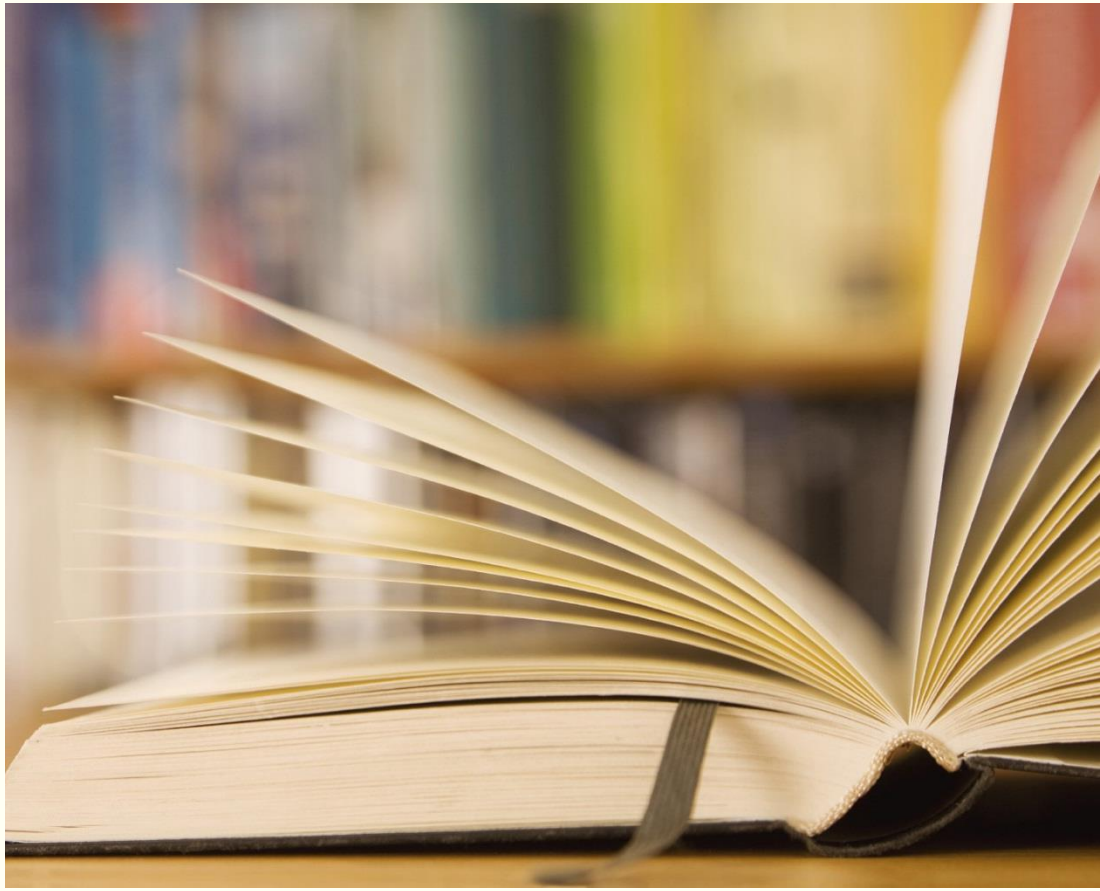
ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ МОВОЮ PYTHON

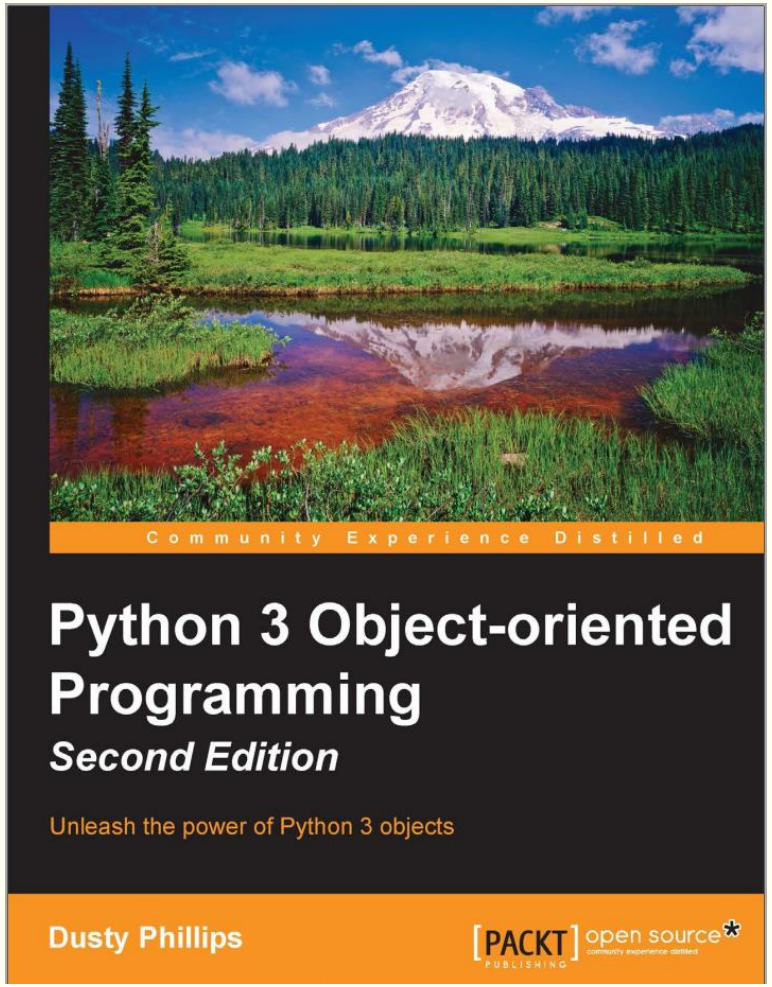
Лекція 08
Основи інформатики, програмування та алгоритмічні мови

