

Наследяване

Йерархии от класове



Учителски екип

Обучение за ИТ кариера

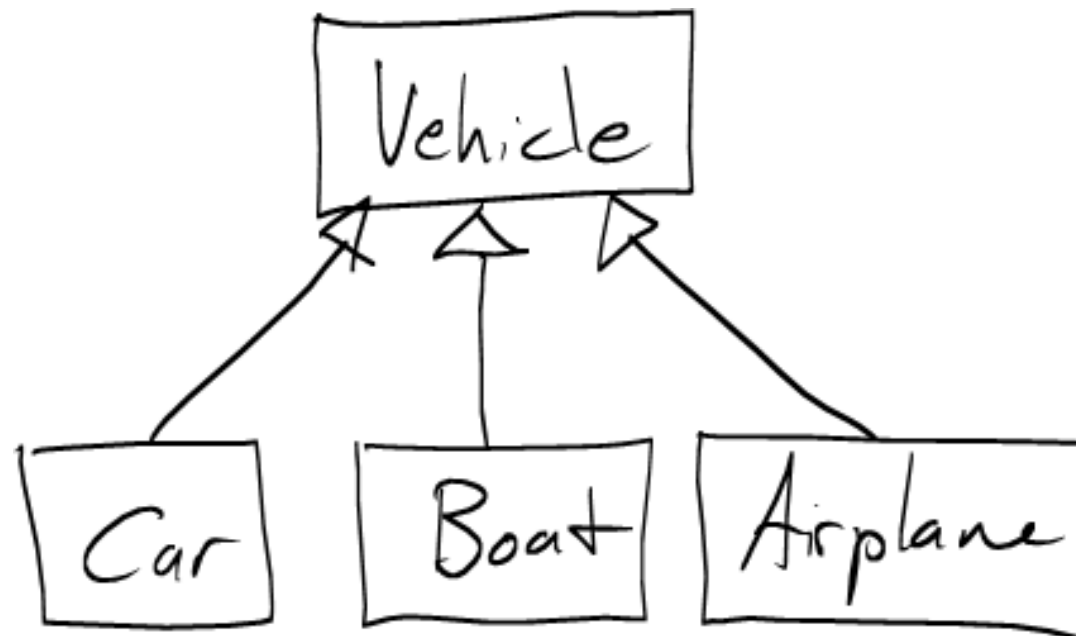
<https://it-kariera.mon.bg/e-learning/>



Съдържание

1. Наследяване
2. Йерархии от класове
3. Наследяване в C#
4. Достъп до членове на базовия клас





