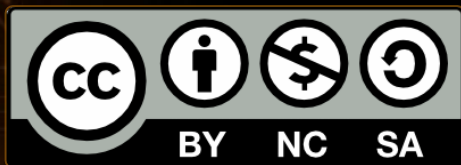


Полиморфизъм



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>

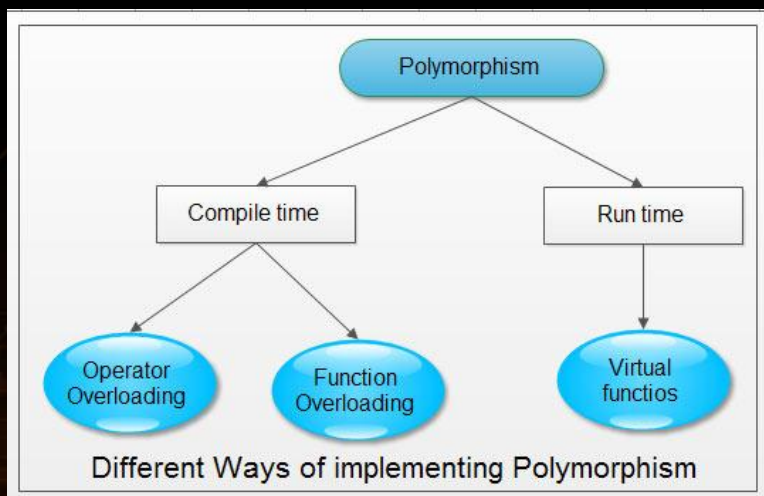


Съдържание

1. Какво е полиморфизъм?

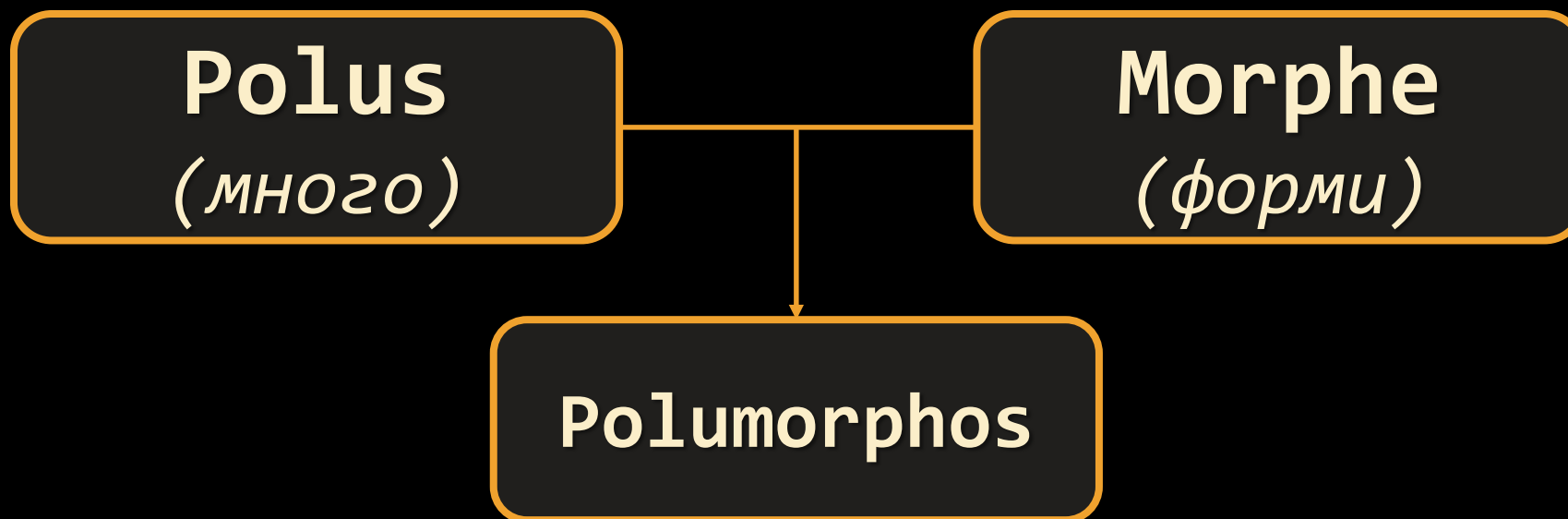
2. Видове полиморфизъм

- предефиниране (overloading)
- пренаписване (overriding)



Какво е полиморфизъм?