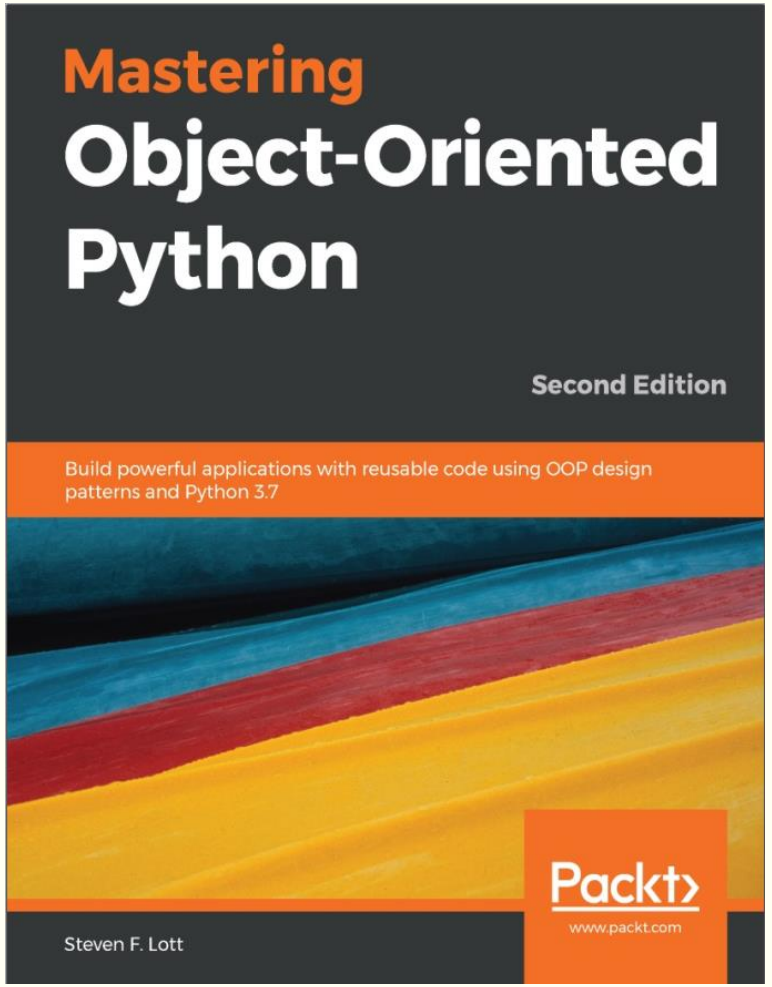


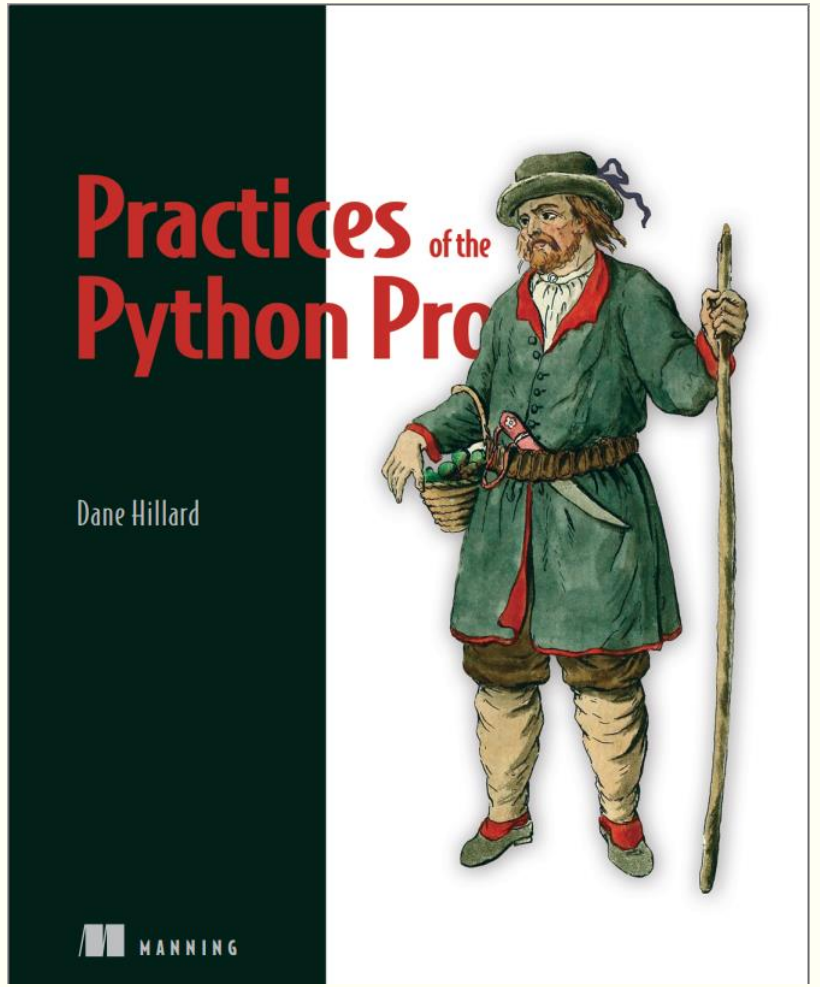
План лекції

- **Принципи об’єктно-орієнтованого програмування.**
- **Об’єкти в мові Python.**
- **Схожість об’єктів: наслідування та поліморфізм.**
- **Об’єктно-орієнтоване програмування в дії.**
- **Структури даних та об'єктно-орієнтоване програмування.**

Рекомендована література







ПРИНЦИПИ ОБ’ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Питання 8.1

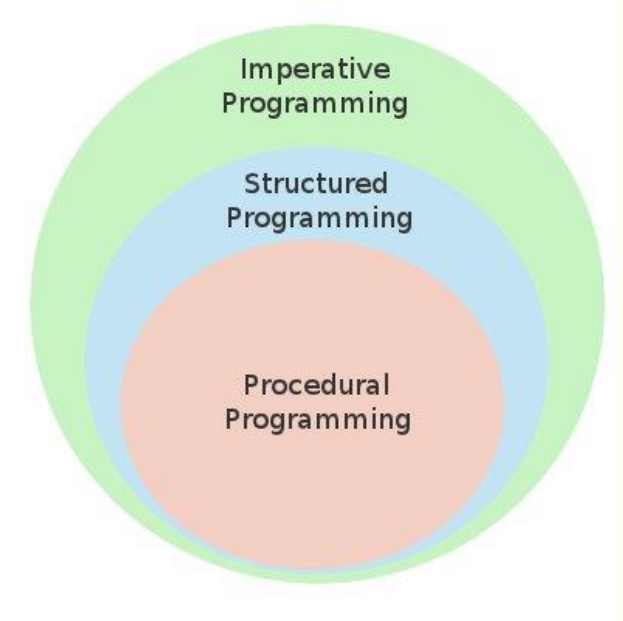
Парадигми програмування

- ***Парадигма програмування*** — сукупність ідей та понять, що визначають стиль написання програм.
- ***Імперативне програмування***: управляючий потік у вигляді послідовності команд, в основному визначається на контрасті з декларативним програмуванням.
- ***Декларативне програмування***: програма описує бажаний результат, а не те, як його отримати.
  - Виконуюче середовище оптимальними способами досягає бажаного результату (SQL та електронні таблиці є декларативними програмними середовищами).

Парадигми програмування

Структурне та процедурне програмування

- ***Процедурне програмування***.
  - Програми виконуються інструкція за інструкцією, зчитуючи/записуючи спільну пам'ять.
  - Стиль програмування тісно пов'язаний з послідовними процесорами та RAM-пам'яттю.
  - Мови: Pascal, Fortran , COBOL
- ***Структурне програмування***: програми мають чисті, goto-free, вкладені (nested) структури керування.
  - З'явилось у відповідь на "goto hell" спагетті-код.



