

Tài liệu hướng dẫn thực hành Buổi 3

Môn : Lập trình Python

Nội dung chính : Lập trình hướng đối tượng và Xử lý ngoại lệ

1. Lập trình hướng đối tượng

- Đối tượng (Object) : là những thực thể tồn tại có những hành vi nhất định
- Lớp (Class) : là một kiểu dữ liệu đặc biệt do người dùng định nghĩa, tập hợp nhiều thuộc tính đặc trưng cho mọi đối tượng được tạo ra từ lớp đó.
- Thuộc tính là các giá trị của lớp. Sau này khi các đối tượng được tạo ra từ lớp, thì thuộc tính của lớp lúc này sẽ trở thành các đặc điểm của đối tượng đó.

Trong Python, lập trình hướng đối tượng (OOP) tuân theo một số nguyên lý cơ bản:

– **Tính kế thừa (Inheritance)** : cho phép một lớp (class) có thể kế thừa các thuộc tính và phương thức từ các lớp khác đã được định nghĩa.

– Cho phép một lớp kế thừa các thuộc tính và phương thức từ các lớp khác đã được định nghĩa.

– Lớp đã định nghĩa gọi là lớp cha, lớp mới phát sinh (kế thừa) gọi là lớp con

– Lớp con kế thừa các thành phần của lớp cha, đồng thời có thể mở rộng các thành phần kế thừa và bổ sung các thành phần mới

```
1  # Lớp cha
2  class Car:
3
4      # Constructor
5      def __init__(self, hangxe, tenxe, mausac):
6          # Lớp Car có 3 thuộc tính: tenxe, mausac, hang xe
7          self.hangxe = hangxe
8          self.tenxe = tenxe
9          self.mausac = mausac
10
11     # phương thức
12     def chayxe(self):
13         print ("{} đang chạy trên đường".format(self.tenxe))
14
15     def dungxe(self, mucdich):
16         print ("{} đang dừng xe để {}".format(self.tenxe, mucdich))
17     --
```

```

18 # Lớp Toyota mở rộng từ lớp Car.
19 class Toyota(Car):
20
21     def __init__(self, hangxe, tenxe, mausac, nguyenvlieu):
22         # Gọi tới constructor của lớp cha (Car)
23         # để gán giá trị vào thuộc tính của lớp cha.
24         super().__init__(hangxe, tenxe, mausac)
25
26         self.nguyenvlieu = nguyenvlieu
27
28         # Kế thừa phương thức cũ
29         def chayxe(self):
30             print("{} đang chạy trên đường".format(self.tenxe))
31
32         # Ghi đè (override) phương thức cùng tên của lớp cha.
33         def dungxe(self, mucdich):
34             print("{} đang dừng xe để {}".format(self.tenxe, mucdich))
35             print("{} chạy bằng {}".format(self.tenxe, self.nguyenvlieu))
36
37         # Bổ sung thêm thành phần mới
38         def nomay(self):
39             print("{} đang nổ máy".format(self.tenxe))

```

– **Tính đa hình (Polymorphism)** : là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau.

– Đây là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau

– Ví dụ : Lớp Car(), Ship() và Human() đều có một phương thức là Move(). Nhưng cách thực hiện phương thức Move cho mỗi class sẽ khác nhau.

```

3 class Toyota:
4
5     def dungxe(self):
6         print("Toyota dừng xe để nạp điện")
7
8     def nomay(self):
9         print("Toyota nổ máy bằng hộp số tự động")
10
11 class Porsche:
12
13     def dungxe(self):
14         print("Porsche dừng xe để bơm xăng")
15
16     def nomay(self):
17         print("Porsche nổ máy bằng hộp số cơ")
18
19 # common interface
20 def kiểmtra_dungxe(car): car.dungxe()

```

– **Tính đóng gói (Encapsulation)** : là quy tắc yêu cầu trạng thái bên trong của một đối tượng được bảo vệ và tránh truy cập được từ code bên ngoài

– Ta có thể hạn chế quyền truy cập vào các thuộc tính và trạng thái bên trong của đối tượng. Điều này giúp ngăn chặn việc dữ liệu bị sửa đổi từ các tác nhân bên ngoài.

-Trong python ta khai báo thuộc tính private bằng cách sử dụng dấu gạch dưới làm tiền tố `__` hoặc `__`

```
3 class Computer:
4
5     def __init__(self):
6         # Thuộc tính private ngăn chặn sửa đổi trực tiếp
7         self.__maxprice = 900
8
9     def sell(self):
10        print("Giá bán sản phẩm: {}".format(self.__maxprice))
11
12    def setMaxPrice(self, price):
13        self.__maxprice = price
```

```
15 c = Computer()
16 c.sell()
17
18 # thay đổi giá bán ( không thành công)
19 c.__maxprice = 1000
20 c.sell()
21
22 # sử dụng hàm setter để thay đổi giá. ( thành công)
23 c.setMaxPrice(1000)
24 c.sell()
```

2. Xử lý ngoại lệ

Ngoại lệ (Exception) là lỗi xảy ra trong quá trình thực thi một chương trình. Khi nó xảy ra, Python tạo ra một exception để xử lý vấn đề đó tránh cho ứng dụng hay server bị crash.

