Знайомство з ООП. Поглиблене вивчення мови C#

Тема 2. Спадкування класів та поліморфізм

Об'єктно-орієнтоване програмування

- ООП – методологія програмування
  - Програма є сукупністю об'єктів
  - Кожен об'єкт – екземпляр класу
  - Класи утворюють ієрархію спадкування
- ООП використовує як базові елементи об'єкти, а не алгоритми

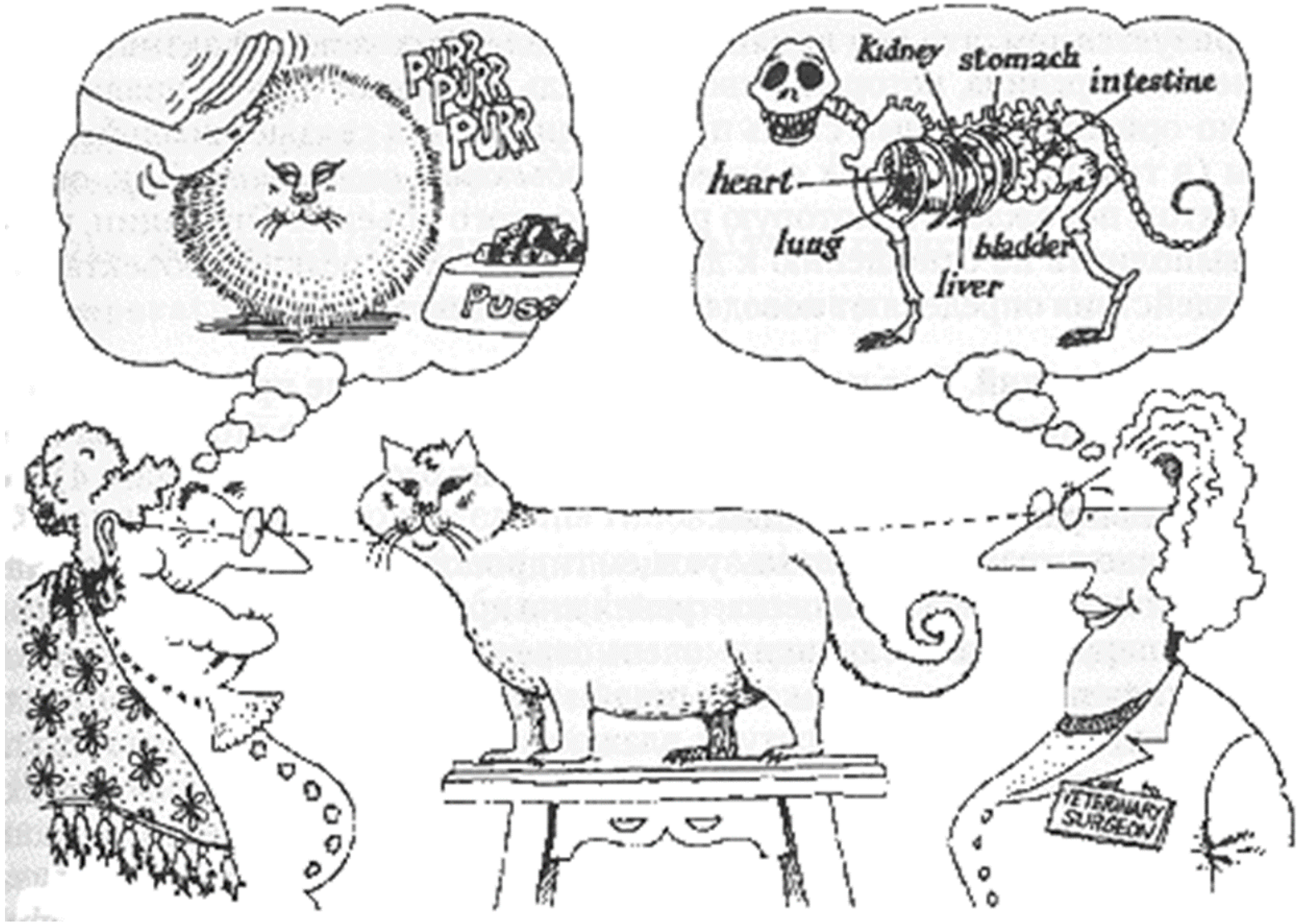
Основні принципи об'єктно-орієнтованого програмування

- Абстракція
- Інкапсуляція
- Спадкування
- Поліморфізм

Абстракція

- Абстрагування - виділення істотних з погляду спостерігача характеристик об'єкта, які відрізняють його від інших видів об'єктів
- Виділяйте лише ті фактори, які потрібні для вирішення задачі
- Відсікайте все зайве, що не впливає на результат розв'язання задачі

Абстракція


PURR PURR PURR
Kidney
stomach
intestine
heart
lung
bladder
liver
VETERINARY SURGEON

Спадкування

- В об'єктно-орієнтованому програмуванні, спадкування (наслідування) — механізм утворення нових класів на основі використання вже існуючих. При цьому властивості та функціональність батьківського класу переходять до класу нащадка (дочірнього).
- Спадкування класів – перенесення структури та функціональності базового класу в похідний клас. Множина класів, пов'язаних спадкуванням, формує ієрархію спадкування.

