас), с едно и също име, но с различни параметри.
- Могат да имат едни и същи или различни типове на връщана стойност.

# Презаписване на методи

- Презаписване на метод (**overriding**) – създаване на метод със същото име и сигнатура в подклас
- В C#, за да позволим даден метод да бъде презаписан поставяме ключовата дума **virtual** пред него.
- За да окажем, че даден метод в подкласа, презаписва метод от базовия клас, поставяме ключовата дума **overrides**

# Задача: Животни





# Решение: Животни [1/2]

```
public class Animal
{
    public string Name { get; protected set; }
    public string FavouriteFood { get; protected set; }
    public virtual string ExplainMyself()
    {
        return $"I am {this.Name} and my favourite food is
            {this.FavouriteFood}";
    }
}
```

# Решение: Животни [1/2]

```
public class Dog : Animal
{
    public Dog(string name, string favouriteFood)
    {
        this.Name = name;
        this.FavouriteFood = favouriteFood;
    }
    public override string ExplainMyself()
    {
        return base.ExplainMyself() + Environment.NewLine +
            "DJAAF";
    }
}
```

# Правила за презаписване на методи

- Презаписването може да се случи в **подкласовете**.
- **Параметрите** трябва да са **същите** като тези на **родителския метод**
- Презаписващият метод трябва да има **същия тип на връщана стойност**
- **Модификатора за достъп** не може да бъде **по-ограничаващ**
- Методи дефинирани като **private u static НЕ** могат да бъдат презаписани
- Презаписващият метод **не трябва** да хвърля нови или по-широко **проверяеми изключения**.

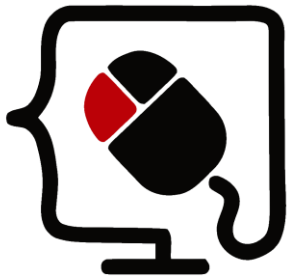


# Презареждане и презаписване

Лаб

# Какво научихме днес?

- Какво е полиморфизъм?
- Видове полиморфизъм
  - Статичен (по време на компилиране)
  - Динамичен (по време на изпълнение)
- Презареждане на методи
- Презаписване на методи



Национална програма  
"Обучение за ИТ умения и кариера"  
<https://it-kariera.mon.bg>

Министерството на  
образованието и науката  
<https://www.mon.bg>



Документът е разработен за нуждите на Национална програма "Обучение за ИТ умения и кариера" на Министерството на образованието и науката (МОН) и се разпространява под свободен лиценз CC-BY-NC-SA (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).