Ngoại lệ có thể là bất kỳ điều kiện bất thường nào trong chương trình phá vỡ luồng thực thi chương trình đó. Bất cứ khi nào một ngoại lệ xuất hiện, chương trình sẽ ngừng thực thi, chuyển qua quá trình gọi và in ra lỗi đến khi nó được xử lý.

Trong python, các ngoại lệ được xử lý bằng khối lệnh try except....

try:

 # lệnh...

except:

 # lệnh...

Phần thân của try sẽ gồm code có thể tạo ra exception, nếu một exception được tạo ra, tất cả câu lệnh trong khối sẽ bị bỏ qua.

Mặt khác, phần thân của except được gọi bởi exception handler, vì nó được dùng để bắt lỗi. Khối except sẽ thực hiện khi lỗi được tạo ra, không thì sẽ được bỏ qua.

```

import sys

randomList = ['a', 0, 2.0]

for i in randomList:
    try:
        print("Iterative :", i)
        r = (1.0/(i))
        break
    except:
        print("Exception: ", sys.exc_info()[0], ".")
        print("Next Iterative : ")
        print()

print("Result with iterative ", i, " is:", r)

```

Try Finally : đây là một cách khác để sử dụng try trong python.

- Khối lệnh trong block Finally sẽ luôn được thực thi dù chương trình có lỗi hay không.
- Thường thì những lệnh trong Finally sẽ dùng để giải phóng tài nguyên

Định nghĩa một exception : Ta cần kế thừa lớp Exception của Python.

Sau đó ta dùng raise để gọi exception vừa định nghĩa

```

class MyPersonalException(Exception):
    def __init__(self, msg):
        print("Error : " + msg)

```

3. Bài tập

Sinh viên tạo file <<Hoten>>_THBuoai3.py (vd : tranvanan_THBuoai3.py), Nhớ chú thích họ tên và MSSV lên đầu file.

Câu 1 : Viết chương trình khởi tạo lớp human gồm có những thông tin sau : họ tên, năm sinh, quê quán, nghề nghiệp. Viết phương thức :

- live(self, noicutru) : in ra màn hình người đó tên gì đang sống ở đâu
- work(self, diachicoquan) : in ra màn hình người đó tên gì đang làm nghề gì tại cơ quan nào (địa chỉ cơ quan)

Câu 2 : Viết chương trình cho phép người dùng nhập vào số lượng người muốn nhập vào hệ thống. Tạo ra số lượng đối tượng thuộc lớp human tương ứng. Sau đó tiến hành nhập thông tin, từng người (họ tên, năm sinh,) và lưu trong các biến thuộc lớp human.

Ví dụ :

```

Nhập số lượng người : 2
Nhập họ tên người thứ 1 : .....
Nhập năm sinh người thứ 1 : ...
Nhập quê quán người thứ 1: ...
Nhập nghề nghiệp người thứ 1: ...
Nhập họ tên người thứ 2: ...
...

```

Sau đó in thông tin của các người dùng đó ra màn hình và sử dụng hàm work() để cho biết thông tin về nơi làm việc của những người đó với diachicoquan của từng người do sinh viên tự nhập vào.

Câu 3 : Khai báo lớp Student là lớp con được kế thừa từ lớp Human với các thuộc tính được kế thừa từ lớp cha (thuộc tính nghề nghiệp được set mặc định thành “student”). Ngoài ra lớp student còn có các thuộc tính riêng là MSSV, ngành học và điểm trung bình (điểm trung bình sử dụng thang điểm 10). Các phương thức :

- Study(self, class) : cho biết sinh viên {tên} có {mssv} thuộc {ngành} đang học tại phòng học {class} nào

- Rank(self) : Cho biết thứ hạng của sv dựa vào điểm trung bình (ĐTB).

 - ĐTB < 4 : Loại Yếu

 - 4 <= ĐTB < 6 : Loại Trung bình

 - 6 <= ĐTB < 8 : Loại Khá

 - 8 <= ĐTB < 10 : Loại Giỏi

- In ra màn hình : Sinh viên {tên} có {MSSV} với điểm trung bình { ĐTB } được xếp loại {loại}

Câu 4 : Khai báo thông tin 2 sinh viên như sau :

- SV1 : Tên : Lê Văn An, sinh năm 2005, quê Vĩnh Long, nghề nghiệp mặc định là sinh viên, MSSV = 12345, ngành CNTT, ĐTB = 7.6

- SV2 : Tên : Trần Văn Bình, sinh năm 2007, quê Trà Vinh, MSSV = 56789, Ngành Tài chính ngân hàng, ĐTB = 4.5

Hãy in thông tin 2 sinh viên này ra màn hình và cho biết thứ hạng của 2 sinh viên này dựa trên điểm trung bình của họ.