Парадигми програмування

- ***Функціональне програмування***: обчислення відбуваються за рахунок (вкладених) викликів функцій, які уникають глобальної зміни стану, разом з функціональною композицією.
- ***Об'єктно-орієнтоване програмування***: обчислення здійснюються за рахунок обміну повідомленнями (messages) між об'єктами; об'єкти інкапсулюють стан та поведінку.
- ***Подійно-орієнтоване програмування***: управляючий потік визначається асинхронними діями у відповідь на події (від людей, датчиків чи інших обчислень).
- ***Логічне програмування:*** програміст задає набір фактів та правил, а рушій відповідає по ним на питання.

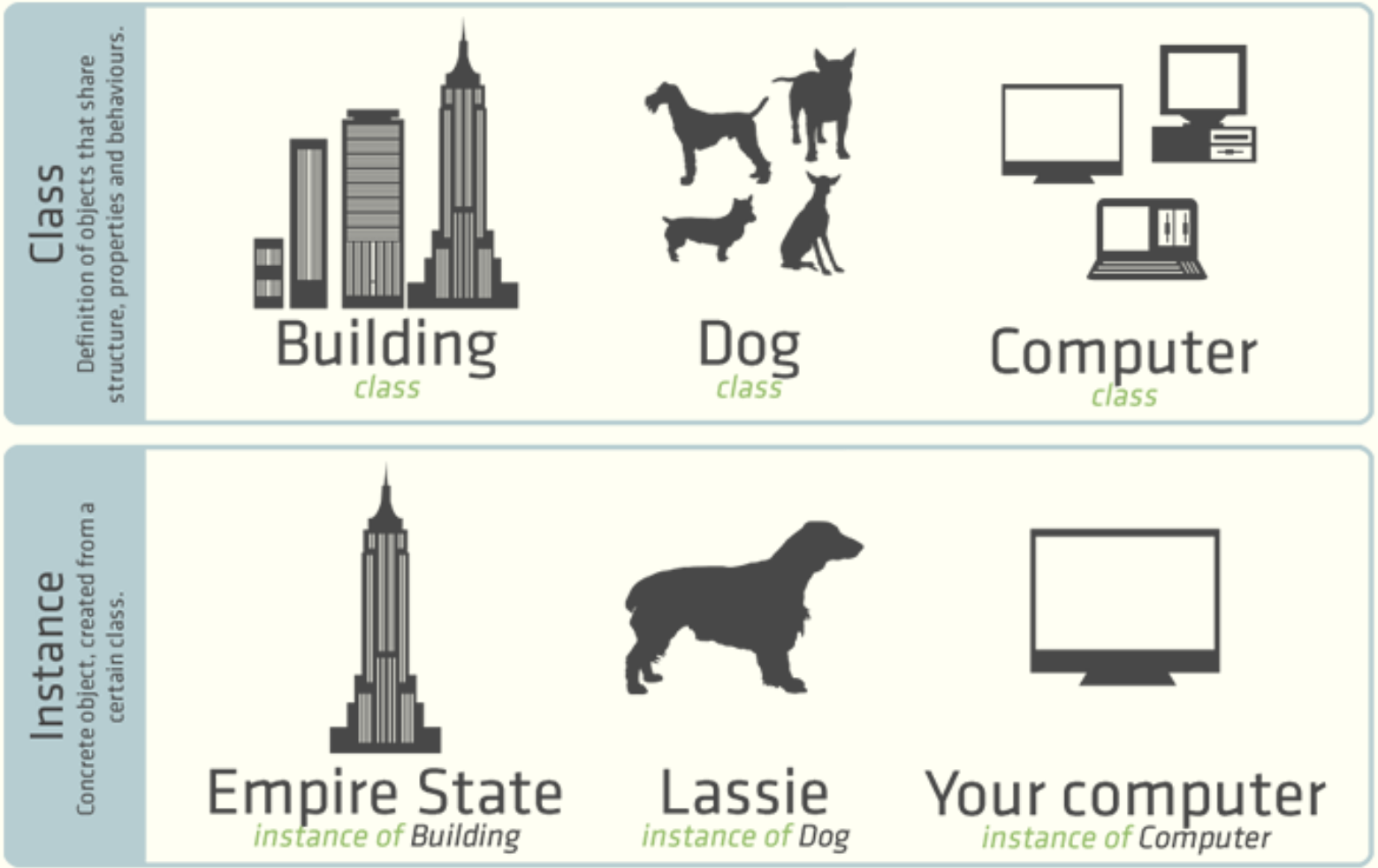
Мотивація вибору парадигми

- **Управляючий потік (Control flow):** як програма виконується всередині (послідовність та розгалуження, багатопоточне виконання, декларативно, реактивно тощо)
- **Організація коду:** як код організується в ієрархію програмних одиниць (units – вирази (expressions), функції, модулі, пакети).
- **Продуктивність (Performance):** як код можна швидко запустити, використовуючи якомога менше ресурсів (RAM, диск, мережа), з кращою поведінкою (респонсивність, масштабованість) підчас виконання.
- **Зв'язування (Coupling) та повторне використання (Reuse) коду:** наскільки просто один і той же код повторно використовувати в різних контекстах
- **Тестування:** наскільки просто код тестувати та верифікувати (verify – перевіряти на коректність).
- **Синтаксис:** наскільки природним, лаконічним, доступним для розуміння є вираз у коді обраною мовою програмування. Можливість розширення синтаксису мови програмістом.
- **Домен (Domain):** до якого application domain парадигма найкраще застосовується (бекенд на стороні сервера, БД, GUI-фронтенд, системи керування).