Наследяване

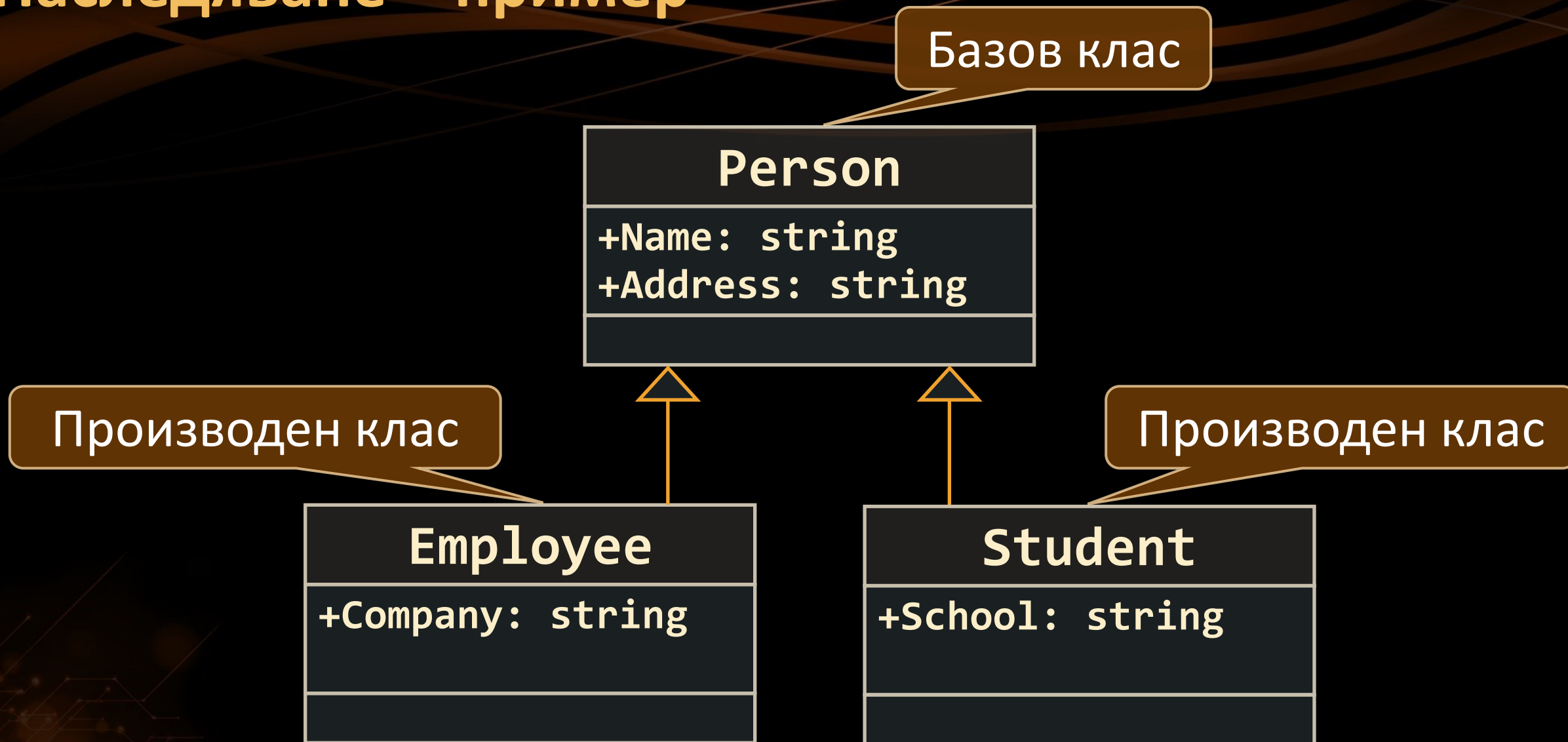
Разширяване на класове

Наследяване

- **Суперклас** – родителски клас, базов клас
 - Класът, който дава своите членове на дъщерния си клас
- **Подклас** – Дъщерен клас, производен клас
 - Класът, който получава членове от своя базов клас

