



# Курс **Advanced – C#**

ЗАНЯТТЯ № 1 - УСПАДКУВАННЯ



# Спадкування

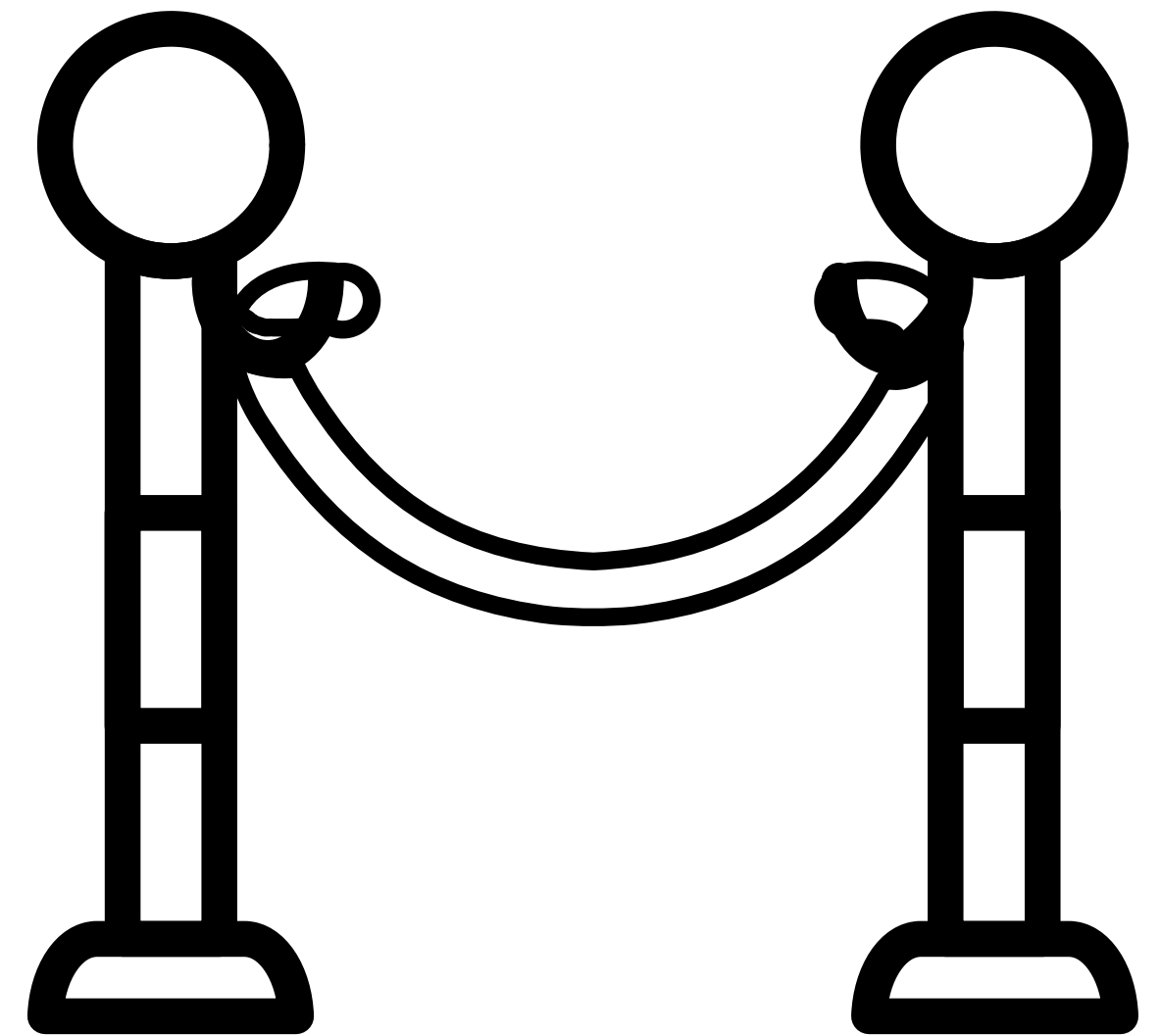
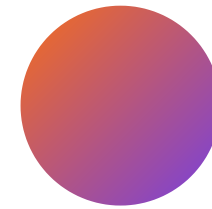
Спадкування (inheritance) є одним із ключових моментів ООП. Завдяки успадкуванню один клас може успадкувати функціональність іншого класу.