Câu 5 : Viết Hàm cho phép người dùng nhập vào thông tin của **MỘT** sinh viên với những điều kiện như sau :

- Họ tên có độ dài không quá 25 ký tự, Nếu dài hơn thì hiện ra thông báo “Ho ten khong vuot qua 25 ky tu !” và yêu cầu người dùng nhập lại

- Năm sinh có độ dài đúng 4 ký tự số, Nếu có ký tự không phải số thì yêu cầu người dùng nhập lại. Nếu dài hơn 4 ký tự thì hiển thị thông báo “Nam sinh phai co 4 chu so ” và cũng yêu cầu người dùng nhập lại

- Quê quán có độ dài không quá 50 ký tự, Nếu dài hơn thì hiện ra thông báo “Que quan khong vuot qua 50 ky tu !” và yêu cầu người dùng nhập lại

- Nghề nghiệp mặc định là sinh viên (Khai báo trực tiếp lúc khai báo lớp Sinh viên) nên không cần phải nhập

- MSSV chứa tối đa 5 ký tự số, không chấp nhận các ký tự chữ cái. Nếu số ký tự dài hơn 5 thì yêu cầu người dùng nhập lại. Nếu có ký tự không phải số thì hiển thị thông báo : “MSSV chi chua cac ky tu so ! ” và yêu cầu người dùng nhập lại.

- Ngành học có độ dài không quá 40 ký tự, Nếu dài hơn thì hiện ra thông báo “nganh hoc khong vuot qua 40 ky tu !” và yêu cầu người dùng nhập lại

- Điểm trung bình phải lớn hơn 0 và nhỏ hơn 10. Nếu ĐTB nhỏ hơn 0 hoặc lớn hơn 10 thì yêu cầu người dùng nhập lại

Câu 6 : Viết hàm cho phép người dùng nhập vào số lượng sinh viên cần Nhập thông tin (ví dụ : 3). Sau đó sử dụng **hàm đã viết ở câu 5** để cho người dùng nhập vào thông tin của từng sinh viên. Sau đó in thông tin các sinh viên ra màn hình theo định dạng như sau :

Sinh viên 1

Ho ten :
Nam sinh :
Que quan :
MSSV :
Nganh Hoc :
Diem trung binh :
Xep Loi :

Sinh viên 2

Ho ten :
Nam sinh :
Que quan :
MSSV :
Nganh Hoc :
Diem trung binh :
Xep Loi :

Sinh viên 3

Ho ten :
Nam sinh :
Que quan :
MSSV :
Nganh Hoc :
Diem trung binh :
Xep Loi :

Câu 7 : Viết chương trình khởi tạo một lớp Exchange gồm có các thuộc tính sau : Serial, Date, Amount, ExchangeType. Lớp này có 2 phương thức là PrintExchange(self) và CashUp(self). 2 phương thức này tạm thời chưa thực hiện chức năng gì cả.

Câu 8 : Khai báo một Lớp tên là GoldExchange kế thừa từ lớp Exchange ở câu 7. Ngoài những thuộc tính ở lớp cha ra thì lớp GoldExchange sẽ có thêm các thuộc tính là GoldType và UnitPrice. GoldType chỉ có 3 loại là : “18k”, “24k”, “9999”. Các phương thức được tái định nghĩa như sau :

- CashUp(self) : được tính bằng $\text{Amount} * \text{UnitPrice}$ tương ứng với GoldType
- PrintExchange(self) : In ra màn hình thông tin của giao dịch với định dạng :

Serial – Date – GoldType – UnitPrice – Amount – Cash Up = số tiền

Ví dụ :

134 – 11/11/2020 – 18k – 3.200.000 – 3 – Cash Up = 9.600.000

Câu 9 : Khai báo một Lớp tên là CurrencyExchange kế thừa từ lớp Exchange ở câu 7. Ngoài những thuộc tính của lớp cha ra thì lớp CurrencyExchange có thêm các thuộc tính là CurrencyType, ExchangeRate và Action. CurrencyType chỉ nhận 3 loại tiền tệ là USD, EUR và AUD, ExchangeRate là tỷ giá trao đổi tương ứng với loại tiền tệ và Action tương ứng với hành động mua/bán. Nếu Action = 1 thì ta hiểu là giao dịch mua, ngược lại là giao dịch bán.