- От гръцки:



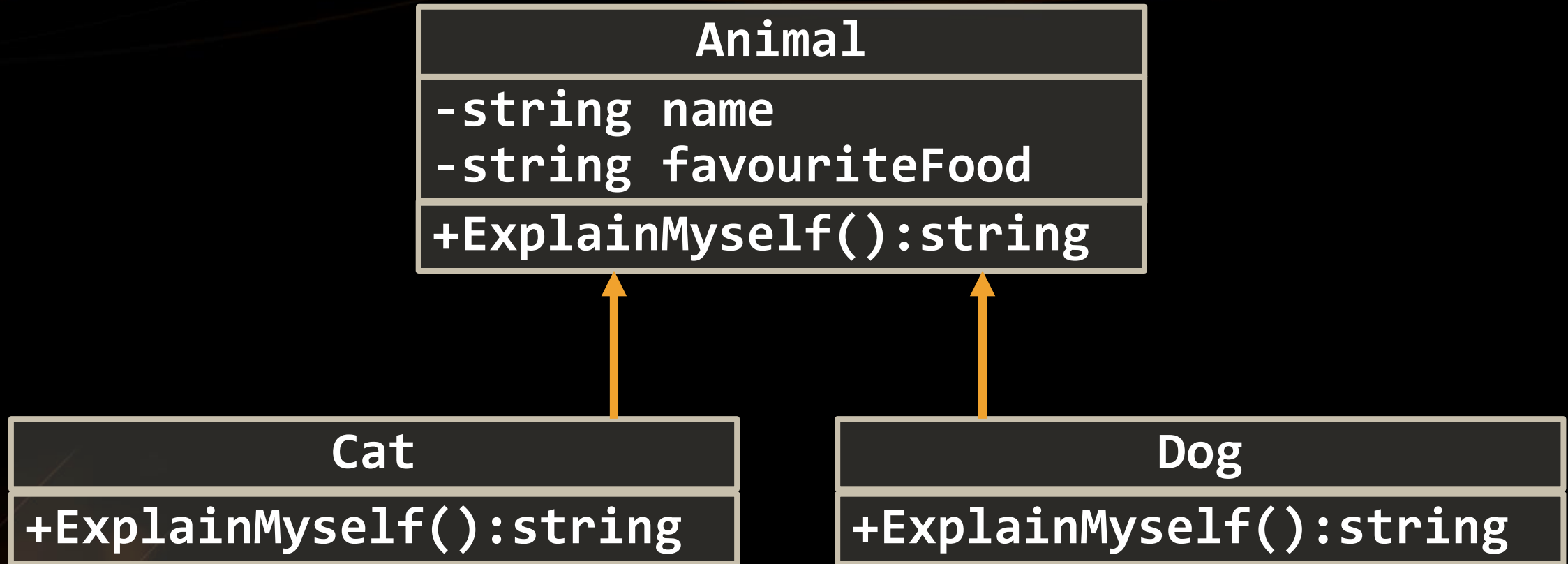
В ООП:

- Възможността под **едно** име да извикаме **различни** методи

Типове полиморфизъм

- Полиморфизъм по време на изпълнение – постига се чрез пренаписване (overriding) на метод – създаване на метод със същото име и сигнатура в подклас
- Полиморфизъм по време на компилиране – постига се чрез предефиниране (overloading) на методи с едно и също име, но с различни сигнатури.

Задача: Животни



Решение: Животни

```
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}";
    }
}
```

Метод, който ще бъде пренаписван

Решение: Животни (2)

```
public class Dog : Animal
{
    public override string ExplainMyself() {
        return base.ExplainMyself() + " DJAAF";
    }
}
public class Cat : Animal
{
    public override string ExplainMyself() {
        return base.ExplainMyself() + " MEOWW";
    }
}
```

Пренаписване на
метод (overriding)

Решение: Животни (3)

```
public static void main(String[] args) {  
    Animal myPet = new Cat("Pesho", "Whiskas");  
    Animal yourPet = new Dog("Gosho", "Meat");  
  
    Console.WriteLine(myPet.ExplainMyself()); // DJAFF  
    Console.WriteLine(yourPet.ExplainMyself()); // MEOWW  
}
```

Динамичен полиморфизъм
(по време на изпълнение)

Правила за пренаписване на методи

- Пренаписването се случва в подкласовете
- Параметрите трябва да са същите като на родителския метод
- Двата метода трябва да имат същия тип на връщана стойност
- Модификатора за достъп не може да бъде по-ограничаващ
- `private` и `static` методи НЕ могат да бъдат пренаписани

Типове полиморфизъм

- Полиморфизъм по време на компилиране – постига се чрез предефиниране (overloading) – методи с едно и също име, но с различни сигнатури.
- Известен и като статичен полиморфизъм
- При компилиране, според подадените параметри компилатора

```
public static void main(String[] args) {  
    int a = Sum(10, 20, 30);  
    double b = Sum(0.5, 0.01);  
}
```

