



TRƯỜNG ĐẠI HỌC VINH
VIỆN KỸ THUẬT VÀ CÔNG NGHỆ

CHƯƠNG 5

THIẾT KẾ MODULE

Ngệ An, 2021

ĐẶT VẤN ĐỀ

Tất cả ví dụ cho đến thời điểm này đều được thực thi trong command line hoặc từ một file python .py . Tuy nhiên, đối với các ứng dụng lớn, có nhiều chức năng thì phân chia nhỏ dự án thành các file khác nhau sẽ giúp dễ bảo trì và tái sử dụng các thành phần đã thiết kế.

Chương này sẽ giúp chúng ta thiết kế các tính năng theo mô hình các module và khi cần thì sẽ gọi file tương ứng và sử dụng.

Chương 5:

THIẾT KẾ MODULE



NỘI DUNG GIẢNG DẠY:

- 5.1. Các loại module/thư viện
- 5.2. Đường dẫn tìm để load module
- 5.3. Lấy danh sách thuộc tính và phương thức của một module
- 5.4. Cách khai báo và sử dụng module
- 5.5. Package module

Chương 5:

THIẾT KẾ MODULE



NỘI DUNG GIẢNG DẠY:

5.1. Các loại module/thư viện

5.2. Đường dẫn tìm để load module

5.3. Lấy danh sách thuộc tính và phương thức của một module

5.4. Cách khai báo và sử dụng module

5.5. Package module

MODULE/THƯ VIỆN LÀ GÌ

Module (mô-đun) đề cập đến một file chứa những câu lệnh Python và các định nghĩa.

Module thường được sử dụng khi muốn chia chương trình lớn thành những file nhỏ hơn để dễ quản lý và tổ chức.

Phổ biến nhất là những hàm Python hay phải sử dụng sẽ được định nghĩa trong một module và nhập vào Python thay vì sao chép định nghĩa trong những chương trình khác nhau. Nhờ thế, module cho phép tái sử dụng code.

CÁC LOẠI MODULE/THƯ VIỆN

Có 3 loại module thường thấy là:

- Viết bằng Python: có phần mở rộng là *.py
- Các thư viện liên kết động: có phần mở rộng là *.dll, *.pyd, *.so, *.sl, ...
- C-Module liên kết với trình biên dịch.

Chương 5:

THIẾT KẾ MODULE



NỘI DUNG GIẢNG DẠY:

5.1. Các loại module/thư viện

5.2. Đường dẫn tìm để load module

5.3. Lấy danh sách thuộc tính và phương thức của một module

5.4. Cách khai báo và sử dụng module

5.5. Package module

ĐƯỜNG DẪN TÌM ĐỂ LOAD MODULE

Để tải một module vào script sử dụng cú pháp:

```
import modulename
```

Khi gặp câu lệnh trên thì trình biên dịch sẽ tiến hành tìm kiếm file module tương ứng theo thứ tự thư mục sau:

1. Thư mục hiện hành mà script đang gọi;
2. Các thư mục trong PYTHONPATH (nếu có set);
3. Các thư mục cài đặt chuẩn trên Linux, Unix...

ĐƯỜNG DẪN TÌM ĐỂ LOAD MODULE

Có thể biết được đường dẫn mà một module đã được load bằng đoạn code dưới đây:

```
import random
```

```
random.__file__
```

(Ví dụ trả về '/usr/lib/python2.5/random.pyc')

THẢO LUẬN NHÓM

NỘI DUNG:

1. Tìm hiểu một số module quan trọng trong Python.
2. Tìm đường dẫn mà một số module.

BÀI TẬP

CHUẨN BỊ CHO BUỔI HỌC TIẾP THEO:

1. Đọc các tài liệu về nội dung mục 5.3; 5.4.
2. Tìm hiểu về đặc điểm của một số loại module trong lập trình Python.