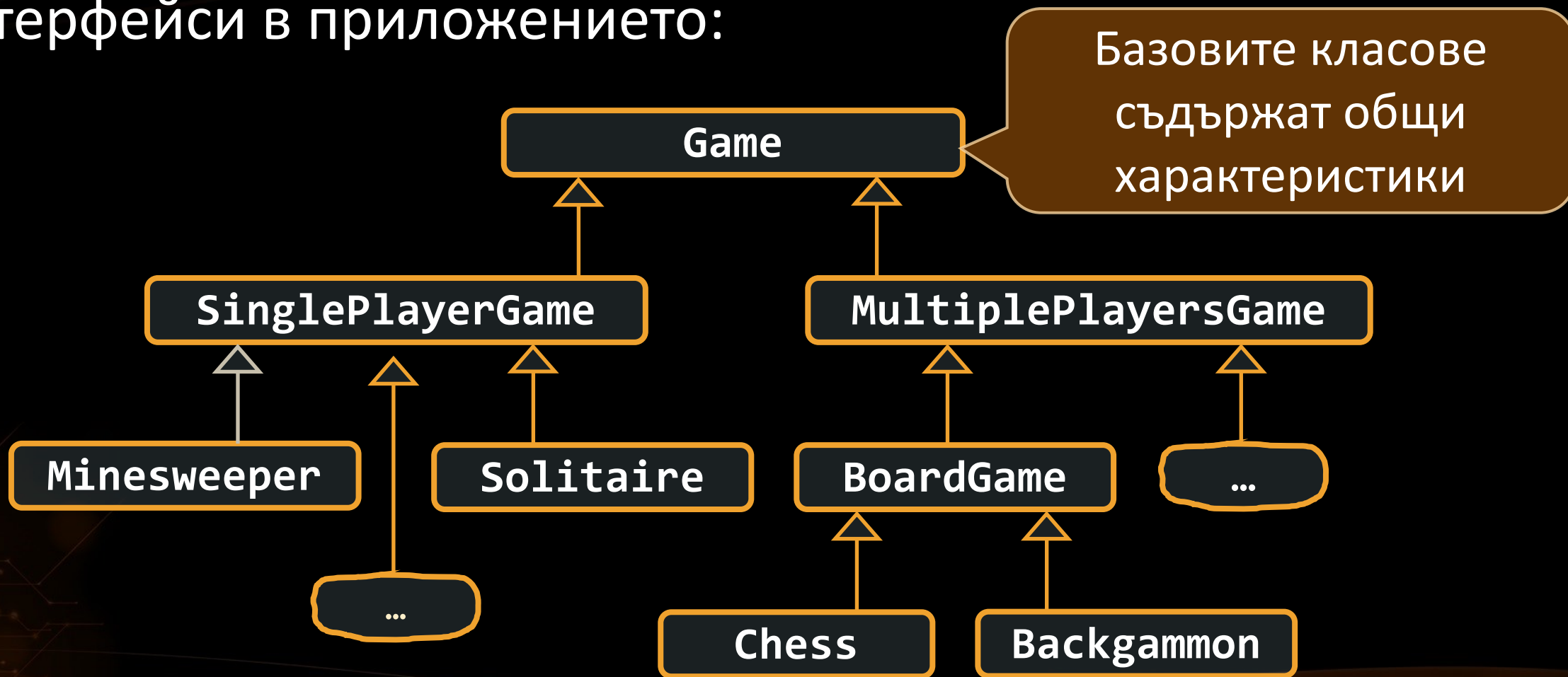


Наследяване – пример

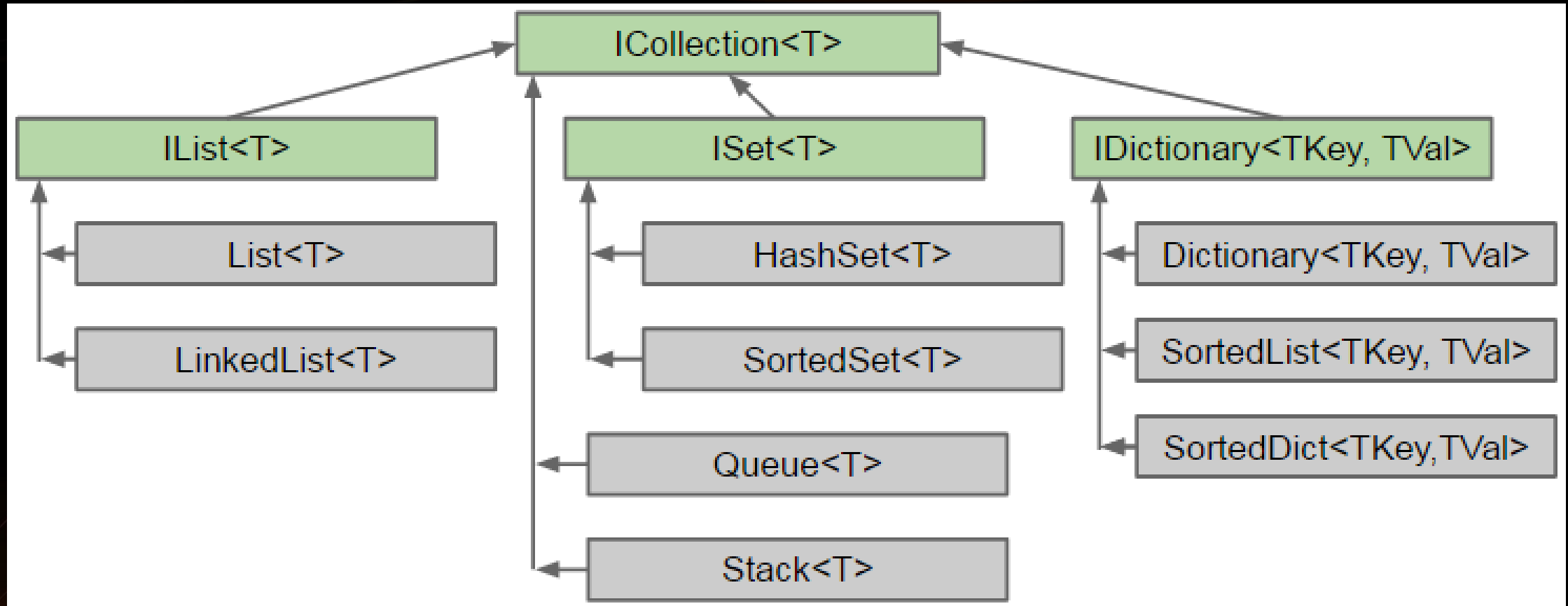


Йерархия от класове

- Наследяването води до йерархии от класове и/или интерфейси в приложението:



Йерархия от класове – C# Collection



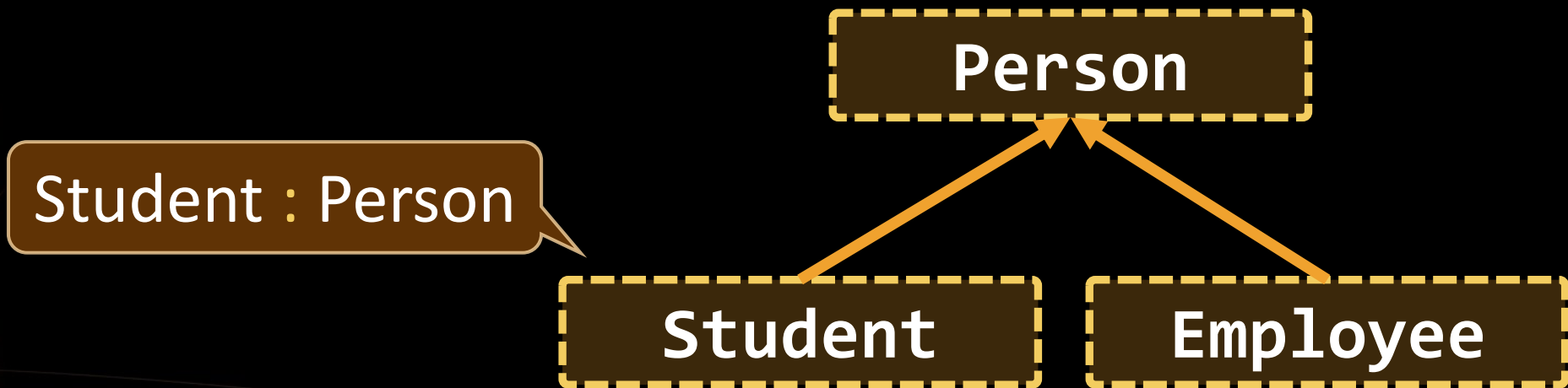
Наследяване в C#

- В C# наследяването се отбелязва чрез `:` оператора

```
class Person { ... }
```

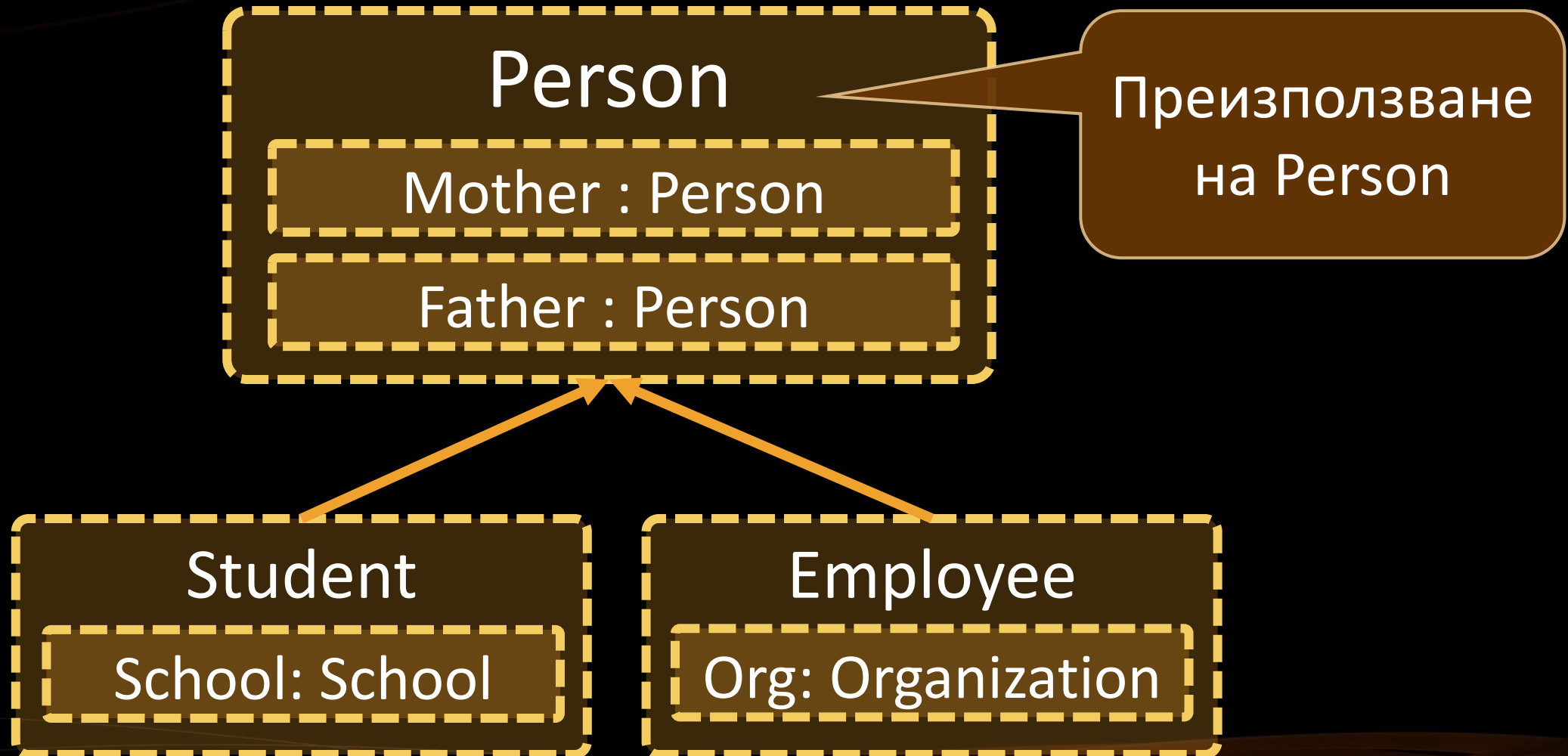
```
class Student : Person { ... }
```

```
class Employee : Person { ... }
```