Таким чином, успадкування реалізує відношення is-a (є)

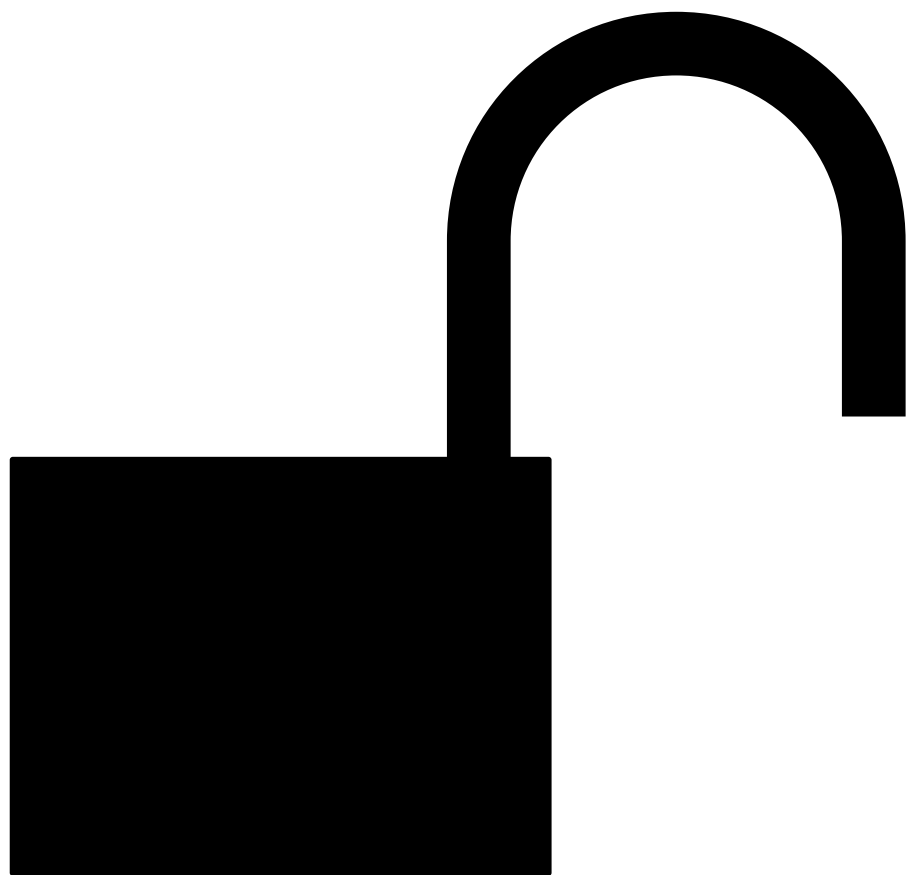
За умовчанням усі класи успадковуються від базового класу `Object`, навіть якщо ми явно не встановлюємо спадкування. Тому крім своїх методів, також матимуть і методи класу `Object`: `ToString()`, `Equals()`, `GetHashCode()` і `GetType()`.

# Обмеження

- В C# не підтримується множинне спадкування, клас може успадковуватися тільки від одного класу.
- При створенні похідного класу треба враховувати тип доступу до базового класу - тип доступу до похідного класу повинен бути таким самим, як і у базового класу, або більш суворим. Тобто, якщо базовий клас у нас має тип доступу `internal`, то похідний клас може мати тип доступу `internal` або `private`, але не `public`. Проте слід враховувати, якщо й базовий і похідний клас перебувають у різних складаннях (проектах), то в цьому випадку похідний клас може успадковувати лише від класу, що має модифікатор `public`.
- Якщо клас оголошений з модифікатором `sealed`, то від цього класу не можна успадковувати і створювати похідні класи.