определя кой
метод ще изпълни

статичен полиморфизъм
(по време на компилиране)

Задача: 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));
```

Решение: 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 (предефиниране) на методи

- Презареждането може да се случи в същия клас или в негов подклас
- Конструкторите могат да бъдат презаредени
- Презаредените методи трябва да имат различни един от друг параметри
- Могат да имат едни и същи или различен тип връщана стойност

Какво научихме днес?

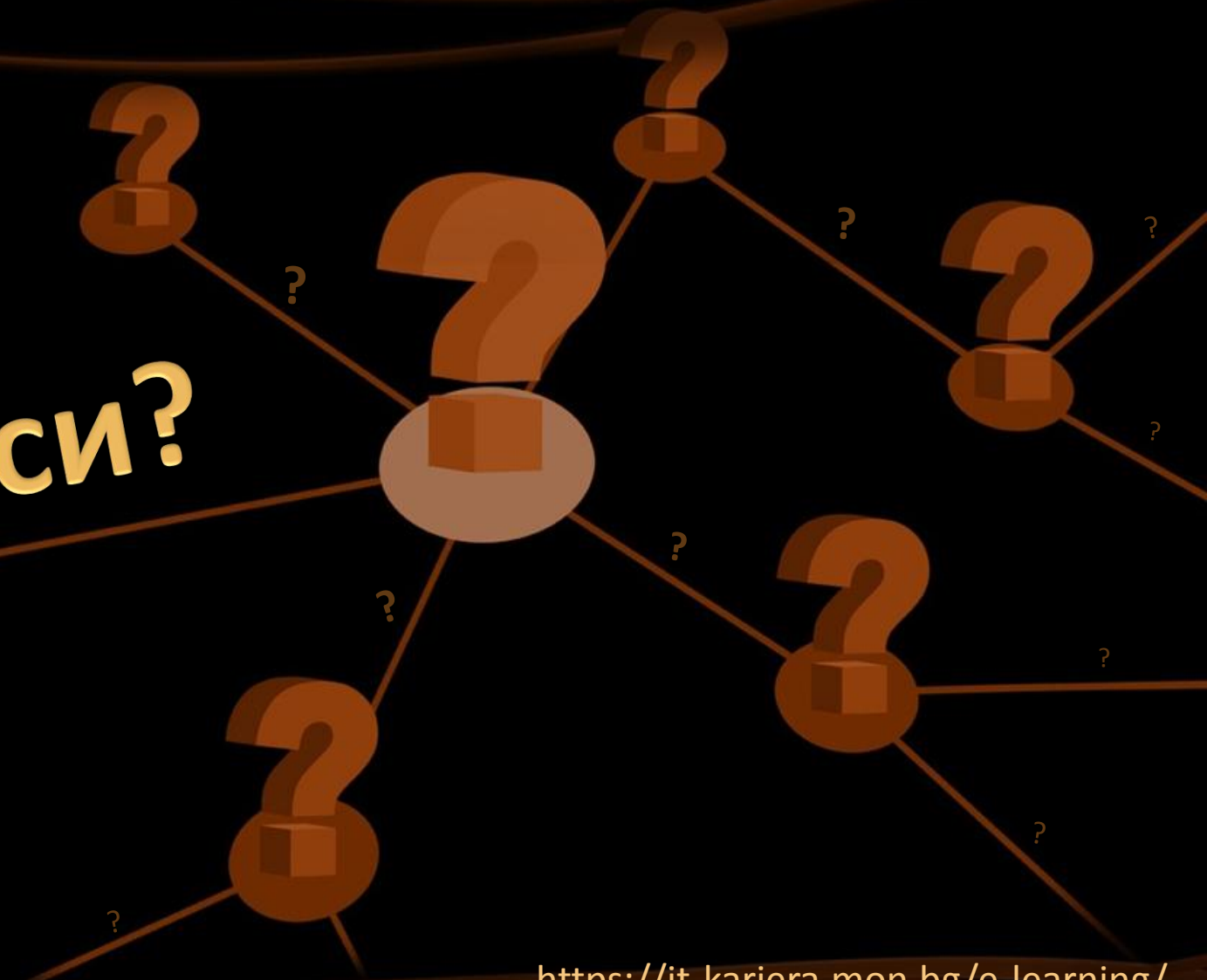
- Какво е полиморфизъм?
- Видове полиморфизъм
 - Статичен (по време на компилиране) – чрез предефиниране на методи
 - Динамичен (по време на изпълнение) – чрез пренаписване на методи



Полиморфизъм



Въпроси?



Лиценз

- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"



- Благодарности: настоящият материал може да съдържа части от следните източници
 - Книга "Основи на програмирането със C#" от Светлин Наков и колектив с лиценз CC-BY-SA