Приклад спадкування

ЛЮДИ

СТУДЕНТИ

ШКОЛЯРІ

Спадкування (клас нащадок успадковує поля)

```
class Person
{
    public string firstname;     // Ім'я
    public string lastname;      // Прізвище
    public int age;              // Вік
}

2 references
class Student : Person
{
    public string university;  // Університет
}
```

Спадкування (клас нащадок успадковує поля та методи)

```
class Person
{
    public string firstname;     // Ім'я
    public string lastname;      // Прізвище
    public int age;              // Вік

    2 references
    public  void AboutMe()
    {
        Console.WriteLine($"Меня звати {firstname} {lastname} і мені {age} років.");
    }
}

2 references
class Student : Person
{
    public string university;
}
```

Перекриття методу батьківського класу

```
class Person
{
    public string firstname;    // Ім'я
    public string lastname;     // Прізвище
    public int age;             // Вік

    0 references
    public void AboutMe()
    {
        Console.WriteLine($"Меня звати {firstname} {lastname} і мені {age} років.");
    }
}
2 references
class Student : Person
{
    public string university;
    // Перекриття методу батьківського класу
    1 reference
    public new void AboutMe()
    {
        Console.WriteLine($"Я {firstname} {lastname} і я студент {university}");
    }
}
```

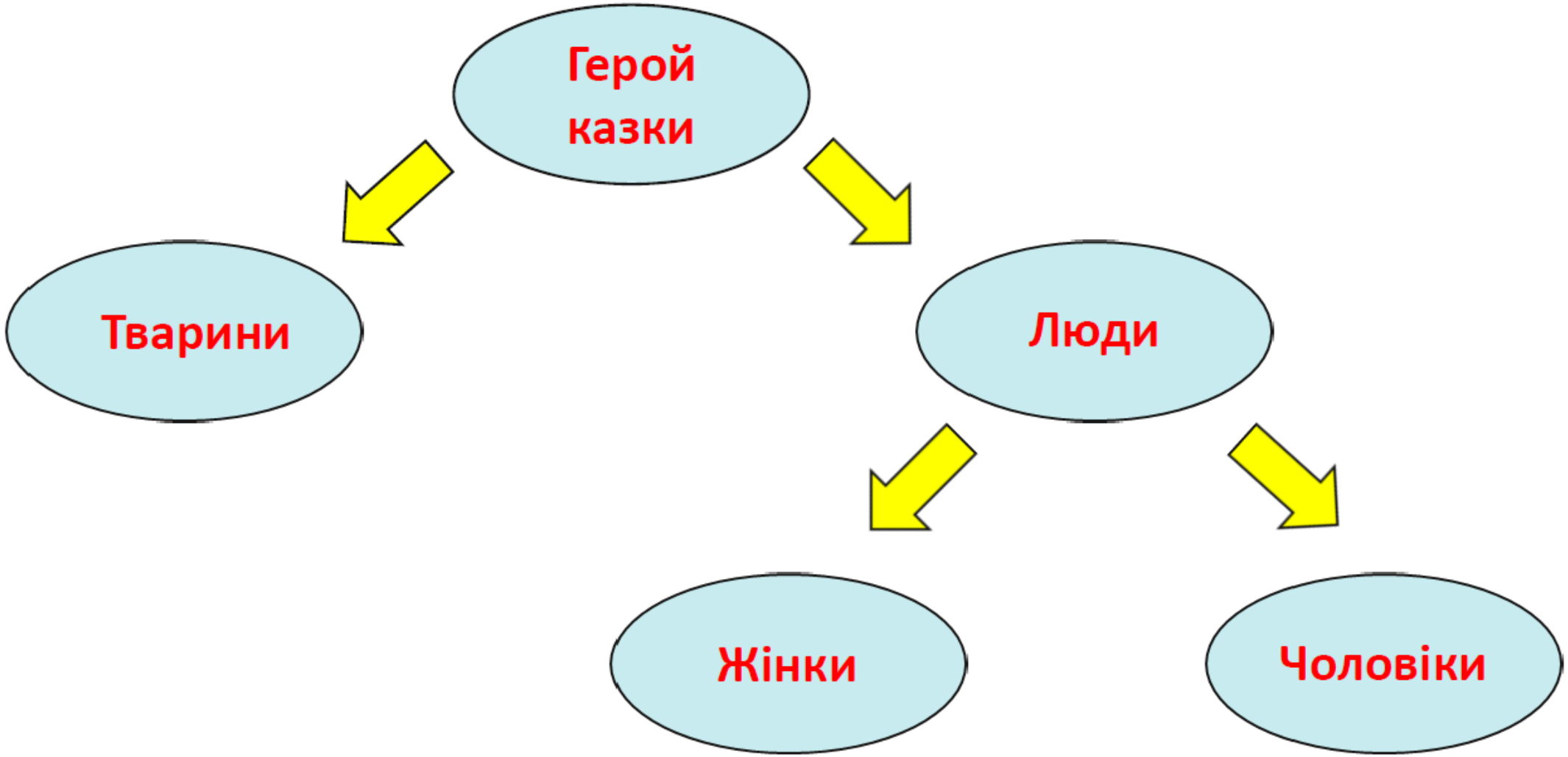
Перевизначення віртуального методу

```
class Person
{
    public string firstname;     // Ім'я
    public string lastname;      // Прізвище
    public int age;              // Вік
    2 references
    public virtual void AboutMe()
    {
        Console.WriteLine($"Меня звати {firstname} {lastname} і мені {age} років.");
    }

}

2 references
class Student : Person
{
    public string university;
    // Перевизначення віртуального методу
    2 references
    public override void AboutMe()
    {
        Console.WriteLine($"Я {firstname} {lastname} і я студент {university}");
    }
}
```

