



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

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



# Преизползване на Класове

Reuse

# Съдържание

1. Преизползване на класове
2. Кога да използване наследяване
3. Композиция



# Преизползване на Класове

Преизползване на код на ниво клас

# Наследяване и модификатори за достъп

- Подкласовете могат да достъпят всички **публични** и **защитени** членове, както и могат да достъпят **вътрешните** членове, ако са в същия проект
- **Private** полетата не се наследяват в подкласовете

```
class Person {  
    private string id;  
    string name;  
    protected string address;  
    public void Sleep();  
}
```

Може да се достъпи  
чрез други методи

# „Засенчване“ на променливи

Подкласовете **могат да скрият** променливи от суперкласа

```
class Person { protected int weight; }
```

```
class Patient : Person  
{  
    protected float weight;  
    public void Method()  
    {  
        double weight = 0.5d;  
    }  
}
```

Скрива int weight

Скрива и двете

„Засенчване“ на променливи – достъп  
Използвайте `base` и `this` да уточните достъпа

```
class Patient : Person
{
    protected float weight,
    public void Method()
    {
        double weight = 0.5d;
        this.weight = 0.6f;
        base.weight = 1;
    }
}
```

Локална  
променлива

Член на инстанцията

Член на базовия клас

# Виртуални методи

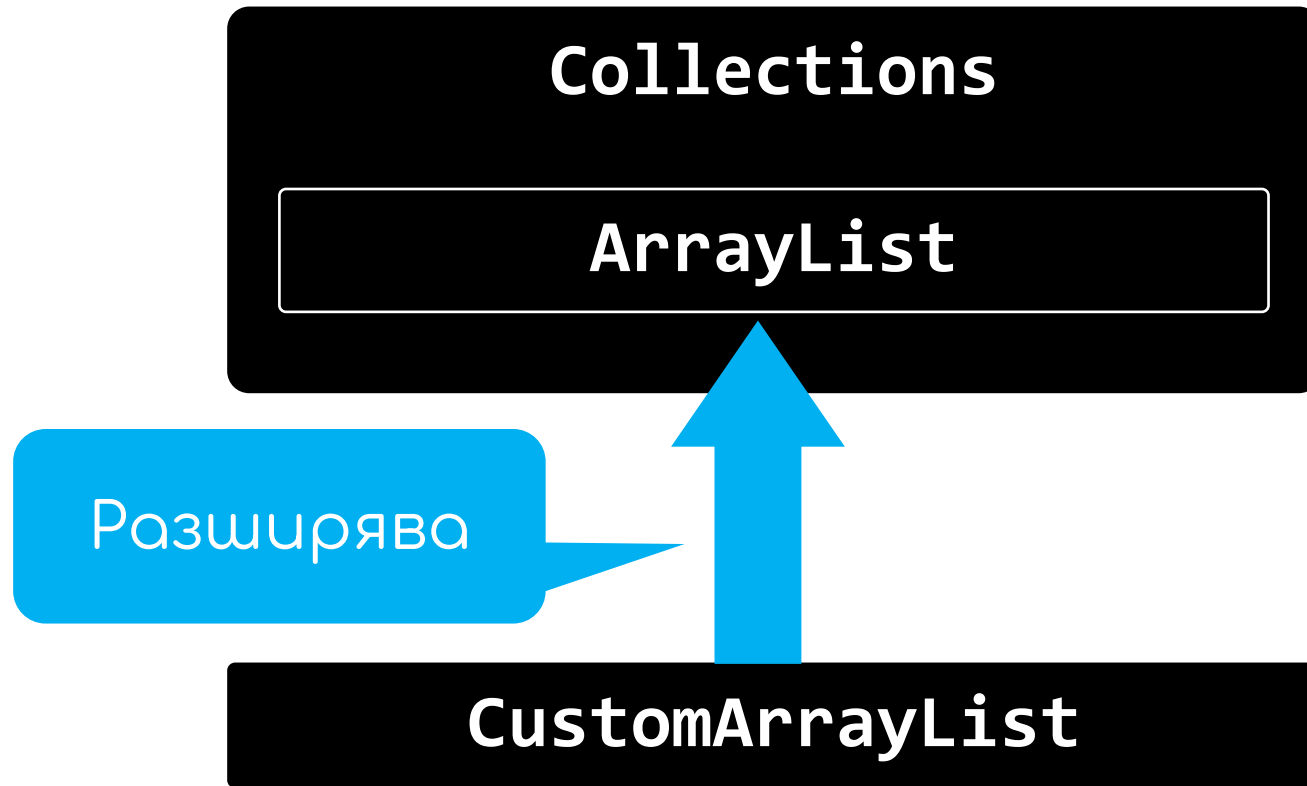
**virtual** метод, може да бъде презаписан с **override**

```
public class Animal
{
    public virtual void Eat() { ... }
}
```

```
public class Dog : Animal
{
    public override void Eat() {}
}
```

