

# Nội dung

48

- Giới thiệu
- Lập trình căn bản
- Các kiểu dữ liệu phức
- **Lập trình hướng đối tượng**
- Vào/ra, ngoại lệ
- Lập trình mạng
- Lập trình Web
- Python-MySQL
- Giao diện GUI
- Vẽ đồ thị

# Lập trình hướng đối tượng

49

- Đối tượng (object)
- Lớp (class)
- Thực thể / thể hiện (instance)
- Trạng thái (state)
- Phương thức (method)
- Truyền thông điệp (message passing)
- Trừu tượng hoá (abstraction)
- Đóng gói (encapsulation)
- Kế thừa (inheritance)
- Đa hình (polymorphism)
- Tổng quát hoá (generalization)
- Cụ thể hoá (specialization)

# Lập trình hướng đối tượng

50

- Định nghĩa lớp

```
class ClassName:  
    'Optional class documentation string'  
    class_suite
```

- class\_suite: các thuộc tính, phương thức (hàm)
- Tham số đầu tiên của phương thức thường được đặt tên là **self** để thỏa:

```
obj.meth(args) = class.meth(obj, args)
```

- **name:** public, **\_name:** protected, **\_\_name:** private

# Định nghĩa lớp

51

```
>>> class Employee:
...     'Common base class for all employees'
...     empCount = 0
...
...     def __init__(self, name, salary):
...         self.name = name
...         self.salary = salary
...         Employee.empCount += 1
...
...     def displayCount(self):
...         print "Total Employee %d" % Employee.empCount
...
...     def displayEmployee(self):
...         print "Name: ", self.name, ", Salary: ", self.salary
...
>>>
```

# Tạo và sử dụng đối tượng

52

```
>>> emp1 = Employee("Toto", 2000)
>>> emp2 = Employee("Tutu", 5000)
>>> emp1.displayEmployee()
Name:  Toto , Salary:  2000
>>> emp2.displayEmployee()
Name:  Tutu , Salary:  5000
>>> print "Total Employee: %d" %Employee.empCount
Total Employee: 2
>>> emp3 = emp2
>>> emp3.displayEmployee()
Name:  Tutu , Salary:  5000
>>> print "Total Employee: %d" %Employee.empCount
Total Employee: 2
```

# Truy xuất thuộc tính

53

```
>>> emp1.age = 7
>>> emp1.age
7
>>> del emp1.age
>>> emp1.age
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: Employee instance has no attribute 'age'
>>> hasattr(emp1, 'age')
False
>>> setattr(emp1, 'age', 8)
>>> getattr(emp1, 'age')
8
>>> delattr(emp1, 'age')
>>> emp1.age
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: Employee instance has no attribute 'age'
```

# Thuộc tính định nghĩa sẵn

54

```
>>> print "Employee.__doc__:", Employee.__doc__
Employee.__doc__: Common base class for all employees
>>> print "Employee.__name__:", Employee.__name__
Employee.__name__: Employee
>>> print "Employee.__module__:", Employee.__module__
Employee.__module__: __main__
>>> print "Employee.__bases__:", Employee.__bases__
Employee.__bases__: ()
>>> print "Employee.__dict__:", Employee.__dict__
Employee.__dict__: {'__module__': '__main__', 'displayCount': <function displayC
ount at 0x7fe91256c500>, 'empCount': 2, 'displayEmployee': <function displayEmpl
oyee at 0x7fe91256c758>, '__doc__': 'Common base class for all employees', '__in
it__': <function __init__ at 0x7fe91256c668>}
```

# Xóa đối tượng

55

```
>>> class Point:
...     def __init__( self, x=0, y=0):
...         self.x = x
...         self.y = y
...     def __del__(self):
...         class_name = self.__class__.__name__
...         print class_name, "destroyed"
...
>>> pt1 = Point()
>>> pt2 = pt1
>>> pt3 = pt1
>>> print id(pt1), id(pt2), id(pt3)
140639011813424 140639011813424 140639011813424
>>> del pt1
>>> del pt2
>>> del pt3
Point destroyed
```



# Kế thừa

56

- Định nghĩa lớp kế thừa

```
class SubClassName (ParentClass1[, ParentClass2, ...]):  
    'Optional class documentation string'  
    class_suite
```

- Hàm `issubclass(sub, sup)`, `isinstance(obj, Class)`,
- Hàm `super()`

# Kế thừa (đa hình)

57

```
>>> class Animal:
...     'Common base class for animals'
...
...     def __init__(self, name='no-name'):
...         self.name = name
...
...     def say(self):
...         print "%s can't say" %self.name
...
>>> class Bird(Animal):
...     def say(self):
...         print "%s twitter" %self.name
...
>>> class Cat(Animal):
...     def say(self):
...         print "%s meow" %self.name
...
>>>
```

# Kế thừa (đa hình)

58

```
>>> a = Animal()
>>> a.say()
no-name can't say
>>> a = Animal('Toto')
>>> a.say()
Toto can't say
>>> b = Bird('Flappy')
>>> b.say()
Flappy twitter
>>> c = Cat('Kitty')
>>> c.say()
Kitty meow
>>>
```

# Đa kế thừa

59

```
>>> class Owl(Bird,Cat):
...     pass
...
>>> o = Owl('Chic')
>>> o.say()
Chic twitter
>>>
>>> class OwlX(Bird,Cat):
...     "extra Owl"
...     def say(self):
...         print "owl..."
...         Cat.say(self)
...
>>> ox =OwlX('ChicX')
>>> ox.say()
owl...
ChicX meow
```

# Định nghĩa chồng phép toán

60

OPERATOR	FUNCTION	METHOD DESCRIPTION
+	<code>__add__(self, other)</code>	Addition
*	<code>__mul__(self, other)</code>	Multiplication
-	<code>__sub__(self, other)</code>	Subtraction
%	<code>__mod__(self, other)</code>	Remainder
/	<code>__truediv__(self, other)</code>	Division
<	<code>__lt__(self, other)</code>	Less than
<=	<code>__le__(self, other)</code>	Less than or equal to
==	<code>__eq__(self, other)</code>	Equal to
!=	<code>__ne__(self, other)</code>	Not equal to
>	<code>__gt__(self, other)</code>	Greater than
>=	<code>__ge__(self, other)</code>	Greater than or equal to
[index]	<code>__getitem__(self, index)</code>	Index operator
in	<code>__contains__(self, value)</code>	Check membership
len	<code>__len__(self)</code>	The number of elements
str	<code>__str__(self)</code>	The string representation

# Định nghĩa chồng phép toán

61

```
>>> class Point2D:
...     def __init__(self, x=0, y=0):
...         self.x = x
...         self.y = y
...
...     def __str__(self):
...         return 'Point2D (%f, %f)' %(self.x, self.y)
...
...     def __add__(self, other):
...         return Point2D(self.x + other.x, self.y + other.y)
...
>>> p1 = Point2D(2,10)
>>> p2 = Point2D(5,-2)
>>> print p1 + p2
Point2D (7.000000, 8.000000)
```

# Thuộc tính có tên bắt đầu \_\_ là thuộc tính ẩn bên trong đối tượng

62

```
>>> class JustCounter:
...     __secretCount = 0
...
...     def count(self):
...         self.__secretCount += 1
...         print self.__secretCount
...
>>> counter = JustCounter()
>>> counter.count()
1
>>> counter.count()
2
>>> print counter.__secretCount
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: JustCounter instance has no attribute '__secretCount'
>>>
>>> print counter._JustCounter__secretCount
2
```