Advanced Calculator, Function Library và Code Organization

Thời gian: 3 tiếng (14:00-17:00)

Mục tiêu:

- Xây dựng máy tính nâng cao với nhiều chức năng
- Tạo thư viện functions có thể tái sử dụng
- Học cách tổ chức code hiệu quả

Kiến thức cần có:

- Function scope và global variables
- Default parameters
- Lambda functions
- Functions cơ bản từ các buổi trước

1. Xây dựng Advanced Calculator

1.1 Calculator cơ bản với Default Parameters

In [ ]:
```python
# Advanced Calculator với default parameters
import math

# Biến global để lưu lịch sử tính toán
lich_su_tinh_toan = []

def luu_lich_su(phep_tinh, ket_qua):
    """Lưu lịch sử tính toán"""
    global lich_su_tinh_toan
    lich_su_tinh_toan.append(f"{phep_tinh} = {ket_qua}")

def hien_thi_lich_su(so_luong=5):
    """Hiển thị lịch sử tính toán (mặc định 5 phép gần nhất)"""
    print(f"\n=== Lịch sử {so_luong} phép tính gần nhất ===")
    for i, phep_tinh in enumerate(lich_su_tinh_toan[-so_luong:], 1):
        print(f"{i}. {phep_tinh}")

def xoa_lich_su():
    """Xóa toàn bộ lịch sử"""
    global lich_su_tinh_toan
    lich_su_tinh_toan = []
    print("Đã xóa lịch sử tính toán!")
```

1.2 Các phép tính cơ bản với error handling

In [ ]:
```python
def cong(a, b, luu_ls=True):
    """Phép cộng với tùy chọn lưu lịch sử"""
    ket_qua = a + b
    if luu_ls:
        luu_lich_su(f"{a} + {b}", ket_qua)
    return ket_qua

def tru(a, b, luu_ls=True):
    """Phép trừ với tùy chọn lưu lịch sử"""
    ket_qua = a - b
    if luu_ls:
        luu_lich_su(f"{a} - {b}", ket_qua)
    return ket_qua

def nhan(a, b, luu_ls=True):
    """Phép nhân với tùy chọn lưu lịch sử"""
    ket_qua = a * b
    if luu_ls:
        luu_lich_su(f"{a} × {b}", ket_qua)
    return ket_qua

def chia(a, b, luu_ls=True):
    """Phép chia với xử lý lỗi chia cho 0"""
    if b == 0:
        print("Lỗi: Không thể chia cho 0!")
        return None
    ket_qua = a / b
    if luu_ls:
        luu_lich_su(f"{a} ÷ {b}", ket_qua)
    return ket_qua

# Test các phép tính cơ bản
print(f"10 + 5 = {cong(10, 5)}")
print(f"10 - 3 = {tru(10, 3)}")
print(f"4 × 6 = {nhan(4, 6)}")
print(f"15 ÷ 3 = {chia(15, 3)}")
print(f"10 ÷ 0 = {chia(10, 0)}")

hien_thi_lich_su()
```

🔥 TODO 1: Thêm phép tính nâng cao

Yêu cầu:

1. Viết function `luy_thua(a, b=2, luu_ls=True)` - mặc định tính bình phương
2. Viết function `can_bac_hai(a, luu_ls=True)` với kiểm tra số âm
3. Viết function `giai_thua(n, luu_ls=True)` với kiểm tra số nguyên dương
4. Test tất cả functions

In [ ]:
```python
# TODO 1: Viết code ở đây

def luy_thua(a, b=2, luu_ls=True):
    """Tính lũy thừa, mặc định bình phương"""
    # Viết code
    pass

def can_bac_hai(a, luu_ls=True):
    """Tính căn bậc hai với kiểm tra số âm"""
    # Viết code (kiểm tra a >= 0)
    pass

def giai_thua(n, luu_ls=True):
    """Tính giai thừa với kiểm tra số nguyên dương"""
    # Viết code (kiểm tra n >= 0 và n là số nguyên)
    pass

# Test
print(f"5^2 = {luy_thua(5)}")
print(f"2^10 = {luy_thua(2, 10)}")
print(f"√16 = {can_bac_hai(16)}")
print(f"√(-4) = {can_bac_hai(-4)}")
print(f"5! = {giai_thua(5)}")
print(f"(-3)! = {giai_thua(-3)}")

hien_thi_lich_su()
```