Строге визначення ООП

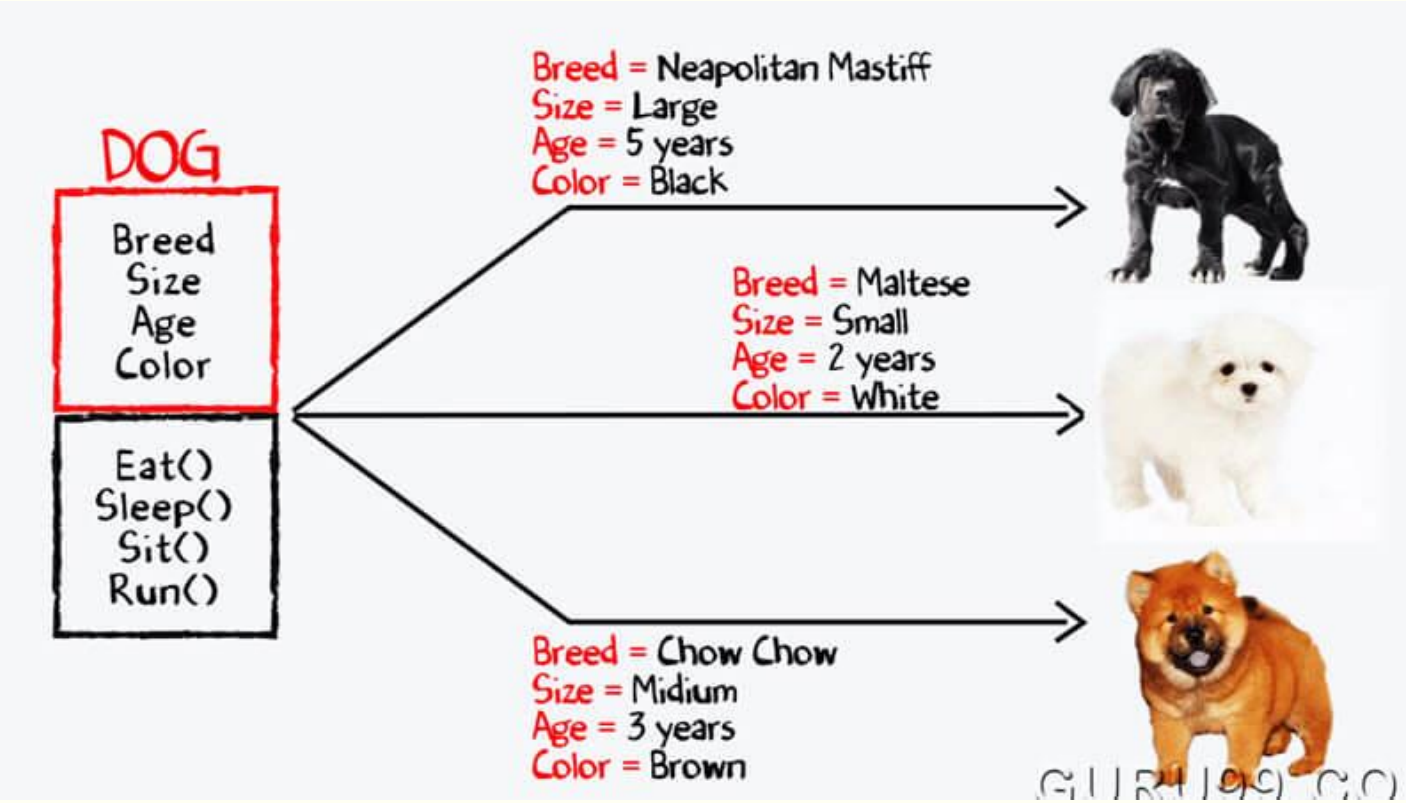
- Алан Кей сформулював наступні принципи ООП:
  - Все є об’єктом.
  - Обчислення здійснюються шляхом ***взаємодії (обміну даними) між об’єктами***.
    - Один об’єкт вимагає, щоб інший об’єкт виконав певну дію.
  - Кожен об’єкт має ***незалежну пам’ять***, яка складається з інших об’єктів.
  - Кожен ***об’єкт є представником (екземпляром) класу***, який виражає загальні властивості об’єктів.
  - У класі задається поведінка (функціональність) об’єкта.
  - Класи організовані в єдину деревовидну структуру із загальним коренем, яку називають ***ієрархією наслідування***.

