

Bài tập Thực hành Lập trình Hướng Đối Tượng trong Python

Hà Minh Tuấn

Khoa Toán - Tin học, Trường Đại học Khoa học Tự nhiên, ĐHQG - HCM.
hmtuan@hcmus.edu.vn

Mục tiêu tổng quát

- Làm chủ các khái niệm cơ bản của lập trình hướng đối tượng (OOP) trong Python.
- Phát triển kỹ năng thiết kế lớp, kế thừa, đa hình và đóng gói.
- Áp dụng OOP vào xử lý, phân tích và mô hình hóa dữ liệu.
- Xây dựng nền tảng để học các môn như Machine Learning và Data Engineering.

Giai đoạn 1: Làm quen với lớp và đối tượng (10 bài)

Mục tiêu: Nắm vững khái niệm lớp, đối tượng, thuộc tính và phương thức.

Bài 1.1: Tạo lớp `Student` có các thuộc tính: `name`, `id`, `major`. Tạo 3 đối tượng và in ra thông tin.

Bài 1.2: Viết phương thức `__init__()` cho lớp `Student` để khởi tạo dữ liệu khi tạo đối tượng.

Bài 1.3: Thêm phương thức `display()` để hiển thị thông tin sinh viên.

Bài 1.4: Viết lớp `Circle` có thuộc tính `radius`, phương thức `area()` và `perimeter()`.

Bài 1.5: Viết lớp `Rectangle` có phương thức tính diện tích và chu vi.

Bài 1.6: Tạo lớp `Book` với các thuộc tính `title`, `author`, `price`. Viết phương thức giảm giá 10%.

Bài 1.7: Xây dựng lớp `Temperature` có phương thức chuyển đổi từ Celsius sang Fahrenheit.

Bài 1.8: Tạo lớp `Calculator` thực hiện các phép tính cộng, trừ, nhân, chia.

Bài 1.9: Tạo lớp `Point` biểu diễn tọa độ (x, y) và tính khoảng cách đến gốc tọa độ.

Bài 1.10: Tạo lớp `ComplexNumber` mô phỏng số phức với các phép cộng, trừ.

Giai đoạn 2: Thuộc tính và phương thức nâng cao (10 bài)

Mục tiêu: Phân biệt thuộc tính lớp, đối tượng; sử dụng phương thức tĩnh và lớp.

Bài 2.11: Viết lớp Employee với thuộc tính lớp company = "TechCorp" và thuộc tính đối tượng name, salary.

Bài 2.12: Thêm phương thức @classmethod để thay đổi tên công ty.

Bài 2.13: Thêm phương thức @staticmethod để kiểm tra định dạng tên nhân viên.

Bài 2.14: Viết lớp BankAccount với thuộc tính riêng __balance, phương thức deposit() và withdraw() có kiểm tra hợp lệ.

Bài 2.15: Tạo lớp Counter đếm số đối tượng được tạo ra.

Bài 2.16: Viết lớp Product có thuộc tính ẩn __price và phương thức set_price() có kiểm tra giá hợp lệ.

Bài 2.17: Xây dựng lớp Vector2D có các phép cộng, trừ vector.

Bài 2.18: Viết lớp Polynomial biểu diễn đa thức, hỗ trợ in ra dạng $a_nx^n + \dots + a_0$.

Bài 2.19: Viết lớp Logger ghi lại các hoạt động của người dùng vào file.

Bài 2.20: Viết lớp Timer có phương thức start(), stop(), và elapsed() để đo thời gian chạy đoạn mã.

Giai đoạn 3: Kế thừa và Đa hình (10 bài)

Mục tiêu: Hiểu cách mở rộng lớp cha và áp dụng đa hình.

Bài 3.21: Tạo lớp cha Person và lớp con Student(Person) kế thừa các thuộc tính cơ bản.

Bài 3.22: Viết lớp Teacher(Person) có thêm thuộc tính subject.

Bài 3.23: Tạo lớp ResearchStudent(Student) có thêm topic.