Наследяване – дъщерен клас

- Класът получава всички членове от родителския си клас



Използване на наследени членове

- Наследените членове се използват както обикновено:

```
class Person { public void Sleep() { ... } }  
class Student : Person { ... }  
class Employee : Person { ... }
```

```
Student student = new Student();  
student.Sleep();  
Employee employee = new Employee();  
employee.Sleep();
```

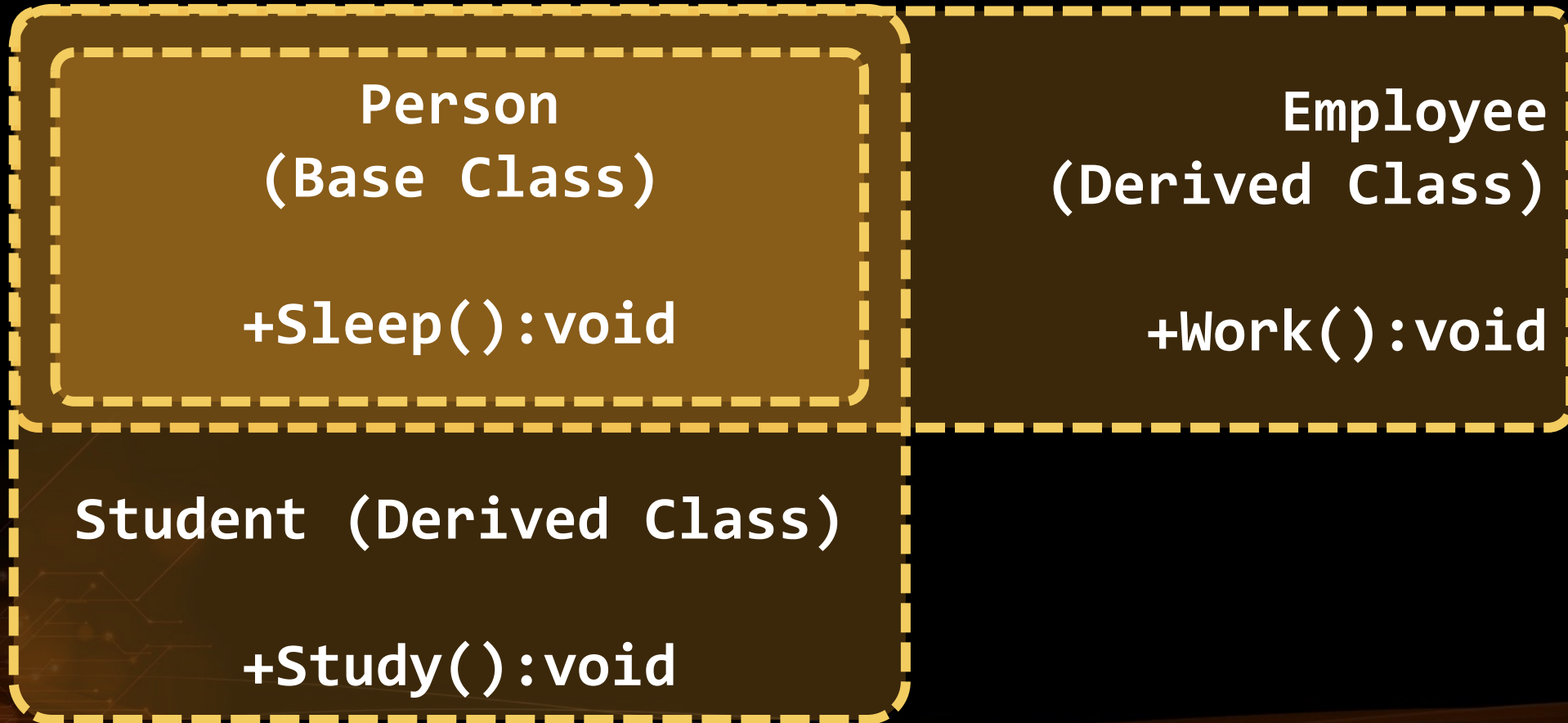
Преизползване на конструктори

- Конструкторите не се наследяват
- Конструкторите може да се ползват от дъщерните класове