Các phương thức của lớp này được khai báo như sau :

- CashUp(self) : Nếu là giao dịch mua thì số tiền sẽ được tính bằng
 $\text{Amount} * \text{ExchangeRate}$

Nếu giao dịch là bán thì số tiền sẽ là

$\text{Amount} * \text{ExchangeRate} * 1.1$

- PrintExchange(self) : In ra màn hình thông tin của giao dịch như sau :

Sell/Buy Exchange : Serial – Date – CurrencyType – ExchangeRate – Amount – Cash Up = số tiền

Ví dụ :

Buy Exchange : 115 – 11/11/2020 – USD – 25.000 – 100 – Cash Up = 2.500.000

Câu 10 : Viết hàm cho phép người dùng nhập vào một giao dịch mà họ muốn với các yêu cầu như sau :

- Nhập mã giao dịch (Serial) : Serial phải là một dãy số, không chấp nhận các ký tự chữ cái, độ dài tối đa 5 ký tự. Nếu người dùng nhập vào dài hơn 5 ký tự hoặc có các ký tự không phải số thì hiển thị thông báo “Serial must be a Number with max length = 5 ” và yêu cầu người dùng nhập lại.
- Nhập ngày giao dịch (Date) : Date là kiểu chuỗi với độ dài tối đa 10 ký tự. Nếu độ dài nhập vào dài hơn thì hiển thị thông báo : “Date must be a string with max length = 10” và yêu cầu người dùng nhập lại.
- Nhập số lượng (Amount) : Amount phải là kiểu số và không chấp nhận các ký tự không phải số. Không giới hạn độ dài. Nếu người dùng nhập vào không phải ký tự số thì hiển thị thông báo : “Amount must be a number” và yêu cầu người dùng nhập lại.
- Nhập Loại Giao Dịch : Lúc này sẽ hiển thị 2 lựa chọn ra màn hình : “1: GoldExchange ; 2 : CurrencyExchange. Người dùng sẽ chọn 1 trong 2 giá trị đó . Nếu giá trị nhập vào không phải là 1 hoặc 2 thì hiển thị thông báo : “You must choose between 1 or 2” và yêu cầu người dùng nhập lại.
- Nếu loại giao dịch là 1 thì sẽ hiển thị 3 lựa chọn cho người dùng chọn :
“1 : 18k , 2 : 24k , 3 : 9999”
Nếu người dùng nhập vào không phải là 1, 2 hoặc 3 thì hiển thị thông báo : “You must choose correct gold type ” và yêu cầu người dùng nhập lại.
- Nếu loại giao dịch là 2 thì sẽ hiển thị 3 lựa chọn cho người dùng chọn :
“1 : USD , 2 : EUR , 3 : AUD”
Nếu người dùng nhập vào không phải là 1, 2 hoặc 3 thì hiển thị thông báo : “You must choose correct currency type ” và yêu cầu người dùng nhập lại.
- Nhập đơn giá (UnitPrice / ExchangeRate) : đơn giá lúc này sẽ được lưu thành Unit Price hoặc Exchange Rate tùy theo loại giao dịch tương ứng . Đơn giá phải là kiểu số và không chấp nhận các ký tự không phải số. Không giới hạn độ dài. Nếu người dùng nhập vào không phải ký tự số thì hiển thị thông báo : “UnitPrice / ExchangeRate must be a number” và yêu cầu người dùng nhập lại.
- Nếu loại giao dịch là 2 thì sẽ hiển thị 2 lựa chọn cho người dùng
“1 : Buy Currency , 2 : Sell Currency”
Nếu người dùng nhập vào không phải là 1, hoặc 2 thì hiển thị thông báo : “You must choose action with your currency ” và yêu cầu người dùng nhập lại.
- Sau đó in thông tin của giao dịch lên màn hình tương ứng với loại giao dịch và thông tin nhập vào.

Câu 11 : Viết chương trình cho phép người dùng nhập vào n giao dịch bằng cách sử dụng lại hàm đã viết ở câu 10. Sau đó in ra màn hình các Giao dịch với định dạng như sau : (Mỗi loại giao dịch sẽ được in khác nhau tùy theo loại giao dịch)

Exchange 1

Serial – Date – GoldType – UnitPrice – Amount – Cash Up = số tiền

Exchange 2

Serial – Date – GoldType – UnitPrice – Amount – Cash Up = số tiền

Exchange 3

Sell/Buy Exchange : Serial – Date – CurrencyType – ExchangeRate – Amount – Cash Up = số tiền

Câu 12 : Viết hàm để đếm số lượng giao dịch mỗi loại ở câu 11 và in ra màn hình với định dạng :
“Total of Gold Exchange :
Total of Currency Exchange : ...”