2. Function Library - Thư viện Functions

2.1 Math Library - Thư viện toán học

In [ ]:
```python
# Math Library
class MathLibrary:
    """Thư viện các functions toán học"""

    @staticmethod
    def la_so_nguyen_to(n):
        """Kiểm tra số nguyên tố"""
        if n < 2:
            return False
        for i in range(2, int(n**0.5) + 1):
            if n % i == 0:
                return False
        return True

    @staticmethod
    def tim_uoc_so(n):
        """Tìm tất cả ước số của n"""
        uoc_so = []
        for i in range(1, n + 1):
            if n % i == 0:
                uoc_so.append(i)
        return uoc_so
```

```
    @staticmethod
    def ucln(a, b):
        """Tìm ước chung lớn nhất"""
        while b:
            a, b = b, a % b
        return a

    @staticmethod
    def bcnn(a, b):
        """Tìm bội chung nhỏ nhất"""
        return abs(a * b) // MathLibrary.ucln(a, b)

# Test Math Library
print(f"17 là số nguyên tố: {MathLibrary.la_so_nguyen_to(17)}")
print(f"Ước số của 12: {MathLibrary.tim_uoc_so(12)}")
print(f"ƯCLN(12, 18) = {MathLibrary.ucln(12, 18)}")
print(f"BCNN(12, 18) = {MathLibrary.bcnn(12, 18)}")
```

2.2 Statistics Library - Thư viện thống kê

In [ ]:

```
# Statistics Library sử dụng lambda functions
class StatsLibrary:
    """Thư viện các functions thống kê"""

    @staticmethod
    def trung_binh(danh_sach):
        """Tính trung bình"""
        if not danh_sach:
            return 0
        return sum(danh_sach) / len(danh_sach)

    @staticmethod
    def trung_vi(danh_sach):
        """Tính trung vị"""
        if not danh_sach:
            return 0
        sorted_list = sorted(danh_sach)
        n = len(sorted_list)
        if n % 2 == 0:
            return (sorted_list[n//2 - 1] + sorted_list[n//2]) / 2
        return sorted_list[n//2]

    @staticmethod
    def phuong_sai(danh_sach):
        """Tính phương sai"""
        if len(danh_sach) < 2:
            return 0
        tb = StatsLibrary.trung_binh(danh_sach)
        # Sử dụng lambda function
        binh_phuong_sai_lech = list(map(lambda x: (x - tb)**2, danh_sach))
        return sum(binh_phuong_sai_lech) / len(danh_sach)

    @staticmethod
    def do_lech_chuan(danh_sach):
```

```
        """Tính độ lệch chuẩn"""
        return StatsLibrary.phuong_sai(danh_sach) ** 0.5

# Test Statistics Library
diem_thi = [7, 8, 6, 9, 7, 8, 5, 9, 8, 7]
print(f"Điểm thi: {diem_thi}")
print(f"Trung bình: {StatsLibrary.trung_binh(diem_thi):.2f}")
print(f"Trung vị: {StatsLibrary.trung_vi(diem_thi)}")
print(f"Phương sai: {StatsLibrary.phuong_sai(diem_thi):.2f}")
print(f"Độ lệch chuẩn: {StatsLibrary.do_lech_chuan(diem_thi):.2f}")
```

🔥 TODO 2: Xây dựng Geometry Library

Yêu cầu:

1. Tạo class `GeometryLibrary`
2. Viết functions tính diện tích: `dien_tich_hinh_chu_nhat`, `dien_tich_hinh_tron`, `dien_tich_tam_giac`
3. Viết functions tính chu vi tương ứng
4. Sử dụng default parameters cho π = 3.14159
5. Test tất cả functions

In [ ]:

```
# TODO 2: Viết code ở đây

class GeometryLibrary:
    """Thư viện các functions hình học"""

    @staticmethod
    def dien_tich_hinh_chu_nhat(dai, rong):
        """Tính diện tích hình chữ nhật"""
        # Viết code
        pass

    @staticmethod
    def chu_vi_hinh_chu_nhat(dai, rong):
        """Tính chu vi hình chữ nhật"""
        # Viết code
        pass

    @staticmethod
    def dien_tich_hinh_tron(ban_kinh, pi=3.14159):
        """Tính diện tích hình tròn"""
        # Viết code
        pass

    @staticmethod
    def chu_vi_hinh_tron(ban_kinh, pi=3.14159):
        """Tính chu vi hình tròn"""
        # Viết code
        pass

    @staticmethod
    def dien_tich_tam_giac(day, cao):
```

```
        """Tính diện tích tam giác"""
        # Viết code
        pass

# Test
print(f"Diện tích HCN 5x3: {GeometryLibrary.dien_tich_hinh_chu_nhat(5, 3)}")
print(f"Chu vi HCN 5x3: {GeometryLibrary.chu_vi_hinh_chu_nhat(5, 3)}")
print(f"Diện tích hình tròn r=5: {GeometryLibrary.dien_tich_hinh_tron(5)}")
print(f"Chu vi hình tròn r=5: {GeometryLibrary.chu_vi_hinh_tron(5)}")
print(f"Diện tích tam giác đáy=6, cao=4: {GeometryLibrary.dien_tich_tam_giac(6, 4)}
```

