

Class



TH.S TRẦN ĐỨC LỢI
PYTHONVIETNAM.INFO

Ôn tập bài cũ



- Ôn lại nội dung đã học về **File I/O**

Mục đích bài học

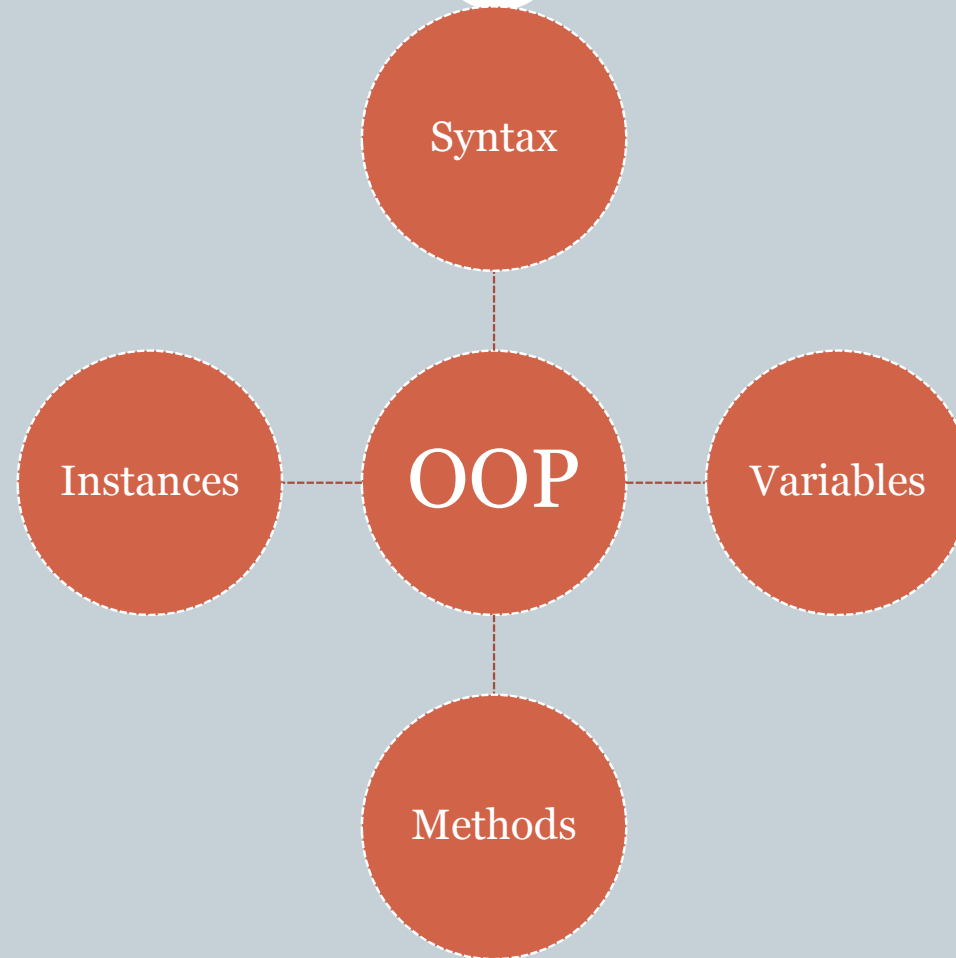


- Tìm hiểu về Class trong python

Class



- Một mẫu do người dùng tự định nghĩa với một tập hợp các thuộc tính (attributes)
- Attributes = data members (variables) + methods
- Truy nhập thuộc tính thông qua dấu “.”



Class: Sample



- Dùng từ khóa class để khai báo
- **class** *classname*:
 - *“Docstring for class”*
 - Attributes
- `Classname.__doc__`

Class: Class variables vs Instance variables



- **Class variables:**
 - Dùng chung bởi các instance của class
 - Định nghĩa ở trong class nhưng ngoài các method
- **Instance variables:**
 - Khác nhau giữa các instance
 - Được định nghĩa trong methods

Class: Method



- Function được khai báo trong khai báo của lớp
- **class** *classname*:
 - *”Docstring for class”*
 - Def method1(self):
 - ✦ Method content

Class: Instance



- Một đối tượng của một class
- `Person = classname()`
- Person được coi là một instance của classname

Class: Ví dụ



- **Class People:**
 - “docstrings here”
 - `Pcount = 0`
 - `Def __init__(self, name):`
 - ✦ `Self.name = name`
 - ✦ `People.pcount += 1`
 - `Def getpopulation(self):`
 - ✦ `Print People.pcount`

Class: `__init__()`



- Một loại method đặc biệt
- Initialization method (# constructor)
- Được gọi khi khởi tạo một instance mới
- Các thuộc tính của đối tượng được **khai báo** trong `__init__()`
- Ví dụ:
 - `Loi = People("Tran Duc Loi")`

Class: self



- Tất cả các method được khai báo trong class đều có self
- Khi gọi một method python sẽ không cần khai báo
- Xem lại ví dụ.

Class: Truy nhập thuộc tính



- Sử dụng “.” để truy nhập thuộc tính
- Ví dụ:
 - `Person.Getpopulation()`
 - `Person.name`
- `Person` là instance
- Truy nhập class variable vs instance variable?