# Ползата от наследяването – разширяние

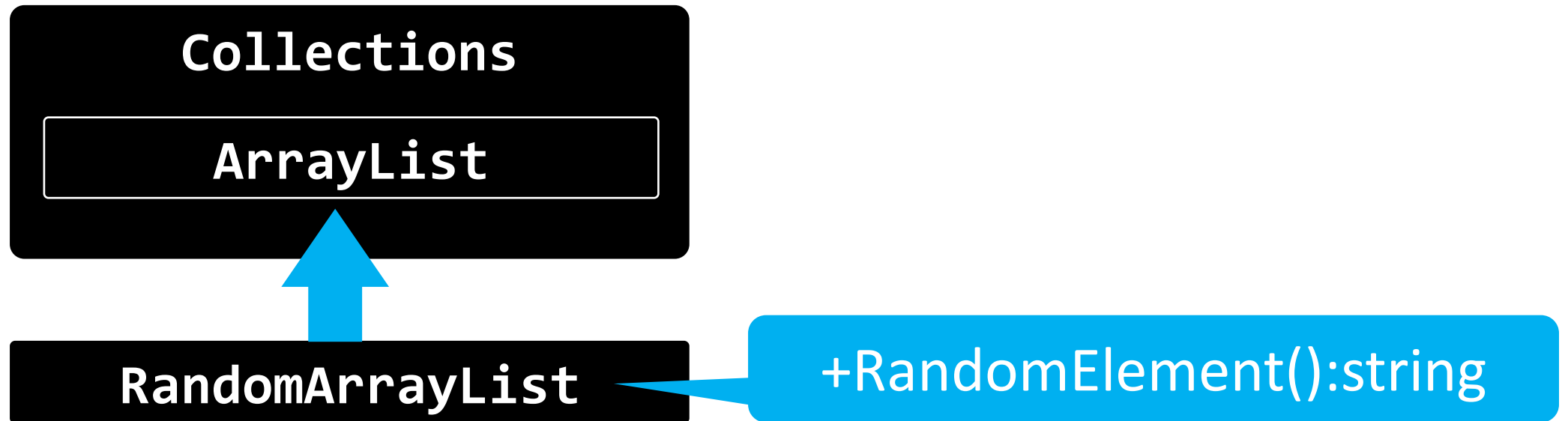
Можем да **разширим** клас, който иначе не можем да променим





# Задача: Случаен Array List

- Създайте **array list**, който има
  - Всичката функционалност на **ArrayList**
  - Функция, която връща и премахва случаен елемент



# Решение: Случаен Array List

```
public class RandomList : ArrayList
{
    private Random rnd;
    public object RandomString()
    {
        int element = rnd.Next(0, data.Count - 1);
        string str = data[element];
        data.Remove(str);
        return str;
    }
}
```