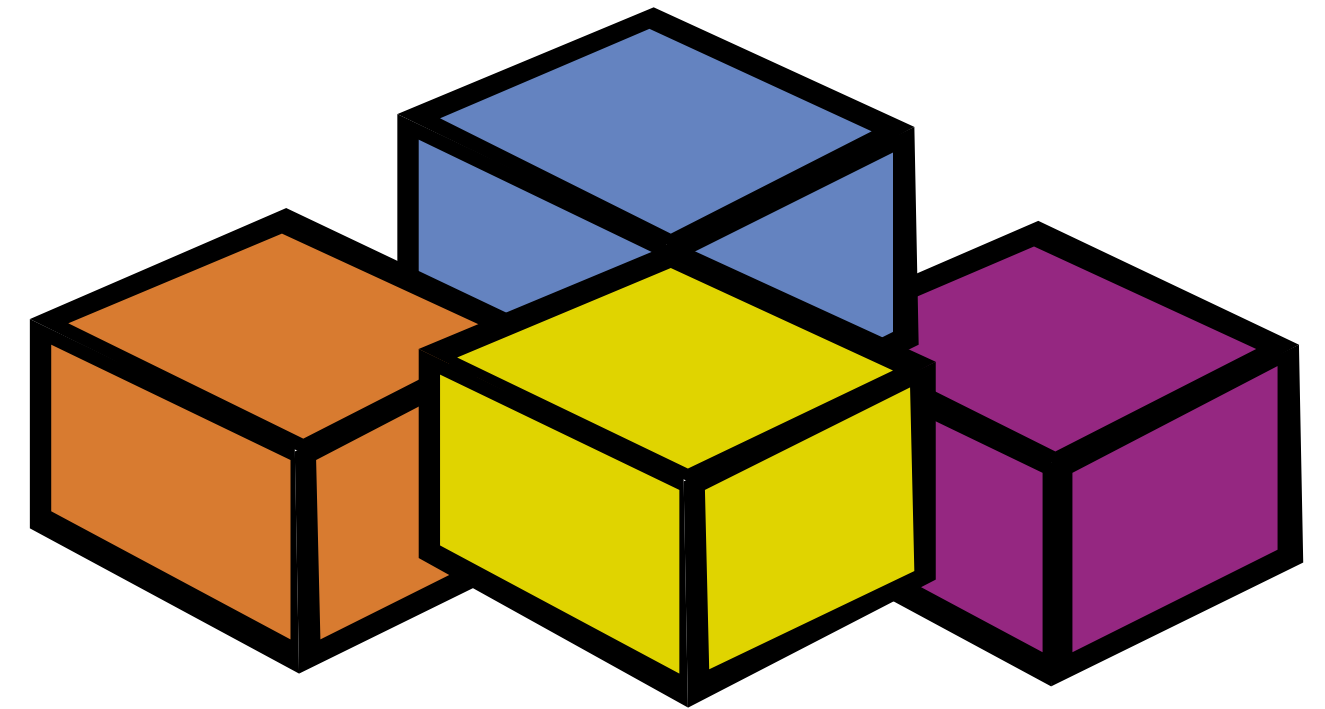


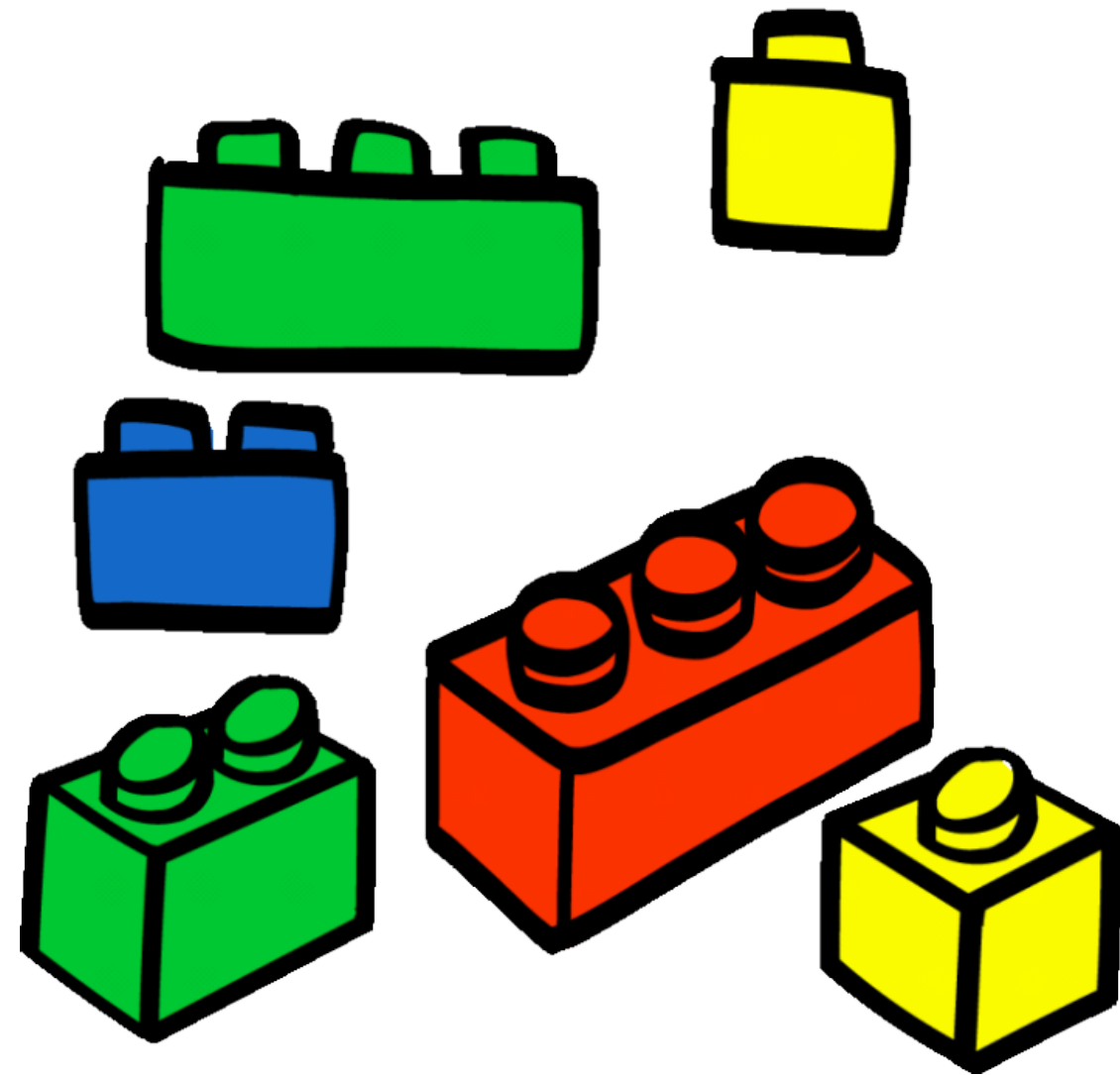
## Доступ до членів базового класу із класу-спадкоємця

