

Полиморфизъм

Много форми на обектите и
извикване през базов интерфейс



Учителски екип

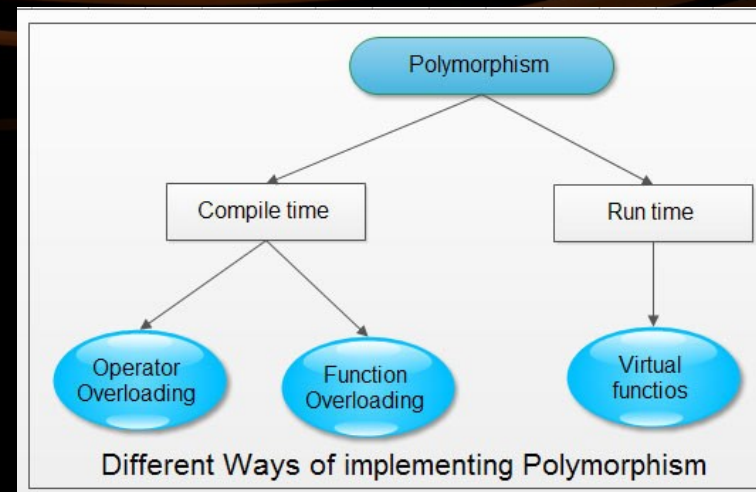
Обучение за ИТ кариера

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



Съдържание

1. Какво е полиморфизъм?
2. Видове полиморфизъм

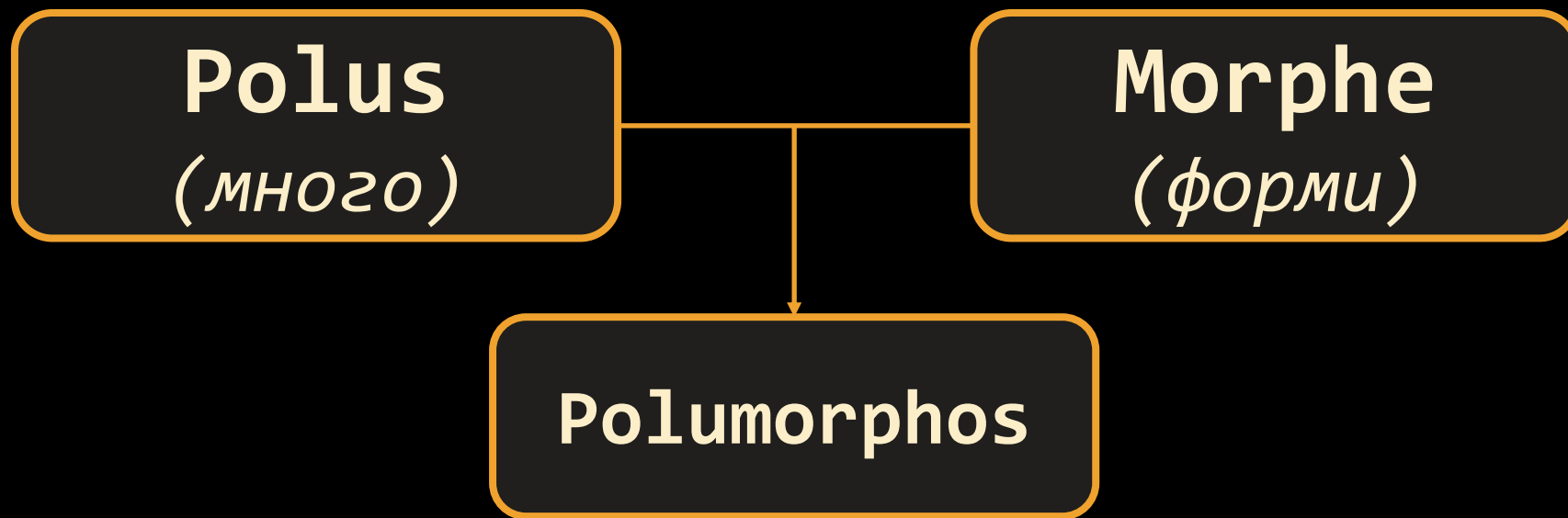




Полиморфизъм

Какво е полиморфизъм?

- От гръцки



Това е нещо подобно на дума с **няколко различни значения, в зависимост от контекста**

Полиморфизъм в ООП

- Възможността на **обект** да заема **много форми**

```
public interface IAnimal {}  
public abstract class Mammal {}  
public class Person : Mammal, IAnimal {}
```

Person **IS-A** Person

Person **IS-A** Animal

Person **IS-A** Mammal

Person **IS-A** Object

Референтен тип и обектен тип

```
public class Person : Mammal, Animal {}  
Animal person = new Person();  
Mammal personOne = new Person();  
Person personTwo = new Person();
```

Референтен тип

Обектен тип

- Променливите са запазени в референтен тип
- Може да използвате само референтни методи
- Ако имате нужда от обектен метод, трябва да го преобразувате или да го презапишете

Ключова дума – is

- Проверява дали **обекта** е **инстанция** на специфичен **клас**

```
public class Person : Mammal, Animal {}  
Animal person = new Person();  
Mammal personOne = new Person();  
Person personTwo = new Person();  
if (person is Person)  
{  
    ((Person) person).getSalary();  
}
```

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

```
public class Shape {}  
public class Circle : Shape {}  
public static void main(String[] args) {  
    Shape shape = new Circle()  
}
```

Презаписване
на метод

- Полиморфизъм по време на компилиране

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

Презареждане
на метод

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) {}  
}
```

Презареждане на
метод

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

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 double Area() {  
        return this.a * this.b;  
    }  
}
```

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

Презаписване на
метод

Полиморфизъм



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

