

Computer Science Advanced

LAB 2. SPECIAL METHODS VÀ KẾ THỪA

Bài 1. Hình Chữ Nhật

Python hỗ trợ chuyển đổi các đối tượng thành dạng *string* khi dùng lệnh `print()` trên đối tượng. Ví dụ, lệnh `print(Rectangle(height=2, width=1))` với lớp **Rectangle** ở bài trước cho ta kết quả:

```
<__main__.Rectangle object at 0x000001E206F85DF0>
```

Tuy nhiên, kết quả này có định dạng khá khó đọc và không cho ta thông tin về các thuộc tính của đối tượng.

Để cải tiến, hãy cài đặt lớp **Rectangle** tương tự như bài trước và có thêm phương thức `__str__(self)` định nghĩa cách chuyển đổi đối tượng thành dạng *string*.

Kết quả mong đợi khi sử dụng class trên như sau:

Code	Output
<pre>rect = Rectangle(2, 1) print(rect)</pre>	<pre>Rectangle object with height = 2 and width = 1</pre>

Bài 2. MathList

List trong Python là một cấu trúc dữ liệu mạnh khi xử lý trên danh sách, tuy nhiên lại không hỗ trợ nhiều về các tính toán trên dữ liệu số.

Ví dụ: Cho một *list* `values = [1, 2, 3, 4, 5]`. Nếu muốn tăng mỗi giá trị trong `values` lên 2 đơn vị, ta không thể dùng toán tử `+` mà phải sử dụng một vòng lặp.

Để có thể sử dụng toán tử cho việc tính toán trên danh sách, hãy cài đặt lớp **MathList** có:

- Phương thức khởi tạo nhận vào một *list* chứa dữ liệu số
- Phương thức `__str__()` trả về *list* đang lưu trữ dưới dạng *string*
- Phương thức `__add__()` và `__sub__()` nhận vào một số và thực hiện phép toán tương ứng với từng phần tử trong *list* đang lưu trữ

Kết quả mong đợi khi sử dụng class trên như sau:

Code	Output
<pre>m_list= MathList([1, 2, 3, 4, 5]) print(m_list) m_list += 2 print(m_list)</pre>	<pre>[1, 2, 3, 4, 5] [3, 4, 5, 6, 7]</pre>

Bài 3. Hình Vuông và Hình Lập Phương

Hình vuông và hình lập phương đều có chiều dài bằng nhau ở tất cả các cạnh. Một điểm khác biệt là hình lập phương có thể tích, còn hình vuông thì không. Do đó, khi thiết kế class, ta có thể thiết kế hình lập phương là class kế thừa từ hình vuông.

Hãy viết class **Square** đại diện cho hình vuông có:

- Phương thức khởi tạo nhận vào độ dài của một cạnh.
- Phương thức `cal_area()` trả về một số là diện tích của hình vuông.

Sau đó, viết class **Cube** đại diện cho hình lập phương, kế thừa từ class **Square** và có:

- Phương thức `cal_area()` trả về một số là diện tích của hình lập phương (*override* phương thức của **Square**).
- Phương thức `cal_volume()` trả về một số là thể tích của hình lập phương.

Kết quả mong đợi khi sử dụng các class trên như sau:

Code	Output
<pre>square = Square(2) print('Square area:', square.cal_area())</pre>	Square area: 4
<pre>cube = Cube(2) print('Cube area:', cube.cal_area()) print('Cube volume:', cube.cal_volume())</pre>	Cube area: 24 Cube volume: 8

Bài 4. Tài Khoản

Một trang web có hai loại tài khoản cho người dùng: *tài khoản mặc định* và *tài khoản đăng ký*.

- Tài khoản đăng ký có thể sử dụng mọi chức năng của tài khoản mặc định và các chức năng khác.
- Tài khoản đăng ký có thời hạn kết thúc.

Hãy viết class **User** đại diện cho *tài khoản mặc định*, có:

- Phương thức khởi tạo nhận vào *tên tài khoản* và *mật khẩu*.
- Phương thức `welcome()` in ra màn hình dòng chữ **Welcome, <tên tài khoản>**.
- Phương thức `check_password(password)` nhận vào một *string*, trả về **True** nếu *string* đã nhận khớp với mật khẩu của tài khoản, ngược lại trả về **False**.

Sau đó, viết class **SubscribedUser** đại diện cho *tài khoản đăng ký*, kế thừa từ **User**, có:

- Phương thức khởi tạo nhận vào *tên tài khoản*, *mật khẩu* và *ngày hết hạn* dưới dạng đối tượng `datetime`.
- Phương thức `is_expired()` trả về **True** nếu tài khoản đã hết hạn tại thời điểm hiện tại, ngược lại trả về **False**.

Kết quả mong đợi khi sử dụng các class trên như sau:

Code	Output
<pre>user = User('mindx', '12345') user.welcome() print(user.check_password('1234'))</pre>	Welcome, mindx False
<pre>s_user = SubscribedUser('s_mindx', '1234', datetime(2021, 1, 1)) s_user.welcome() print(s_user.check_password('1234')) print(s_user.is_expired())</pre>	Welcome, s_mindx True False

Gợi ý: Sử dụng hàm `datetime.now()` để lấy thời gian hiện tại và các toán tử so sánh `<`, `>`, `==` để kiểm tra tính trước sau của hai đối tượng `datetime`.