Chương 5:

THIẾT KẾ MODULE



NỘI DUNG GIẢNG DẠY:

5.1. Các loại module/thư viện

5.2. Đường dẫn tìm để load module

5.3. Lấy danh sách thuộc tính và phương thức của một module

5.4. Cách khai báo và sử dụng module

5.5. Package module

LẤY DANH SÁCH THUỘC TÍNH VÀ PHƯƠNG THỨC CỦA MỘT MODULE



Để lấy được danh sách các thuộc tính và phương thức mà module hỗ trợ, sử dụng hàm **dir(modulename)**.

Ví dụ:

```
['__doc__', '__file__', '__name__', '__package__', 'acos', 'acosh', 'asin',  
'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees',  
'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum',  
'gamma', 'hypot', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p',  
'modf', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
```

LẤY DANH SÁCH THUỘC TÍNH VÀ PHƯƠNG THỨC CỦA MỘT MODULE



Có thể gọi hàm **dir()** không truyền tham số để lấy các thuộc tính và phương thức của scope hiện tại đang thực thi.

Chương 5:

THIẾT KẾ MODULE



NỘI DUNG GIẢNG DẠY:

- 5.1. Các loại module/thư viện
- 5.2. Đường dẫn tìm để load module
- 5.3. Lấy danh sách thuộc tính và phương thức của một module
- 5.4. Cách khai báo và sử dụng module**
- 5.5. Package module

CÁCH KHAI BÁO VÀ SỬ DỤNG MODULE



Giả sử chúng ta tạo một file python *mymath.py* có nội dung như sau:

```
def cong(a, b):  
    return a + b  
  
def tru(a, b):  
    return a - b  
  
def nhan(a, b):  
    return a * b
```


CÁCH KHAI BÁO VÀ SỬ DỤNG MODULE



Sau đó tạo một file có tên `myexample.py`, trong cùng thư mục với file `mymath.py` vừa tạo ở trên, có nội dung như sau:

```
import mymath  
num1 = 1  
num2 = 2  
print 'Tong hai so la: ',mymath.cong(num1, num2)
```

Vào command line, thực hiện gọi file `myexample`, sau khi thực hiện sẽ hiển thị lên màn hình là:

Tong hai so la: 3

Chương 5:

THIẾT KẾ MODULE



NỘI DUNG GIẢNG DẠY:

- 5.1. Các loại module/thư viện
- 5.2. Đường dẫn tìm để load module
- 5.3. Lấy danh sách thuộc tính và phương thức của một module
- 5.4. Cách khai báo và sử dụng module
- 5.5. Package module**

PACKAGE MODULE

Có thể gom nhiều module *.py* vào một thư mục và tên thư mục là tên của package và tạo một file `__init__.py` trong thư mục này.

Như vậy, cấu trúc thư của một package sẽ như sau:

```
| -- mypack
|   |-- __init__.py
|   |-- mymodule1.py
|   |-- mymodule2.py
|
```

PACKAGE MODULE

Có thể sử dụng mymodule1 theo cú pháp import sau:

```
import mypack.mymodule1
```

hoặc

```
import mypack.mymodule1 as mymodule1
```

hoặc

```
import mypack.mymodule1 as mod
```

PACKAGE MODULE

Khi sử dụng một module thuộc một package thì các lệnh trong file `__init__.py` sẽ được thực hiện trước.