Приклад спадкування



Поліморфізм

- Поліморфізм - можливість об'єктів з однаковою специфікацією мати різну реалізацію. Іншими словами - коли функції з однаковою назвою реалізовують різну логіку. Підтримується в багатьох мовах програмування через перевантаження функцій.
- Поліморфізм – полягає у використанні однакових підходів для роботи з різними за функціональністю об'єктами.

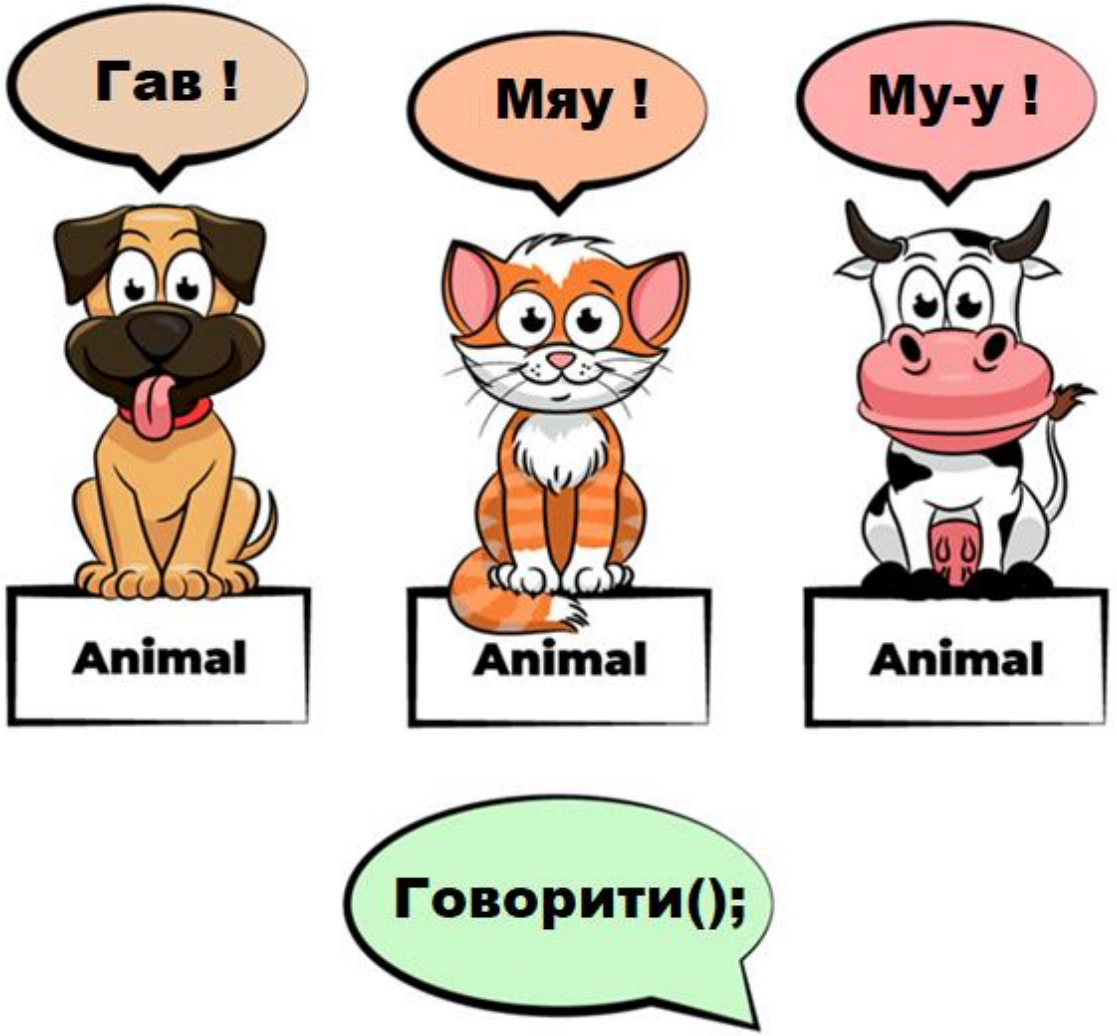
Поліморфізм (метод «Увімкнути»)