```
class Student : Person {  
    private School school;  
    public Student(String name, School school)  
        :base(name)  
    {  
        this.school = school;  
    }  
}
```

Наследяването е разширяване

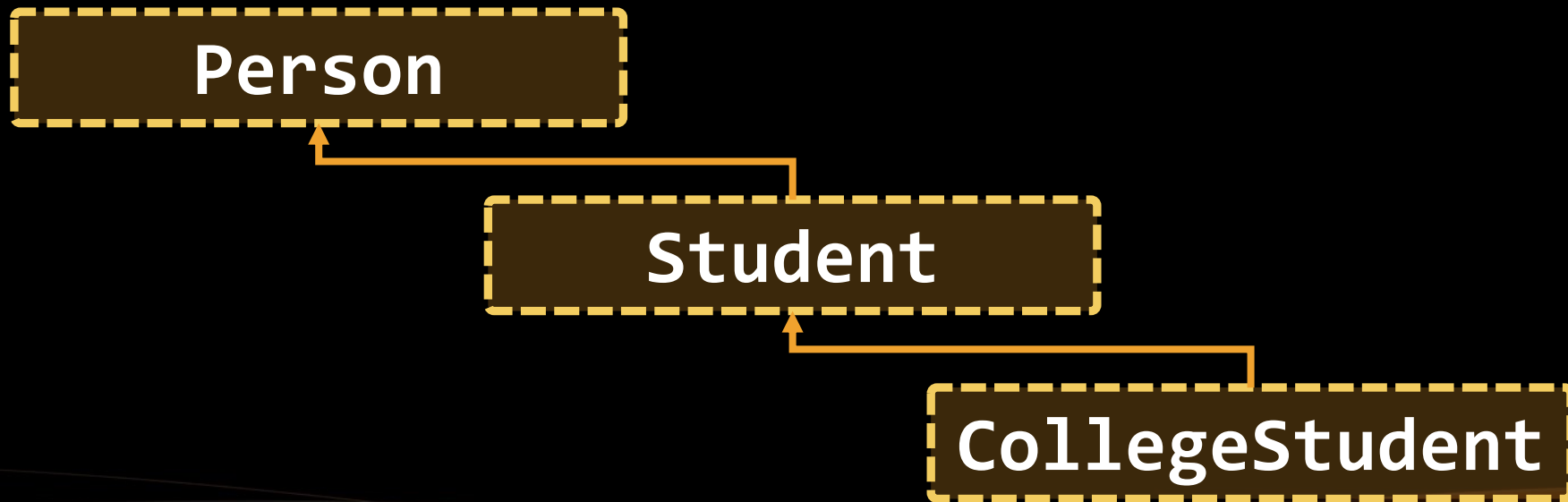
- Инстанцията на дъщерен клас **съдържа** инстанция на неговия базов клас



Наследяване

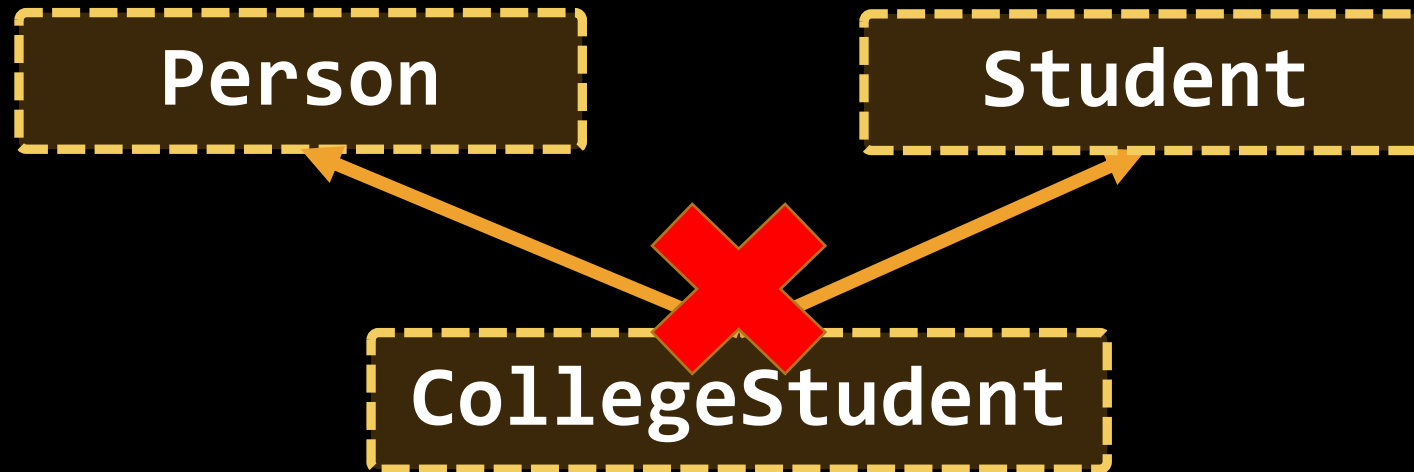
- Наследяването има **преходна връзка**

```
class Person { ... }  
class Student : Person { ... }  
class CollegeStudent : Student { ... }
```