Класи та об’єкти

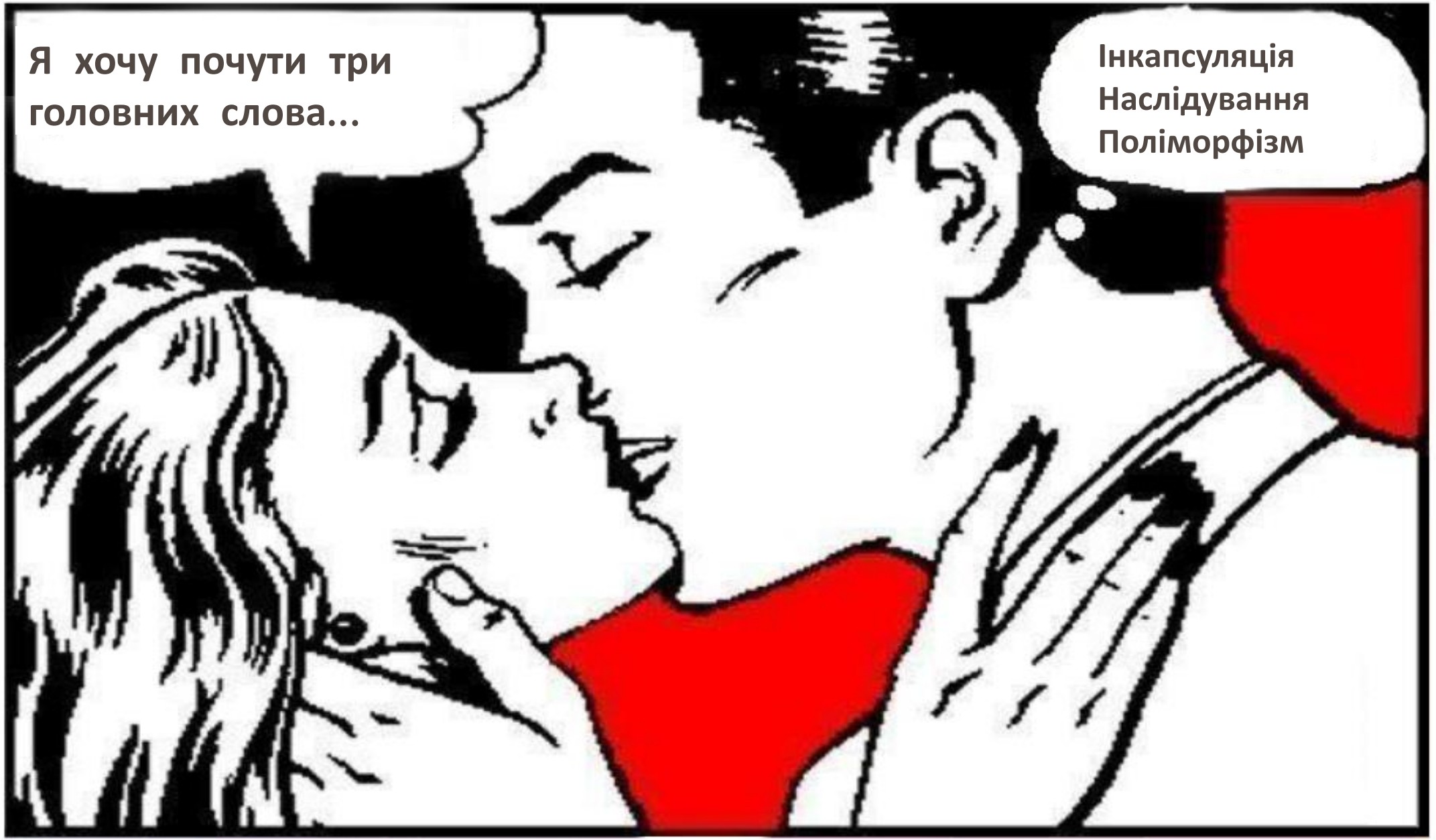


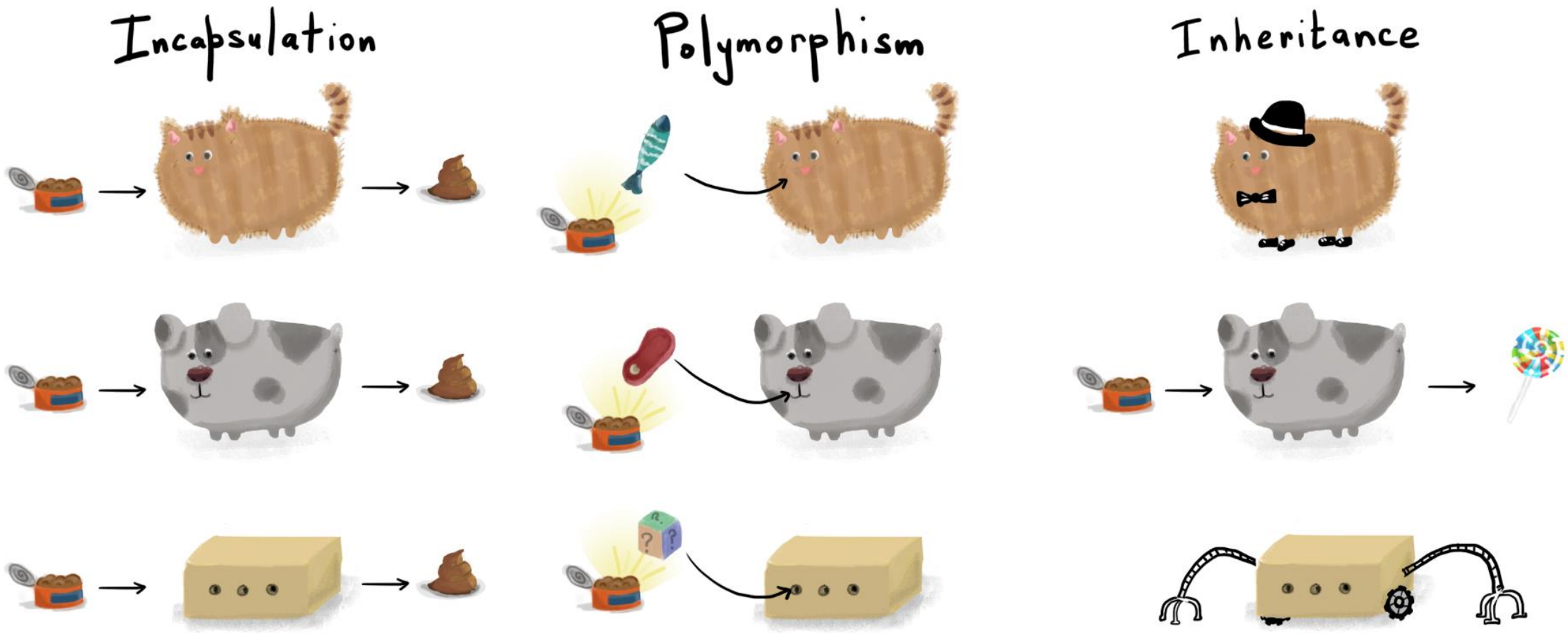
- ***Клас*** – це деякий шаблон, за яким конструюється ***об’єкт*** (екземпляр класу).
  - Клас визначає *тип* для свого екземпляру

Класи та об’єкти

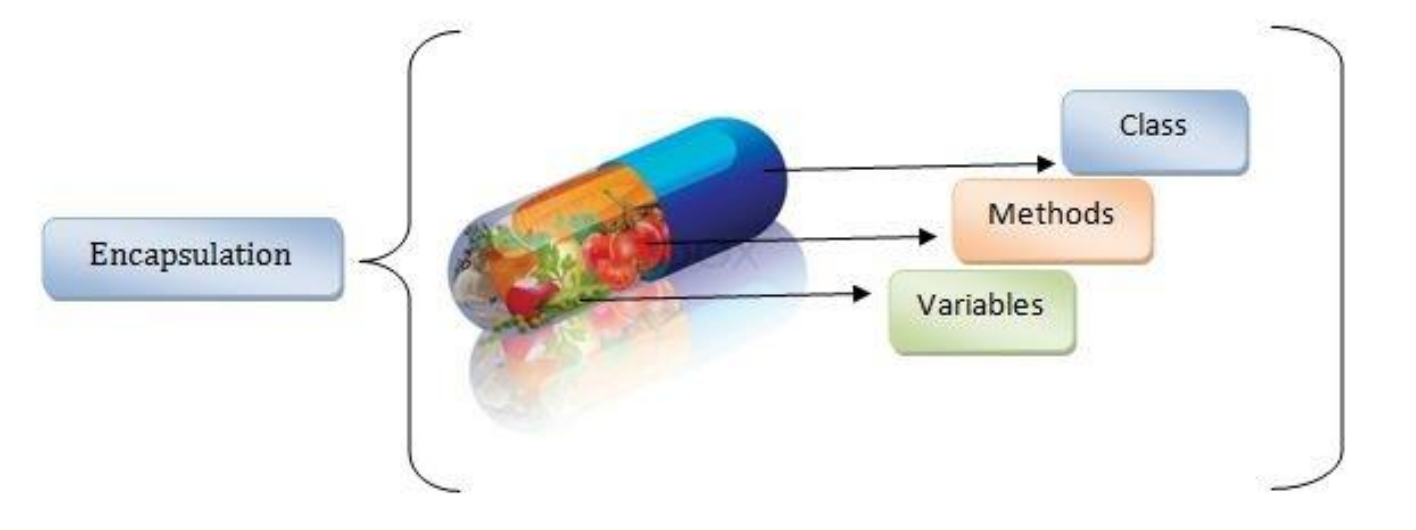


- Клас складається з атрибутів класу:
  - поля
  - методи
- Будь-яке значення, яке характеризує об’єкт, називають ***полем***.
  - об’єкт.поле
- Операції з даними можна виконувати за допомогою функцій, оголошених всередині класу, - ***методів***.
  - Їх можна розглядати як атрибути, що викликаються (callable attribute)


Я хочу почути три головних слова...
Інкапсуляція
Наслідування
Поліморфізм


Incapsulation
Polymorphism
Inheritance
every animal eats
and then poop
each animal can eat
its own type of food
you can create new type of animal
changing or adding properties