Щоб увімкнути, треба підняти

Щоб увімкнути, треба натиснути

Щоб увімкнути, треба повернути

Поліморфізм (метод «Говорити»)



Поліморфізм (метод «Говорити»)

```
class Animal
{
    3 references | 0 changes | 0 authors, 0 changes
    public virtual void Speak()
    {
        Console.WriteLine("Я можу говорити");
    }
}
```

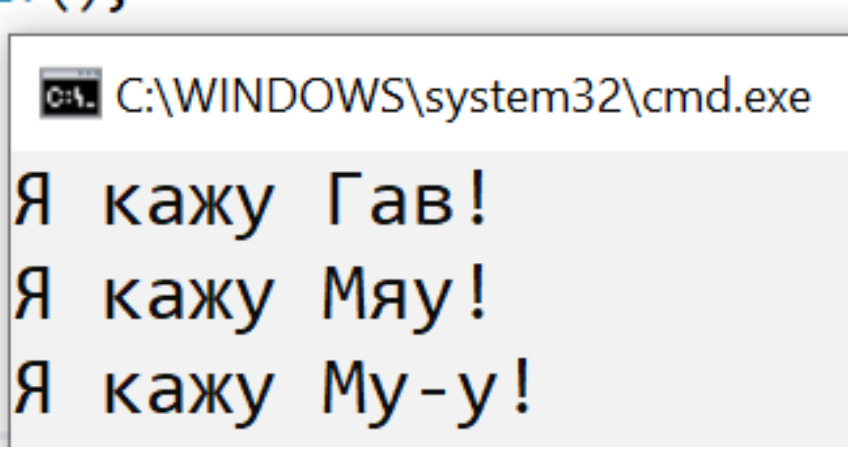
```
0 references | 0 changes | 0 authors, 0 changes
class Puppy : Animal
{
    3 references | 0 changes | 0 authors, 0 changes
    public override void Speak()
    {
        Console.WriteLine("Я кажу Гав!");
    }
}
```

```
class Kitten : Animal
{
    3 references | 0 changes | 0 authors, 0 changes
    public override void Speak()
    {
        Console.WriteLine("Я кажу Мяу!");
    }
}
```

```
0 references | 0 changes | 0 authors, 0 changes
class Calf : Animal
{
    3 references | 0 changes | 0 authors, 0 changes
    public override void Speak()
    {
        Console.WriteLine("Я кажу Му-у!");
    }
}
```

Поліморфізм (метод «Говорити»)

```
class Program
{
    0 references | Олександр Щербаков, 14 minutes ago | 1 author, 1 change
    static void Main(string[] args)
    {
        Puppy puppy = new Puppy();
        Kitten kitten = new Kitten();
        Calf calf = new Calf();
        puppy.Speak();
        kitten.Speak();
        calf.Speak();
    }
}
```



Домашнє завдання

- Напишіть клас Automobile (Автомобіль) з полями «Назва» та «Максимальна швидкість» автомобіля. Додайте до класу необхідні методи, конструктори та властивості.
- Створіть ще два класи, які повинні бути нащадками від цього класу. Клас Car (Легковий автомобіль) та клас Vehicle (Вантажний автомобіль). До першого класу додайте поле, яке визначає кількість місць для пасажирів, а до другого –поле, яке визначає вагу вантажу, який може перевозити вантажний автомобіль.
- Створіть у класі Program масив об’єктів типу Автомобіль, додайте до нього два об’єкти, один класу Car, а другий – класу Vehicle. Знайдіть та виведіть усю інформацію про автомобіль, який має більшу швидкість.