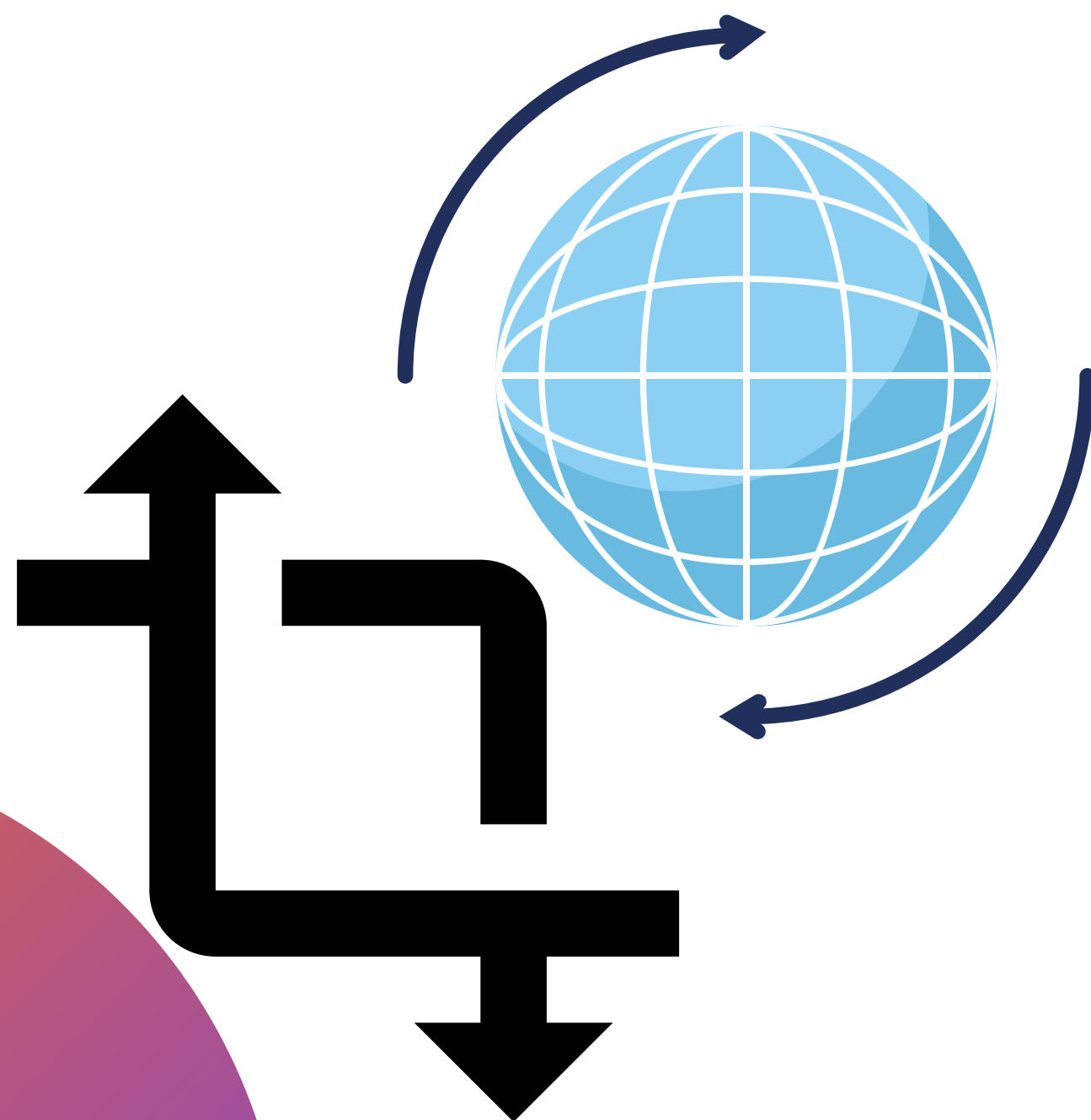




# Курс

# Advanced – C#

ЗАНЯТТЯ № 2 - ВІРТУАЛЬНІ МЕТОДИ ТА  
ПРИХОВУВАННЯ МЕТОДІВ



# Віртуальні методи та властивості

При наслідуванні нерідко виникає необхідність змінити в класі-спадкоємці функціонал методу, який успадкував від базового класу. У цьому випадку клас-спадкоємець може перевизначати методи та властивості базового класу.

Ті методи та властивості, які ми хочемо зробити доступними для перевизначення, у базовому класі позначається модифікатором `virtual`. Такі методи та властивості називають віртуальними.

