Полиморфизъм

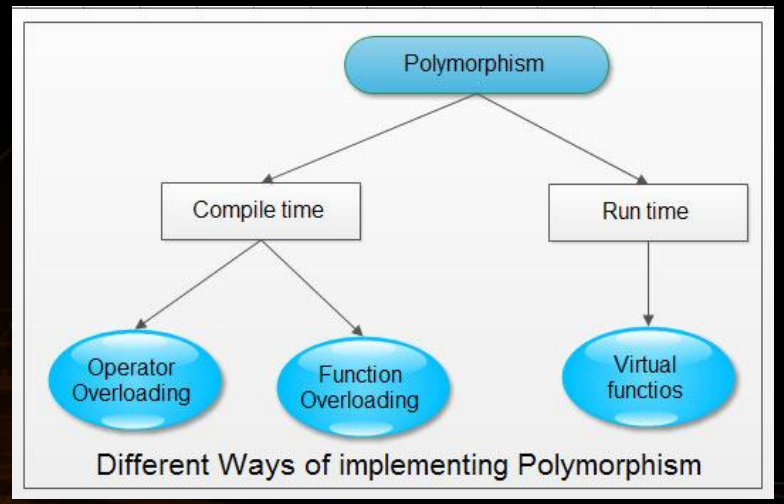
Учителски екип

Обучение за ИТ кариера

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

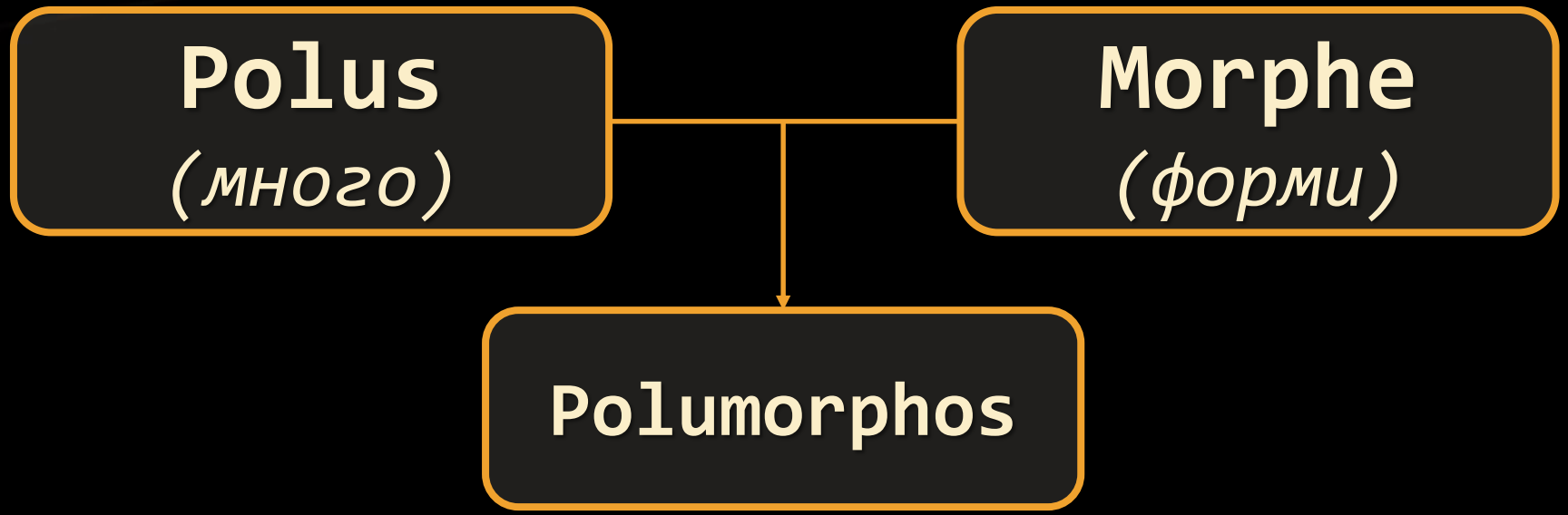
Съдържание

1. Какво е полиморфизъм?
2. Видове полиморфизъм
   - предефиниране (overloading)
   - пренаписване (overriding)



Какво е полиморфизъм?

- От гръцки



≈ дума с няколко различни значения, в зависимост от контекста

Полиморфизъм в ООП

- Възможността под едно име да има различни форми (или програмни реализации)

```
public class Animal {}
public class Mammal {} : Animal {}
public class Person : Mammal {}
```

```
Person IS-A Person
```

```
Person IS-A Animal
```

```
Person IS-A Mammal
```

```
Person IS-A Object
```

Референтен тип и обектен тип

```
public class Mammal : Animal {}
public class Person : Mammal {}
Animal animal = new Person();
Mammal mammal = new Person();
Person person = new Person();
```

Референтен тип

Обектен тип

- Променливите са запазени в референтен тип
- Може да използвате само`членовете на референтния клас
- Ако имате нужда от достъп до елемент на обектния клас, трябва да преобразувате референтния тип или да пренапишете метода

Ключова дума - is

- Проверява дали обекта е инстанция на специфичен клас

```
public class Person : Mammal, Animal {}
Animal animal    = new Person();
Mammal mammal    = new Person();
Person person    = new Person();
if (animal is Person)
{
  ((Person) animal).Eat();
}
```

Проверка на обектния тип на person

Преобразуване на обектния тип и използване на методите му

Ключова дума - is (2)

Всеки път, когато усетиш, че пишеш код от типа „ако обекта е от тип T1, то направи нещо, но ако е от тип T2, то направи друго нещо“, си забий шамар.