1. Інкапсуляція



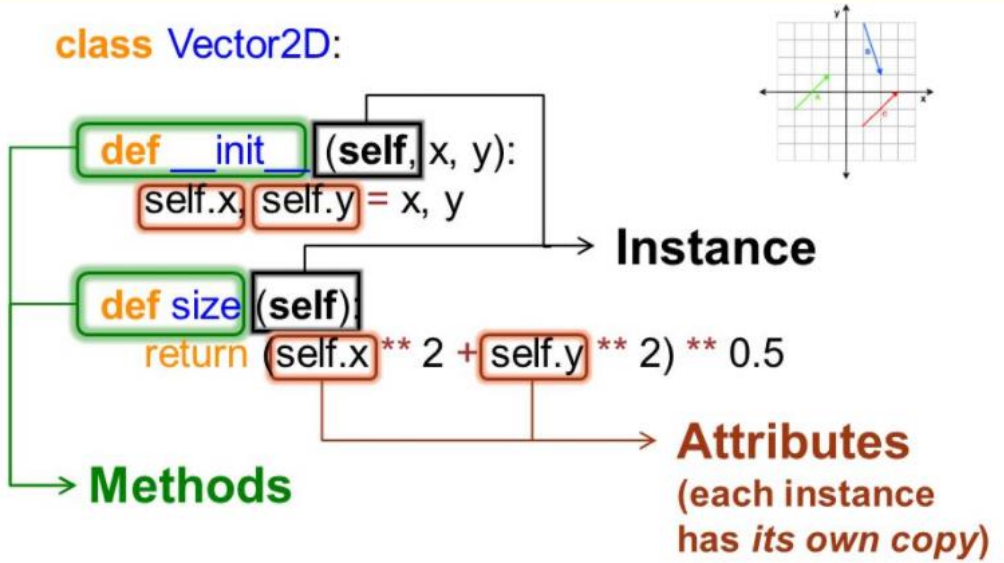
- Клас інкапсулює змінні/поля та функції/методи.
  - Загалом їх називають ***атрибутами класу.***

Student
name grade
getName() printGrade()

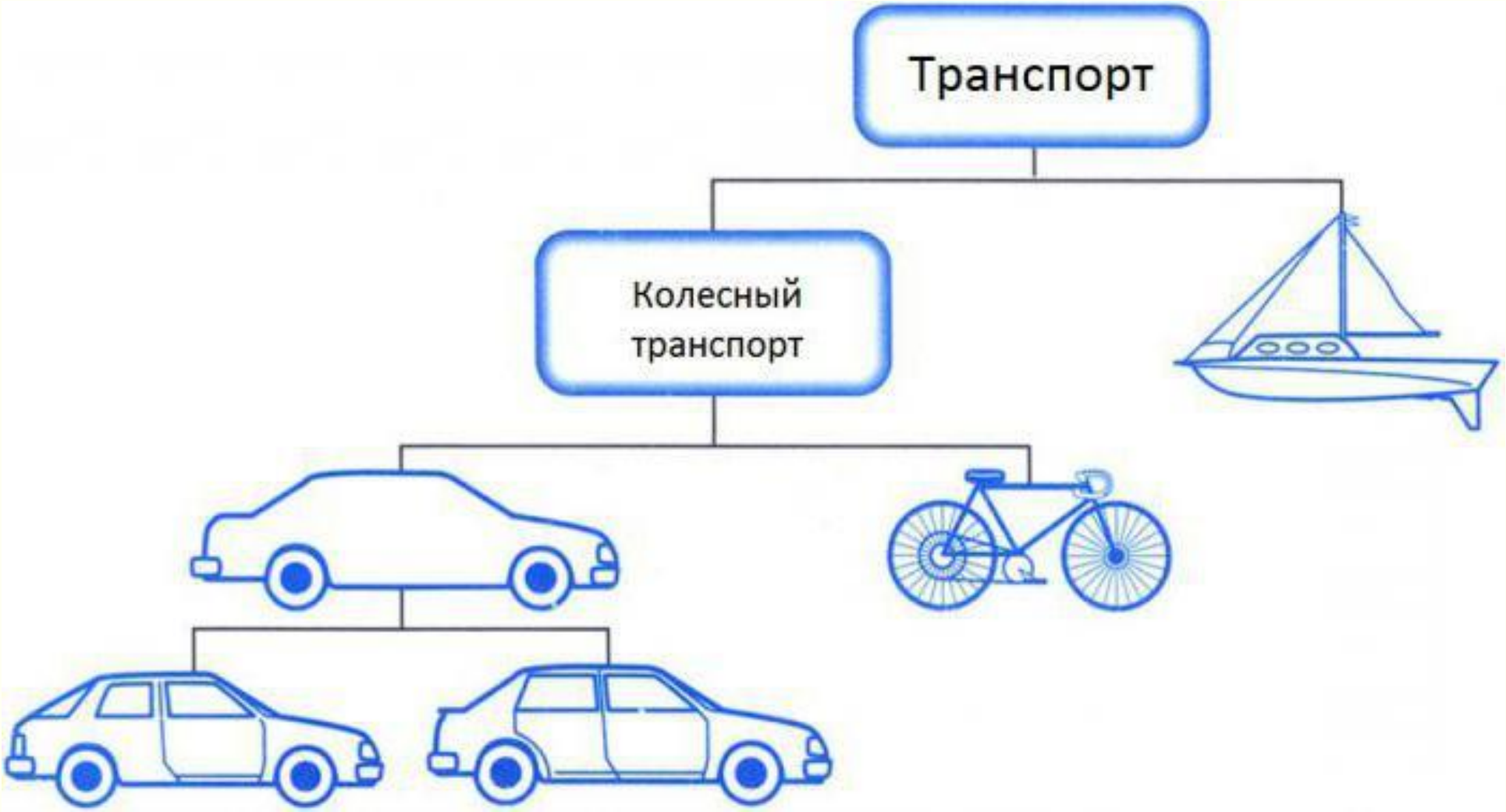
Circle
radius color
getRadius() getArea()

SoccerPlayer
name number xLocation yLocation
run() jump() kickBall()