А щоб перевизначити метод у класі-спадкоємці, цей метод визначається з модифікатором `override`. Перевизначений метод у класі-спадкоємці повинен мати той самий набір параметрів, що й віртуальний метод у базовому класі.

# Правила використання

- Віртуальний і перевизначений методи повинні мати той самий модифікатор доступу. Тобто якщо віртуальний метод визначено з допомогою модифікатора `public`, то й перевизначений метод повинен мати модифікатор `public`.
- Не можна перевизначити чи оголосити віртуальним статичний метод.







# Ключове слово **base**

Крім конструкторів, ми можемо звернутись за допомогою ключового слова `base` до інших членів базового класу.

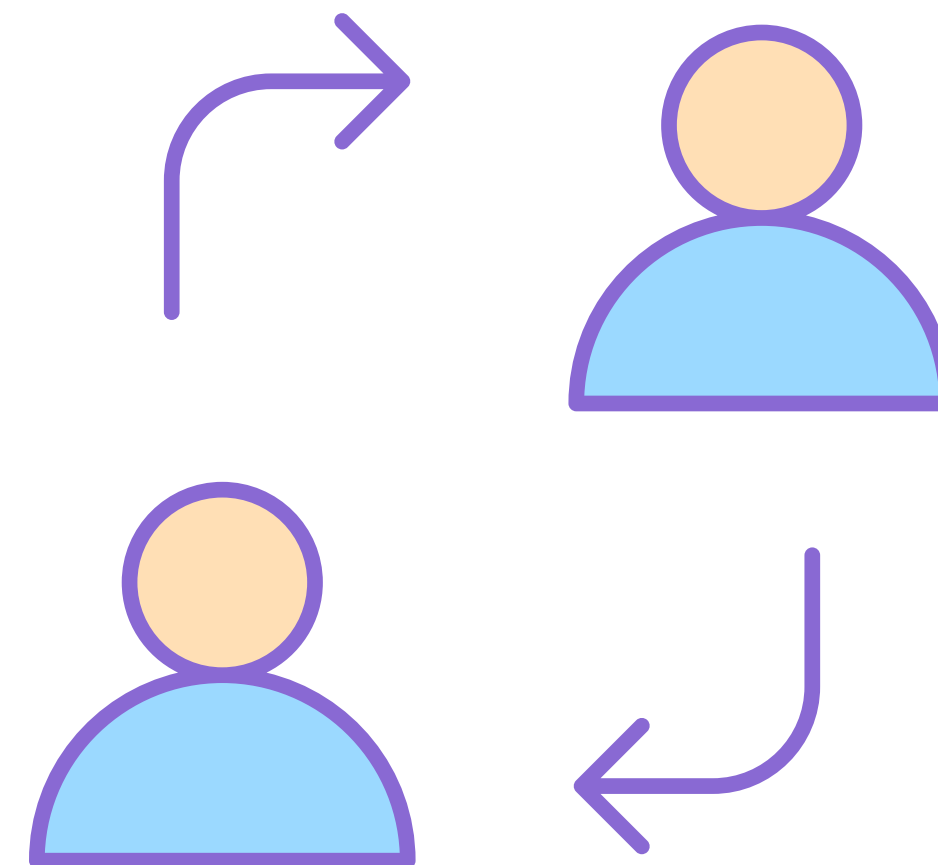


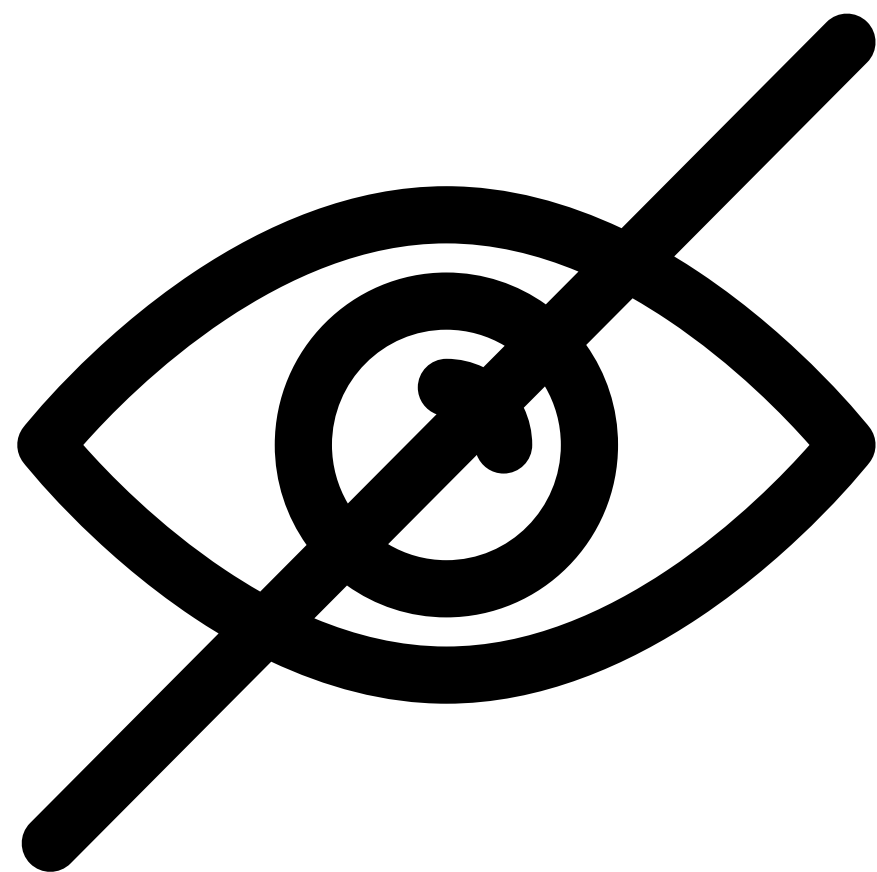
# Перевизначення властивостей

Так само як і методи, можна перевизначати властивості:

```
public virtual int Age
{
    get => age;
    set{ if(value > 0 && value < 110) age = value; }
}

public override int Age
{
    get => base.Age;
    set { if (value >= 18 && value < 110) base.Age = value; }
}
```





# Приховування методів та властивостей

Іншим способом змінити функціональність методу, успадкованого від базового класу, є приховування (shadowing/hiding).

Фактично приховування методу/властивості представляє визначення в класі-спадкоємці методу або властивості, які відповідає за ім'ям та набором параметрів методу або властивості базового класу. Для приховування членів класу застосовується ключове слово `new`.

```
public new void Print()  
{  
    Console.WriteLine($"Name: {Name} Company: {Company}");  
}
```



# Ключове слово **base**

Так само як при наслідуванні або перевизначенні, якщо ми хочемо звернутися саме до реалізації властивості або методу в базовому класі, то ми можемо використовувати ключове слово `base` і через нього звертатися до функціональності базового класу.