Class

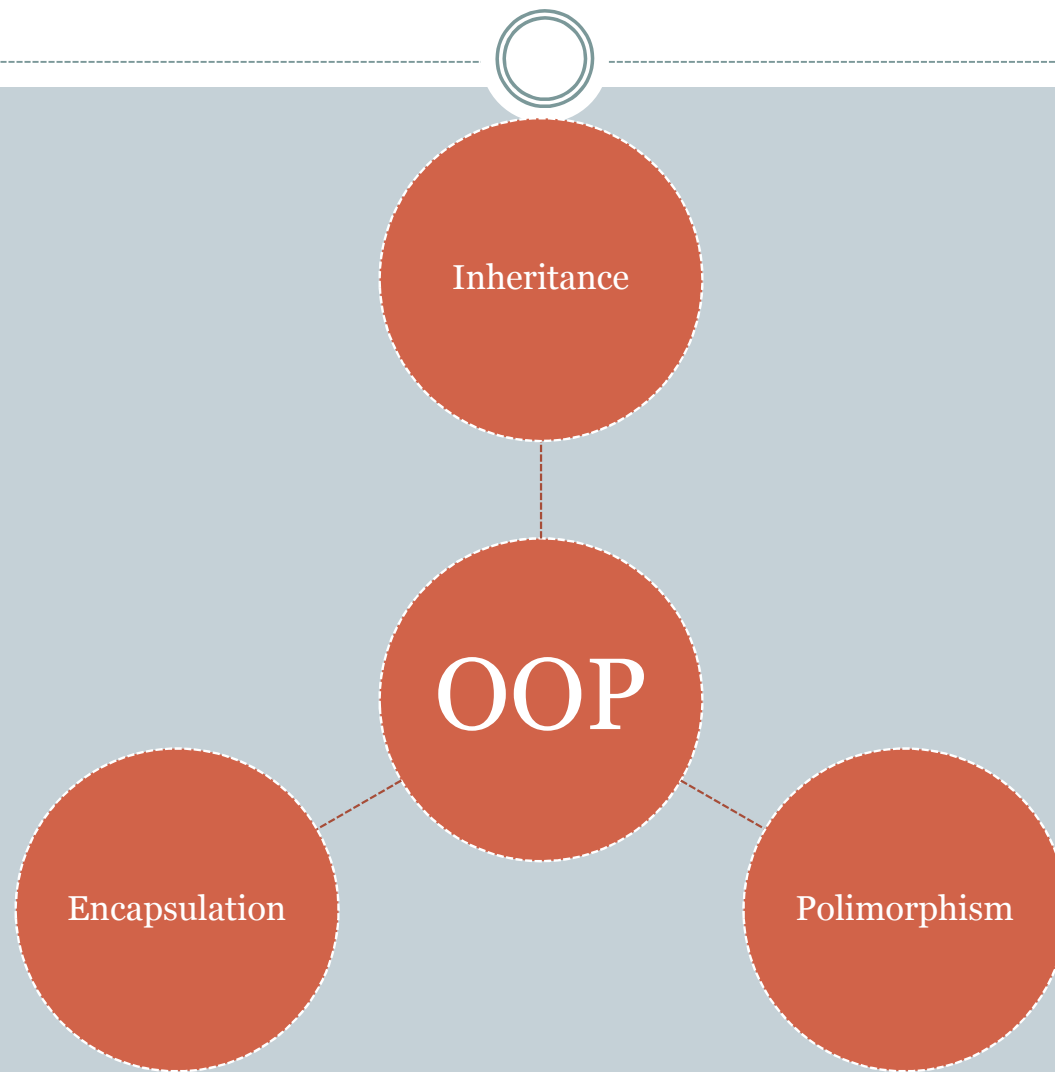


- `Hasattr(object, name)`
- `Getattr(object, name, default)`
- `Setattr(object, name, value)`
- `Delattr(object, name)`

Class: `__del__()`



- Xóa các đối tượng không còn cần thiết
- Python tự động thực hiện garbage collection
- `__del__()`
- Ví dụ:
- Class People:
 - `"""docstrings here"""`
 - `Pcount = 0`
 - `Def __init__(self, name):`
 - `Def __del__(self):`



Class: Inheritance



- Tính kế thừa trong hướng đối tượng
- Ví dụ:
 - `Class Children(Parent1, Parent2, ...):`
 - ✦ `"""docstring"""`
- Class Children sẽ được kế thừa các thuộc tính trong các class Parent1, ...
- Class Children có thể ghi đè các thuộc tính của class Parent1, ...

Class: Inheritance



- Ví dụ:
- Class Parent(**object**):
 - Age = 100
 - Def __init__(self):
 - Def setattr(self, value):
 - Def getattr(self):
- Class Child(Parent):
 - Def __init__(self):
 - ✦ Super(Child, self).__init__()
 - Def childget(self):

Class: Polymorphism



- Overriding methods
- Cho các phương thức khác nhau ở các lớp con khác nhau
- Hàm `__init__()` là một ví dụ về overriding
- Xem xét ví dụ

Class: Encapsulation



- Hạn chế truy cập vào các thuộc tính của đối tượng
- Public, protected, private?
- `__wheels`
- `Object.__classname__attributename`
- Python đã bảo vệ truy nhập thuộc tính đối tượng như thế nào?