От *Effective C++*, автор: Scott Meyers

Типове полиморфизъм

- Полиморфизъм по време на изпълнение – постига се чрез пренаписване (overriding) на метод – създаване на метод със същото име и сигнатура в подклас
- В C#, за да позволим даден метод да бъде пренаписан поставяме ключовата дума virtual пред него.
- За да укажем, че даден метод в подкласа пренаписва метод от базовия клас, поставяме ключовата дума overrides

Типове полиморфизъм

- Полиморфизъм по време на изпълнение

```
public class Shape {
  public virtual void Draw() {}
}
public class Circle : Shape {
  public override void Draw() {
    // рисуване на кръг
  }
}
public static void main(String[] args) {
  Shape shape = new Circle();
  shape.Draw();
}
```

Метод, който може да бъде пренаписан

Пренаписване на метод (overriding)

Полиморфизъм по време на изпълнението

- Използва се презаписващ метод

```
public static void Main()
{
  Rectangle rect = new Rectangle(3.0, 4.0);
  Rectangle square = new Square(4.0);

  Console.WriteLine(rect.Area());
  Console.WriteLine(square.Area());
}
```

Презапис на метод

Полиморфизъм по време на изпълнението (2)

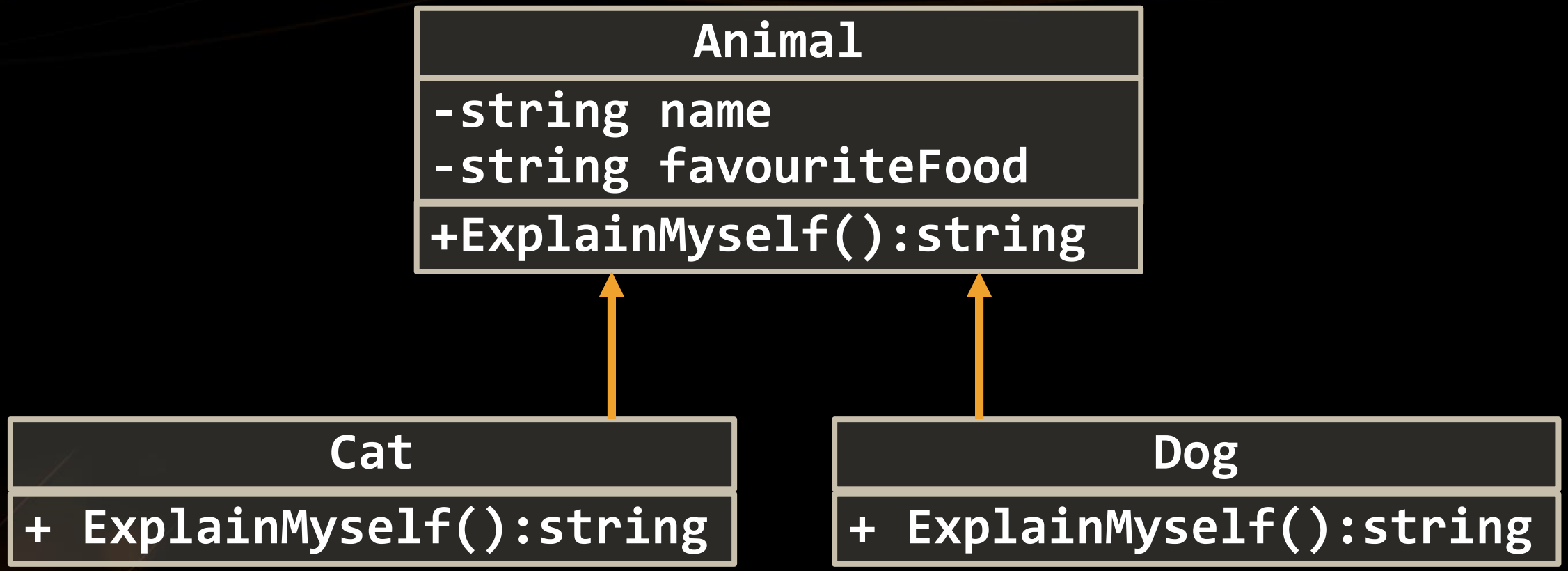
- Също известен като динамичен полиморфизъм

```
public class Rectangle {
  public virtual double Area() {
    return this.a * this.b;
  }
}

public class Square : Rectangle {
  public override double Area() {
    return this.a * this.a;
  }
}
```

Презаписване на метод

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 fovourite food is
                                {this.FavouriteFood}";
  }
}
```

Решение: Животни (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 пренаписване 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 връщаната стойност
- Модификатора за достъп не може да бъде по-ограничаващ
- Методи дефинирани като private и static 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.
- При компилиране, според подадените параметри компилатора определя кой метод точно ще изпълни

```
public static void main(String[] args) {
  int Sum(int a, int b, int c)
  double Sum(Double a, Double b)
}
```

Предефиниране на метод (overloading)

Полиморфизъм по време на компилиране

- Известен и като статичен полиморфизъм

```
public static void Main() {
  static int MyMethod(int a, int b) {}
  static double MyMethod(double a, double b) {}
}
```

Предефиниране на метод

- Метод с едно и също име може да се различава по:
  - Брой параметри
  - Тип на параметрите
  - Поредицата от типовете на параметрите

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
- 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 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