Множествено наследяване

- В C# не се поддържа **множествено** наследяване
- Поддържа се само имплементиране на **множество** интерфейси

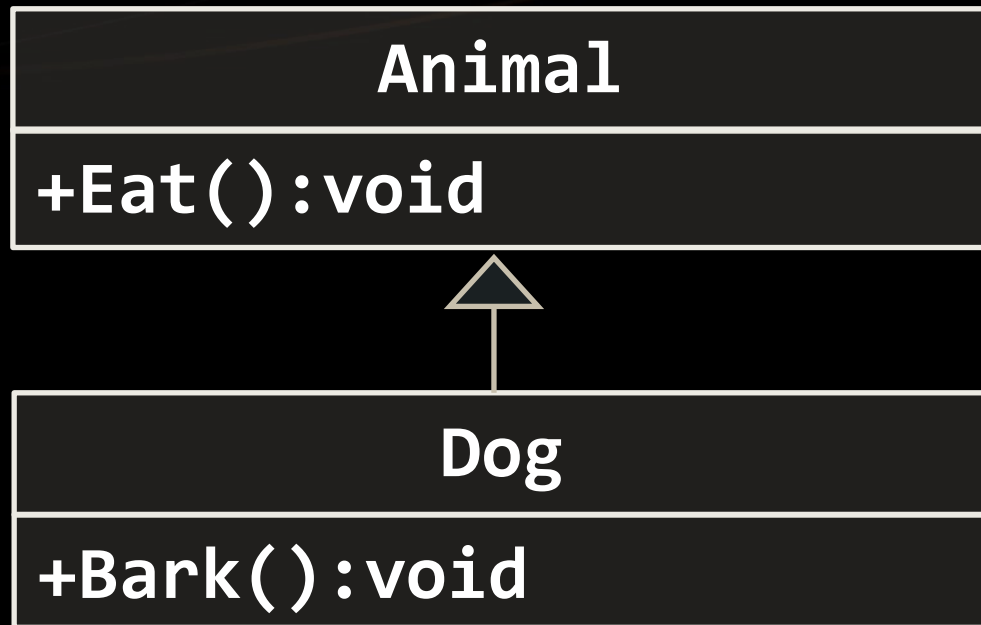


Достъп до членове на базовия клас

- Чрез ключовата дума **base**

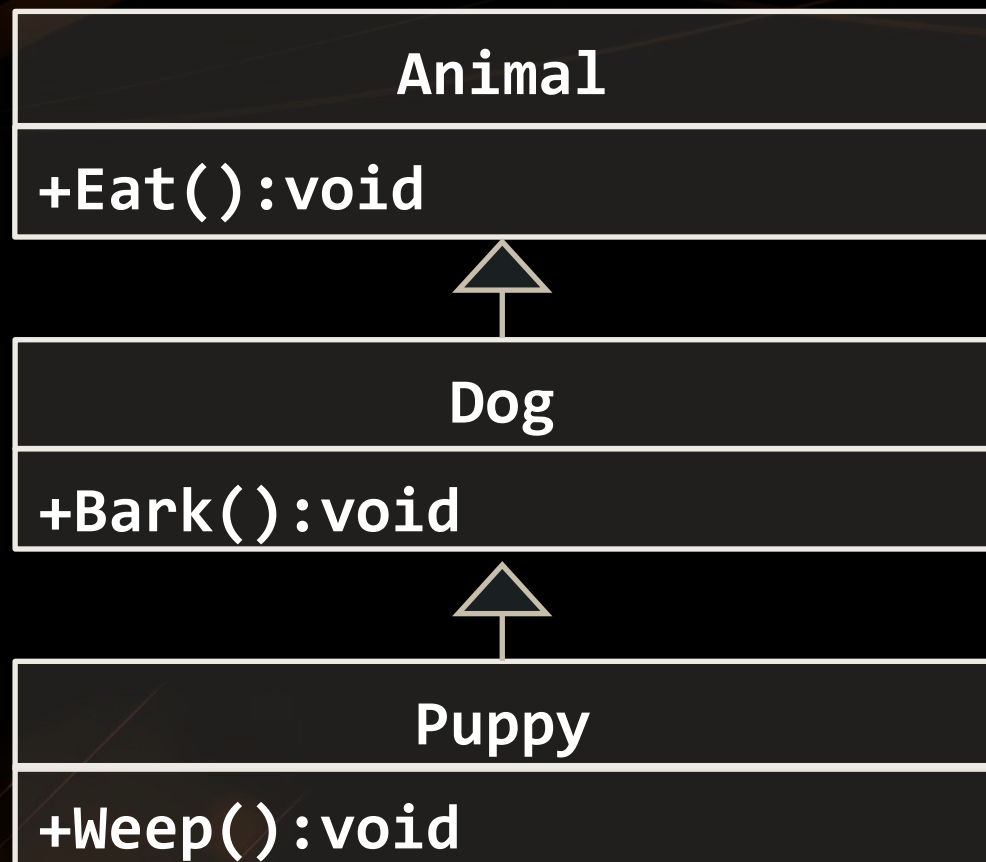
```
class Person { ... }  
class Employee : Person  
{  
    void Fire(string reasons) {  
        Console.WriteLine  
            ($"{base.name} got fired because {reasons}");  
    }  
}
```

Задача: Наследяване



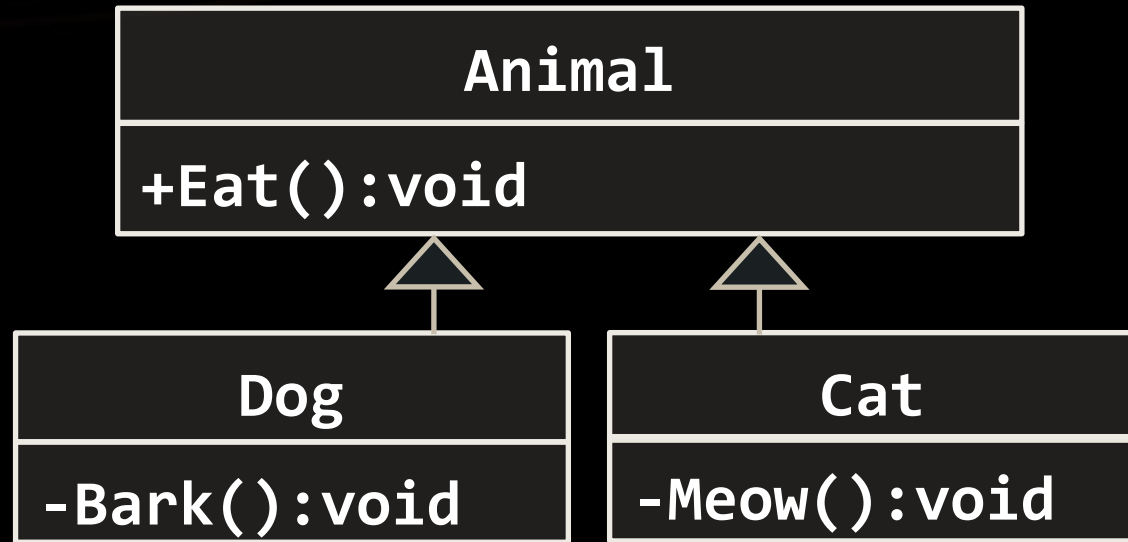
```
Dog dog = new Dog();  
dog.Eat();  
dog.Bark();
```


Задача: Наследяване на няколко нива



```
Puppy puppy = new Puppy();  
puppy.Eat();  
puppy.Bark();  
puppy.Weep();
```

Задача: Йерархично наследяване



```
Dog dog = new Dog();
dog.Eat();
dog.Bark();
```

```
Cat cat = new Cat();
cat.Eat();
cat.Meow();
```



Наследяване

Лаб

Какво научихме днес?

- Наследяването е силен инструмент за **преизползване на код**
- Подкласа наследява членове от **суперкласа**



Наследяване



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