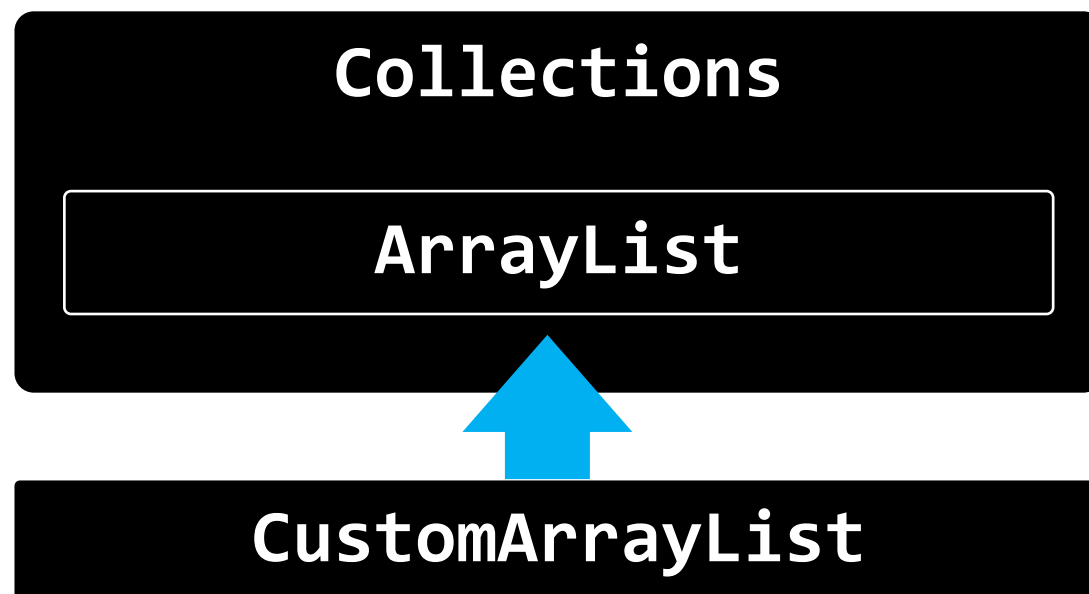


# Видове преизползване на класове

Разширяване, композиция, делегиране

# Разширяване

- **Дублирането** на код е податливо на грешки
- **Преизползване** на код чрез **разширение**
- Понякога това е единствения начин



# Композиция

Използване на класове за дефиниране на друг клас

```
class Laptop {  
    Monitor monitor;  
    Touchpad touchpad;  
    Keyboard keyboard;  
    ...  
}
```

Преизползване  
на класове

Laptop

Monitor

Touchpad

Keyboard

# Делегуране

```
class Laptop
{
    Monitor monitor;

    void IncrBrightness()
    monitor.Brighten();

    void DecrBrightness()
    monitor.Dim();
}
```

Laptop

Monitor

increaseBrightness()  
decreaseBrightness()

# Задача: Стек от низове

Създайте прост клас **Stack**, който може да съдържа само низове

**StackOfStrings**

**-data: List<String>**

**+Push(string): void**

**+Pop(): string**

**+Peek(): string**

**+IsEmpty(): boolean**

**StackOfStrings**

**List**

# Решение: Стек от низове

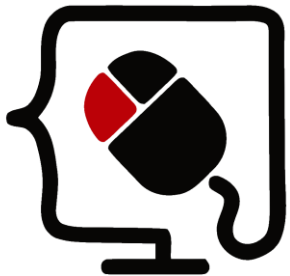
```
public class StackOfStrings {  
    private List<String> data;  
    public void Push(string element)  
    {  
        this.data.Add(element);  
    }  
    public string Pop()  
    {  
        var element = this.data.Last();  
        this.data.Remove(element);  
        return element;  
    }  
}
```

**TODO:** Проверявайте за празен списък



# Какво научихме днес?

- Търсете класове със **същата роля**
- Използвайте **композиция** и **делигиране** вместо това



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).