Bài 3.24: Ghi đè phương thức (*method overriding*) trong lớp con để hiển thị thông tin chi tiết hơn.

Bài 3.25: Tạo lớp Animal, các lớp con Dog, Cat có phương thức speak() thể hiện đa hình.

Bài 3.26: Xây dựng lớp Shape trừu tượng và các lớp con Rectangle, Circle, Triangle đều có phương thức area().

Bài 3.27: Viết hàm nhận danh sách các Shape và tính tổng diện tích (đa hình).

Bài 3.28: Xây dựng hệ thống quản lý nhân sự gồm lớp cha Employee và các lớp con Manager, Engineer, Intern.

Bài 3.29: Viết ví dụ kế thừa đa tầng: Vehicle → Car → ElectricCar.

Bài 3.30: Tạo ví dụ kế thừa đa kế (multiple inheritance) giữa lớp Flyable và Swimmable.

Giai đoạn 4: Ứng dụng OOP trong Khoa học Dữ liệu (10 bài)

Mục tiêu: Thiết kế lớp cho các quy trình phân tích dữ liệu.

- Bài 4.31:** Viết lớp `DataLoader` đọc file CSV bằng `pandas`, có phương thức `load()` và `summary()`.
- Bài 4.32:** Tạo lớp `Preprocessor` có phương thức xử lý giá trị thiếu và chuẩn hóa dữ liệu.
- Bài 4.33:** Xây dựng lớp `DatasetSplitter` chia dữ liệu thành train/test.
- Bài 4.34:** Viết lớp `StatisticsAnalyzer` có các phương thức tính `mean()`, `variance()`, `correlation()`.
- Bài 4.35:** Viết lớp `FeatureSelector` chọn cột dữ liệu theo điều kiện.
- Bài 4.36:** Tạo lớp `Visualizer` vẽ biểu đồ histogram, scatter bằng `matplotlib`.
- Bài 4.37:** Viết lớp `LinearModel` với `fit()`, `predict()`, `evaluate()`.
- Bài 4.38:** Tạo lớp `Scaler` thực hiện chuẩn hóa MinMax hoặc StandardScaler.
- Bài 4.39:** Xây dựng lớp `Pipeline` kết hợp các bước tiền xử lý và huấn luyện mô hình.
- Bài 4.40:** Viết lớp `ReportGenerator` sinh báo cáo tóm tắt kết quả mô hình ra file text.

Giai đoạn 5: Dự án tổng hợp (10 bài)

Mục tiêu: Áp dụng toàn bộ kiến thức để thiết kế hệ thống phần mềm nhỏ bằng OOP.

Bài 5.41: Mini Project 1: Hệ thống quản lý sinh viên (thêm, xoá, sửa, tìm kiếm, lưu CSV).

Bài 5.42: Mini Project 2: Hệ thống quản lý thư viện (sách, người mượn, hạn trả).

Bài 5.43: Mini Project 3: Ứng dụng quản lý chi tiêu cá nhân (thêm giao dịch, thống kê).

Bài 5.44: Mini Project 4: Mô phỏng quá trình huấn luyện mô hình hồi quy tuyến tính bằng OOP.

Bài 5.45: Mini Project 5: Hệ thống pipeline xử lý dữ liệu: đọc → xử lý → huấn luyện → đánh giá.

Bài 5.46: Mini Project 6: Tích hợp lớp `Visualizer` để tự động sinh biểu đồ kết quả.

Bài 5.47: Mini Project 7: Thiết kế lớp `Experiment` quản lý các lần huấn luyện mô hình khác nhau.

Bài 5.48: Mini Project 8: Tạo hệ thống logging và error handling trong pipeline.

Bài 5.49: Mini Project 9: Mở rộng pipeline cho bài toán phân loại (classification).

Bài 5.50: Mini Project 10: Dự án cuối kỳ – sinh viên tự chọn đề tài (có thể là phân tích dữ liệu hoặc mô hình hoá) và trình bày cấu trúc OOP của dự án.