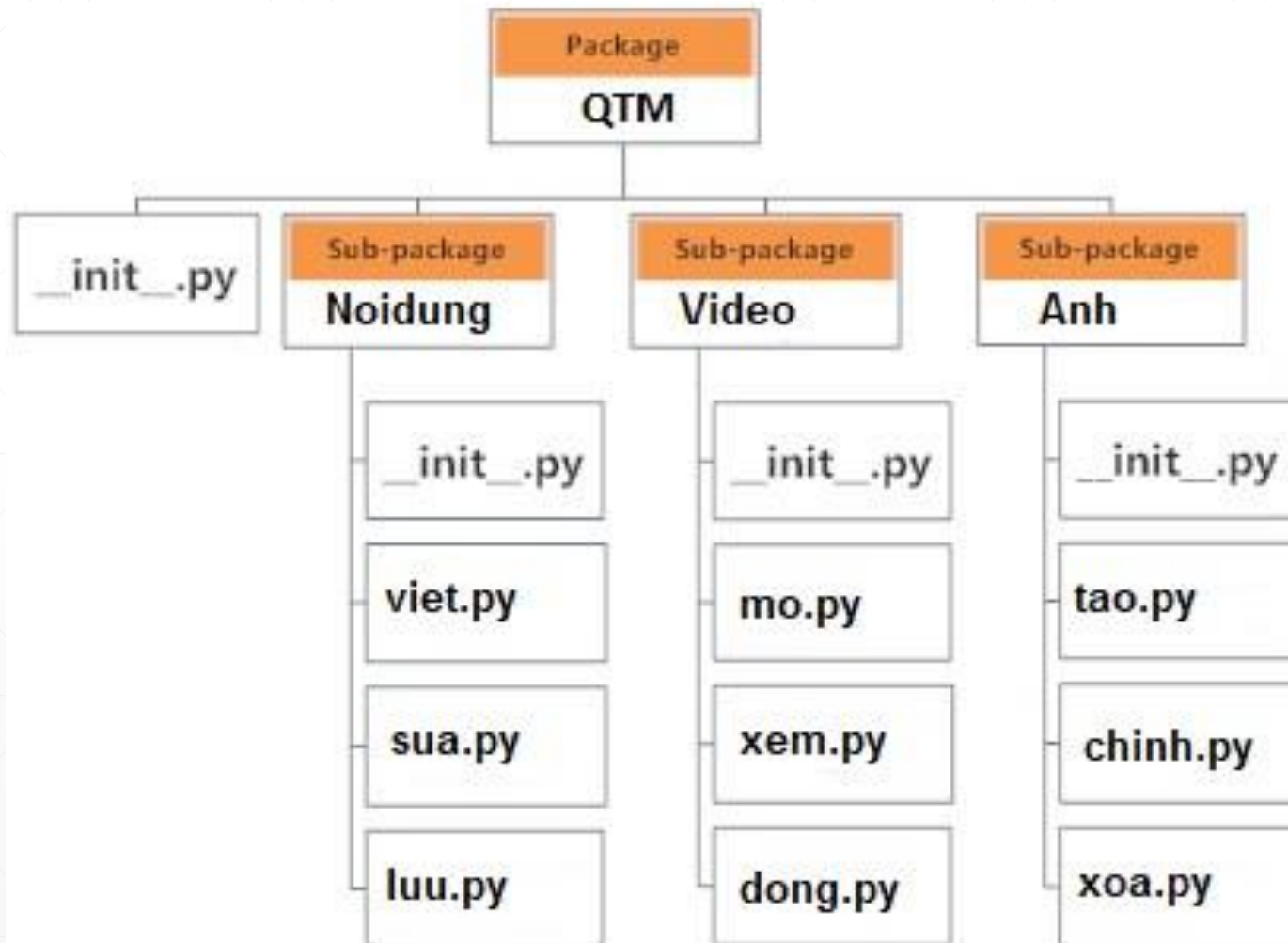
Thông thường thì file `__init__.py` sẽ rỗng.

Có thể tạo các subpackage bên trong một package theo đúng cấu trúc thư mục, có file `__init__.py`.

Ví dụ:

```
import mypack.mysubpack.mysubpack.module
```

PACKAGE MODULE



```
import QTM.Video.mo
```

```
import QTM.Video.mo.chon_video(video1)
```

```
from QTM.Video import mo
mo.chon_video(video1)
```

```
from QTM.Video.mo import chon_video
chon_video(video1)
```

BÀI TẬP

NỘI DUNG:

1. Tạo một số thập phân ngẫu nhiên, có giá trị nằm trong khoảng từ 10 đến 100 bằng cách sử dụng module random của Python.
2. Tạo module “hinhtron” với các phương thức tính chu vi, diện tích sử dụng module đã tạo trong chương trình.

BÀI TẬP

CHUẨN BỊ CHO BUỔI HỌC TIẾP THEO:

1. Đọc các tài liệu về nội dung mục 6.1; 6.2; 6.3 và 6.4.
2. Tìm hiểu và so sánh lập trình cấu trúc với hướng đối tượng.