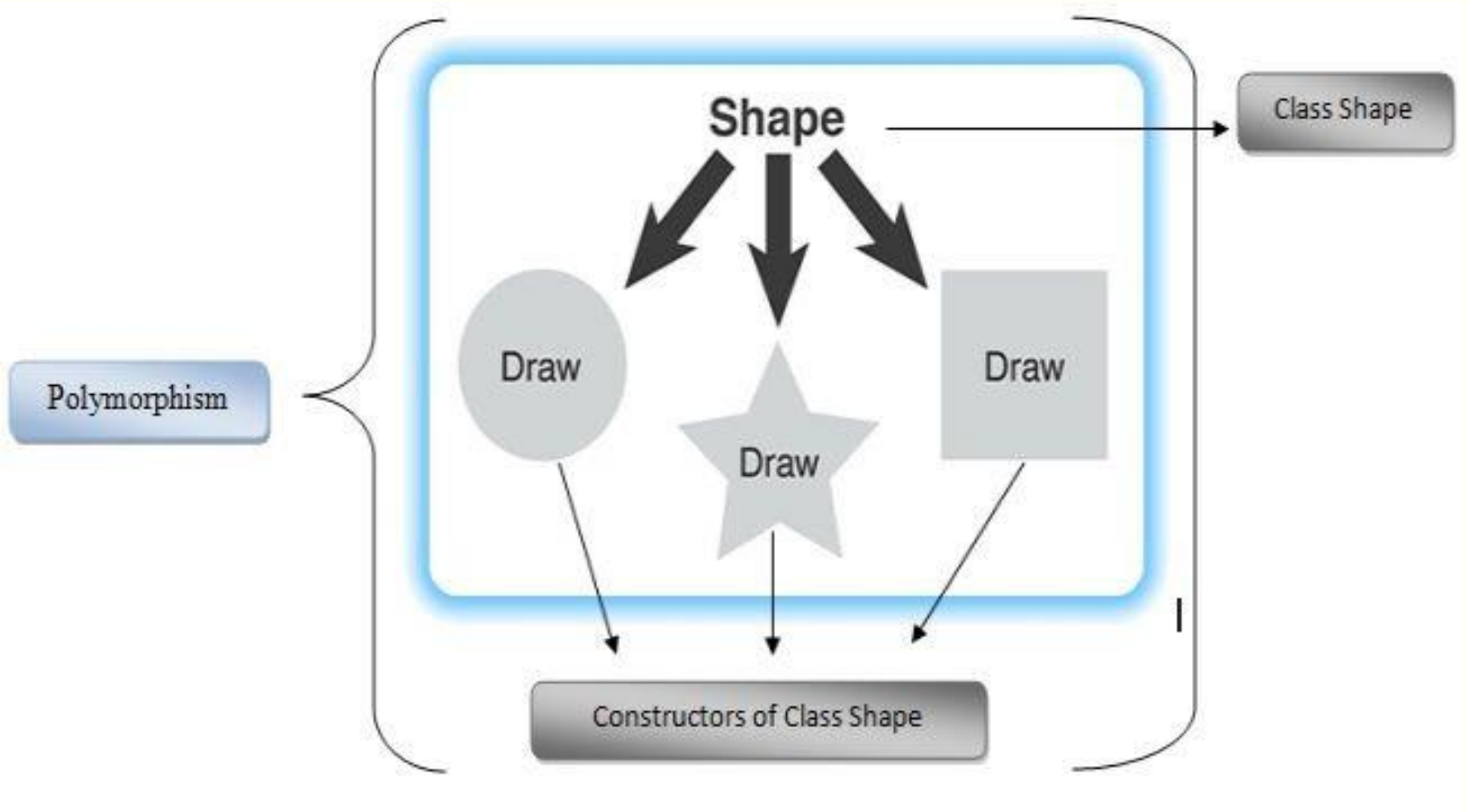
Car
plateNumber xLocation yLocation speed
move() park() accelerate()



2. Наслідування



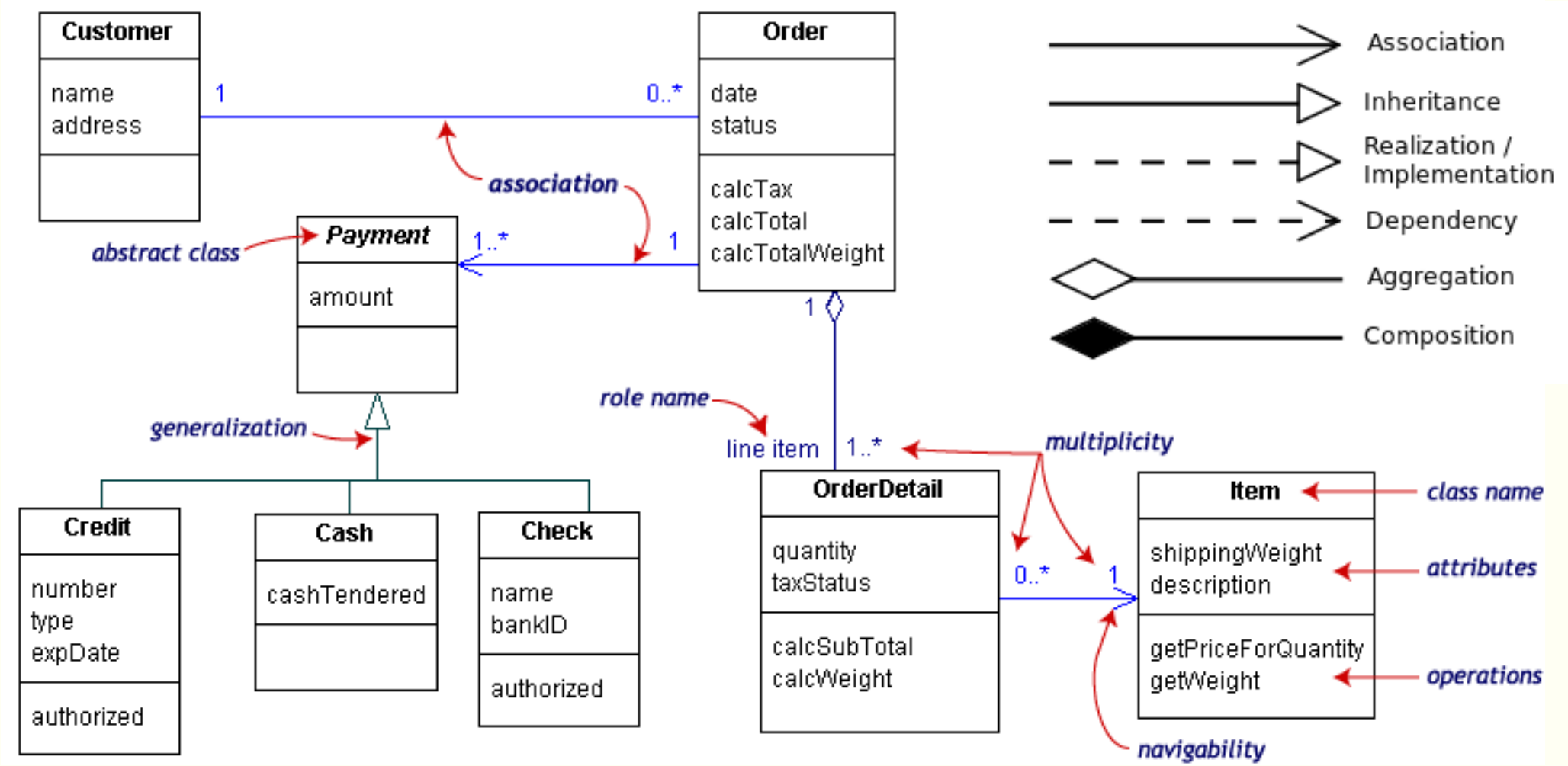
3. Поліморфізм



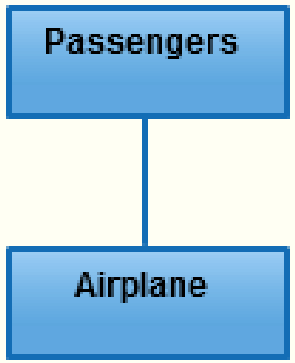
4. Абстракція

- Під абстракцією розуміємо представлення тільки базових властивостей об’єктів реального світу в програмних об’єктах з приховуванням деталей реалізації.