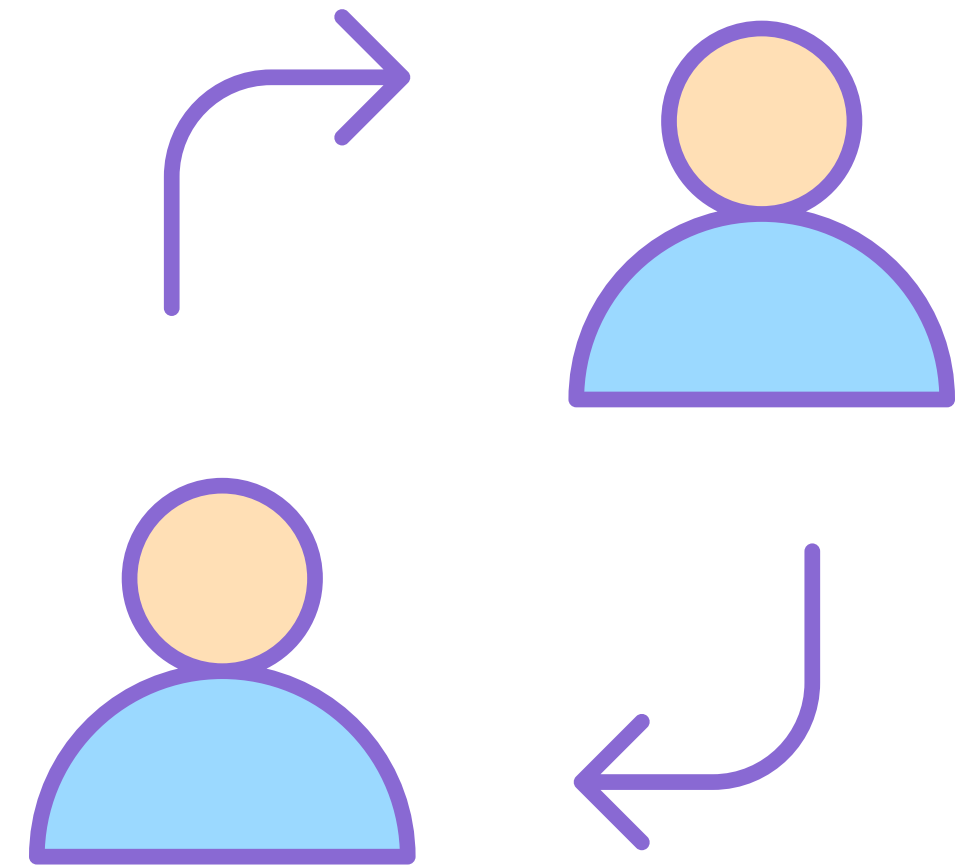


# Приховування властивостей

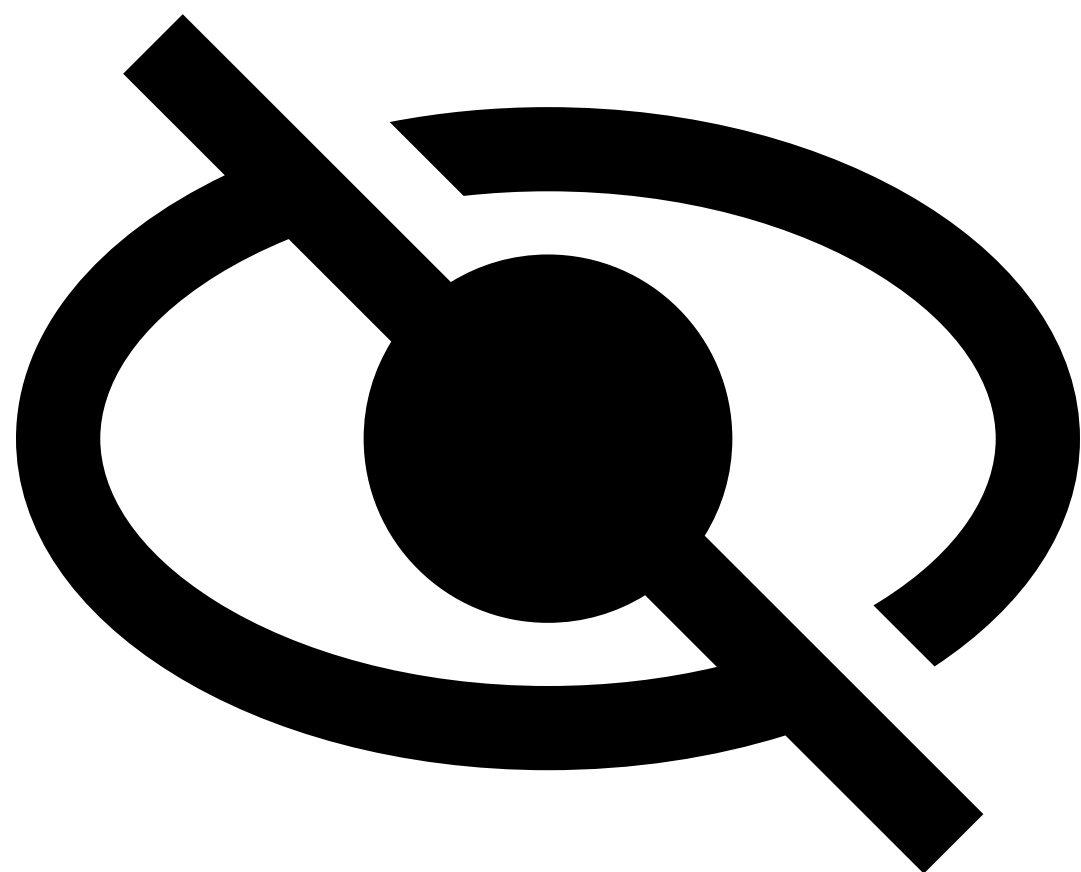
Подібним образом ми можемо організувати приховування властивостей:

```
public int Age
{
    get => age;
    set{ if(value > 0 && value < 110) age = value; }
}

public new int Age
{
    get => base.Age;
    set { if (value >= 18 && value < 110) base.Age = value; }
}
```







# Приховування змінних та констант

На відміну від перевизначення C# дозволяє застосовувати приховування до змінних (як до статичних, так і нестатичних) та константів, також використовуючи ключове слово `new` :

```
class Person
{
    public readonly static int minAge = 1;
    public const string typeName = "Person";
}
class Employee : Person
{
    // скриваєм поля і константи базового класу
    public new readonly static int minAge = 18;
    public new const string typeName = "Employee";
}
```





Дякую за увагу