



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
Posts and Telecommunications Institute of Technology

BÀI TẬP THỰC HÀNH

VŨ HOÀI THƯ

NỘI DUNG THỰC HÀNH

- ☐ Tạo và sử dụng lớp
- ☐ Làm việc với thể hiện của một lớp
- ☐ Kế thừa và sử dụng các lớp

THƯ VIỆN CHUẨN CỦA PYTHON

- ❑ Thư viện chuẩn Python là một tập hợp các module đi kèm khi cài đặt python.
- ❑ Một số thư viện chuẩn của Python như: math, random, datetime
- ❑ Cú pháp chung để sử dụng các thư viện của Python:

```
import name_library
```

❑ Module random trong Python chứa các hàm tạo số ngẫu nhiên, hàm lựa chọn các phần tử ngẫu nhiên,... Một số hàm được sử dụng trong module random:

- ✓ random()
- ✓ randint()
- ✓ choice()
- ✓ shuffle()

THƯ VIỆN RANDOM

- ❑ Hàm ***random()*** tạo ra một số thực ngẫu nhiên trong khoảng từ 0 đến 1, hàm này không yêu cầu bất kỳ đối số nào.

```
import random  
  
print(random.random())
```

0.39479752041404115

- ❑ Hàm ***randint()*** trả về một số nguyên ngẫu nhiên được chọn trong phạm vi chỉ định

```
print(random.randint(1, 10))
```

7

THƯ VIỆN RANDOM

- ❑ Hàm ***choice()*** trả về một phần tử được chọn ngẫu nhiên từ một tập hợp. Tập hợp có thể là một chuỗi ký tự, một danh sách,...

```
import random  
  
fruits = ["apple", "banana", "orange"]  
  
print(random.choice(fruits))
```

apple

```
strings = 'abcdefghij'  
  
random.choice(strings)
```

'a'

- ❑ Hàm ***choices()*** trả về một danh sách được chọn ngẫu nhiên từ một tập hợp.

```
import random  
  
fruits = ["apple", "banana", "orange"]  
  
print(random.choices(fruits, k=2))
```

['orange', 'apple']

BÀI TẬP

- ❑ Bài tập 1: Tạo một lớp có tên Number gồm các thuộc tính và phương thức sau:
 - ✓ Gồm 2 thuộc tính: number1 và number2
 - ✓ Phương thức input_info() dùng để hiển thị 2 số number1 và number2
 - ✓ Phương thức addition() để cộng 2 số number1 và number2
 - ✓ Phương thức subtract() để trừ 2 số number1 và number2
 - ✓ Phương thức multi() để nhân 2 số number1 và number2
 - ✓ Phương thức division() để chia 2 số number1 và number2
 - ✓ Khởi tạo đối tượng từ lớp vừa tạo và kiểm tra các phương thức đã viết.

BÀI TẬP

- ❑ Bài tập 2: Tạo một lớp có tên là ***Prime***. Tạo phương thức ***__init__()*** lưu trữ một thuộc tính x – là một số nguyên bất kỳ.
 - ✓ Tạo phương thức ***isprime()*** kiểm tra xem x có phải là số nguyên tố không.
 - ✓ Tạo phương thức ***next_prime()*** tìm và trả về số nguyên tố liền sau x .
 - ✓ Khởi tạo một đối tượng từ lớp ***Prime*** và kiểm tra các phương thức đã viết.

BÀI TẬP

- ❑ Bài tập 3: Tạo một lớp có tên Circle. Hàm khởi tạo `__init__()` lưu trữ hai thuộc tính: tâm hình tròn (lưu dưới dạng một tuple gồm 2 phần tử (x, y)) và bán kính của hình tròn.
 - ✓ Tạo phương thức `describe()` trả về mô tả của hình tròn có dạng “Hình tròn tâm $O(x,y)$ bán kính $= R$ ”
 - ✓ Tạo phương thức `get_area()` tính diện tích của hình tròn
 - ✓ Tạo phương thức `get_perimeter()` để tính chu vi của hình tròn.

BÀI TẬP

- ❑ Bài tập 4: Tạo một lớp có tên Matrix và các phương thức là các phép toán trên ma trận (cộng, trừ, nhân). Tạo một đối tượng từ lớp vừa tạo, thực hiện tính tổng và tích của hai ma trận.

BÀI TẬP

- ❑ Bài tập 5: Tạo một lớp có tên CheckNumber kế thừa từ lớp Prime gồm các thuộc tính và phương thức sau:
 - ✓ Phương thức `__init__()` lưu trữ một thuộc tính là một số nguyên bất kỳ
 - ✓ Phương thức `parity_check()` kiểm tra số nguyên là chẵn hay lẻ
 - ✓ Phương thức `is_prime()` kiểm tra số có phải số nguyên tố không
 - ✓ Phương thức `is_perfect()` kiểm tra số có phải là số hoàn hảo không
 - ✓ Phương thức `is_square()` kiểm tra xem số có phải là số chính phương không.