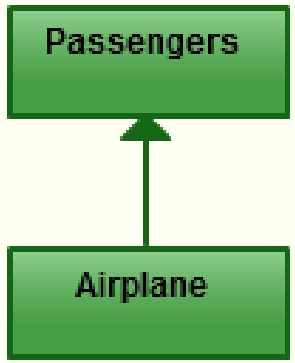
Зв'язки між класами



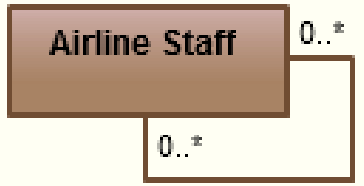
Зв'язки між класами



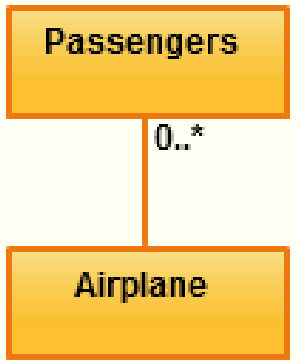
Association



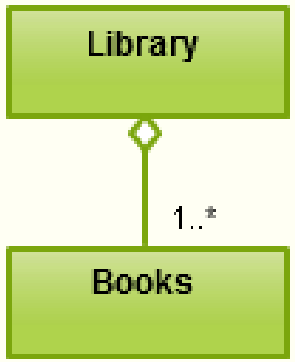
Directed Asscoation



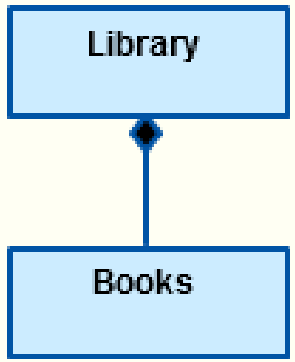
Reflexive Assciation



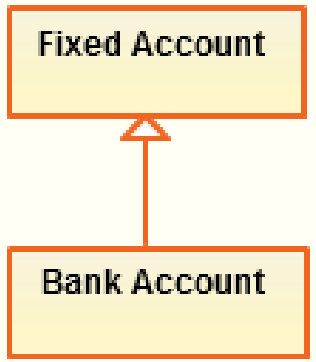
Multiplicity



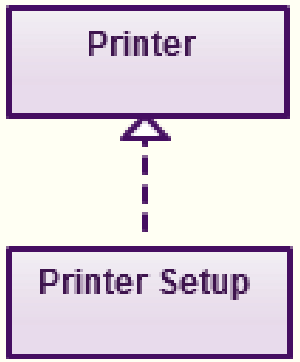
Aggregation



Composition

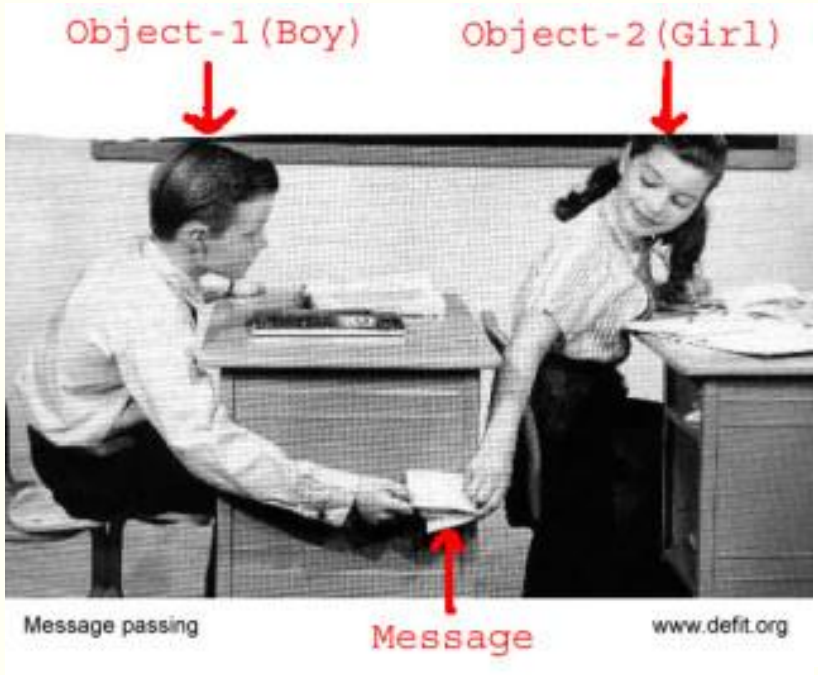


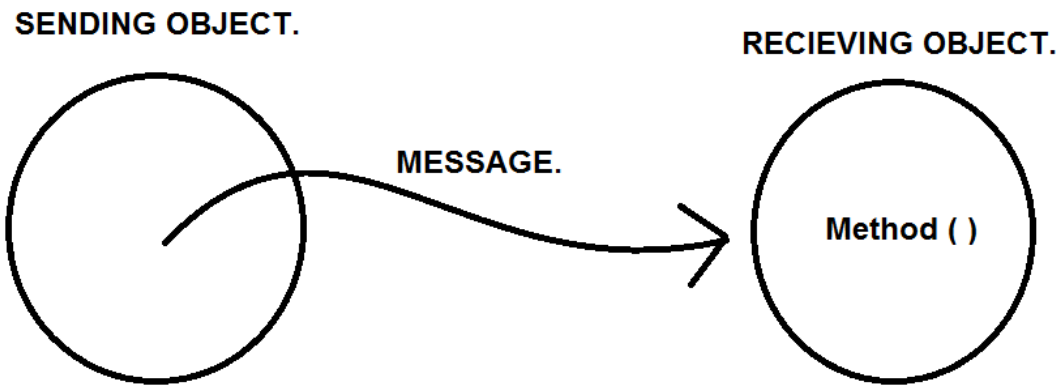
Inheritance



Realization

5. Передача повідомлень (message passing)





Triggering a Method via Message.

- Передача повідомлень спрощує комунікацію об’єктів між собою.
  - Повідомлення – це запит на виконання процедури об’єкта, якому це повідомлення надіслане.

ДЯКУЮ ЗА УВАГУ!

Наступне питання: Об’єкти в мові Python