Похідний клас може мати доступ тільки до тих членів базового класу, які визначені з модифікаторами `protected`, `public`, `internal` і `protected internal`.

# Ключове слово `base`

За допомогою ключового слова `base` ми можемо звернутися до базового класу. Наприклад у конструкторі класу `Employee` нам треба встановити ім'я та компанію. Але ім'я ми передаємо на встановлення в конструктор базового класу, тобто конструктор класу `Person`, за допомогою виразу `base(name)`.



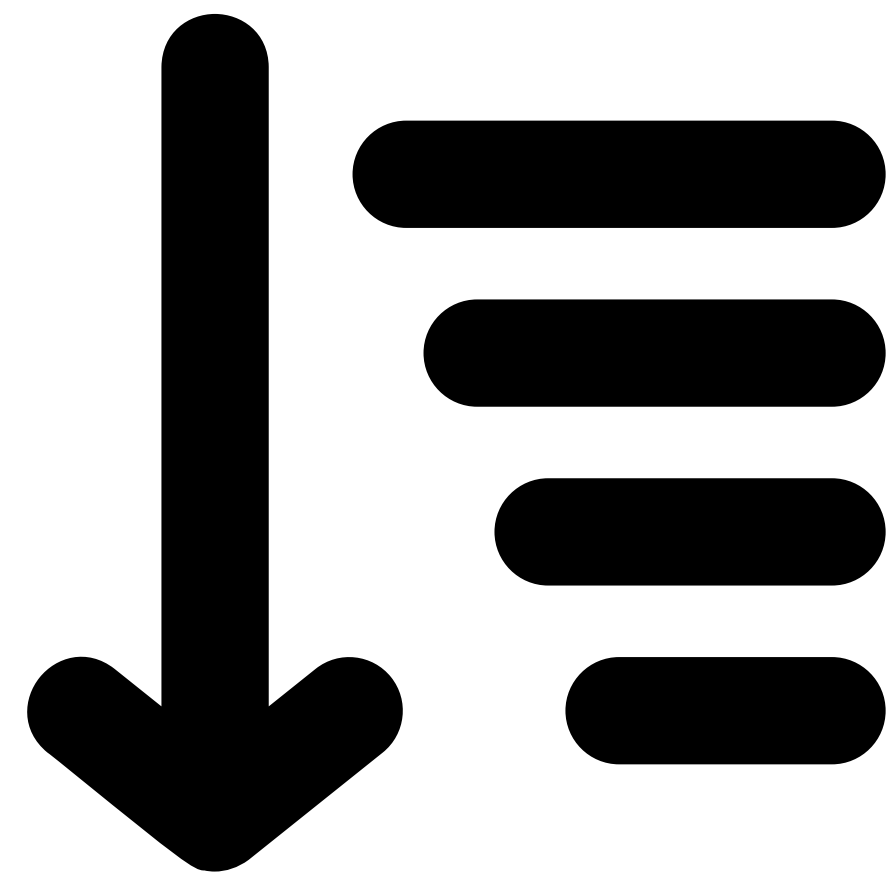


## Конструктори у похідних класах

Конструктори не передаються похідному класу під час успадкування. І якщо в базовому класі не визначено конструктора за замовчуванням без параметрів, а лише конструктори з параметрами (як у випадку з базовим класом `Person`), то у похідному класі ми обов'язково повинні викликати один з цих конструкторів через ключове слово `base`.

# Порядок виклику конструкторів

При виклику конструктора класу спочатку відпрацьовують конструктори базових класів і потім конструктори похідних.







Дякую за увагу