3. Code Organization - Tổ chức code

3.1 Menu-driven Calculator

In [ ]:

```
# Menu-driven Calculator với code organization tốt

def hien_thi_menu():
    """Hiển thị menu chính"""
    print("\n" + "="*50)
    print("           ADVANCED CALCULATOR")
    print("="*50)
    print("1. Phép tính cơ bản (+, -, ×, ÷)")
    print("2. Phép tính nâng cao (^, √, !)")
    print("3. Toán học (Số nguyên tố, ƯCLN, BCNN)")
    print("4. Thống kê (TB, trung vị, độ lệch chuẩn)")
    print("5. Hình học (Diện tích, chu vi)")
    print("6. Lịch sử tính toán")
    print("0. Thoát")
    print("="*50)

def menu_phep_tinh_co_ban():
    """Menu phép tính cơ bản"""
    print("\n=== PHÉP TÍNH CƠ BẢN ===")
    try:
        a = float(input("Nhập số thứ nhất: "))
        phep_tinh = input("Nhập phép tính (+, -, *, /): ")
        b = float(input("Nhập số thứ hai: "))

        if phep_tinh == "+":
            ket_qua = cong(a, b)
        elif phep_tinh == "-":
            ket_qua = tru(a, b)
        elif phep_tinh == "*":
            ket_qua = nhan(a, b)
        elif phep_tinh == "/":
            ket_qua = chia(a, b)
        else:
            print("Phép tính không hợp lệ!")
            return

        if ket_qua is not None:
            print(f"Kết quả: {ket_qua}")
    except ValueError:
```

```
        print("Lỗi: Vui lòng nhập số hợp lệ!")

def menu_lich_su():
    """Menu quản lý lịch sử"""
    print("\n=== LỊCH SỬ TÍNH TOÁN ===")
    print("1. Xem lịch sử")
    print("2. Xóa lịch sử")

    lua_chon = input("Chọn chức năng (1-2): ")

    if lua_chon == "1":
        try:
            so_luong = int(input("Số phép tính muốn xem (mặc định 5): ") or 5)
            hien_thi_lich_su(so_luong)
        except ValueError:
            hien_thi_lich_su()
    elif lua_chon == "2":
        xac_nhan = input("Bạn có chắc muốn xóa lịch sử? (y/n): ")
        if xac_nhan.lower() == 'y':
            xoa_lich_su()

# Test menu
hien_thi_menu()
menu_phep_tinh_co_ban()
```

3.2 Dictionary-based Function Mapping

In [ ]:

```
# Sử dụng dictionary để map functions - cách tổ chức code hiệu quả

# Dictionary chứa các phép tính với lambda functions
phep_tinh_co_ban = {
    '+': lambda x, y: x + y,
    '-': lambda x, y: x - y,
    '*': lambda x, y: x * y,
    '/': lambda x, y: x / y if y != 0 else None,
    '^': lambda x, y: x ** y,
    '%': lambda x, y: x % y if y != 0 else None
}

# Dictionary chứa các function một tham số
phep_tinh_mot_tham_so = {
    'sqrt': lambda x: x ** 0.5 if x >= 0 else None,
    'square': lambda x: x ** 2,
    'cube': lambda x: x ** 3,
    'abs': lambda x: abs(x),
    'neg': lambda x: -x
}

def tinh_toan_nhanh(bieu_thuc):
    """Tính toán nhanh từ biểu thức string"""
    try:
        # Thay thế các ký hiệu
        bieu_thuc = bieu_thuc.replace('^', '**')
        bieu_thuc = bieu_thuc.replace('√', 'sqrt')
```

```
        # Đánh giá biểu thức (cẩn thận với eval!)
        ket_qua = eval(bieu_thuc)
        luu_lich_su(bieu_thuc, ket_qua)
        return ket_qua
    except:
        return "Lỗi: Biểu thức không hợp lệ"

# Test function mapping
print("=== TEST FUNCTION MAPPING ===")
print(f"5 + 3 = {phep_tinh_co_ban['+'](5, 3)}")
print(f"10 / 2 = {phep_tinh_co_ban['/'](10, 2)}")
print(f"2 ^ 8 = {phep_tinh_co_ban['^'](2, 8)}")
print(f"sqrt(16) = {phep_tinh_mot_tham_so['sqrt'](16)}")
print(f"square(7) = {phep_tinh_mot_tham_so['square'](7)}")

print("\n=== TEST TÍNH TOÁN NHANH ===")
print(f"2 + 3 * 4 = {tinh_toan_nhanh('2 + 3 * 4')}")
print(f"(5 + 3) * 2 = {tinh_toan_nhanh('(5 + 3) * 2')}")
```

🔥 TODO 3: Xây dựng Complete Calculator System

Yêu cầu:

1. Tạo function `calculator_main()` chạy calculator hoàn chỉnh
2. Sử dụng dictionary để map menu choices với functions
3. Thêm error handling cho tất cả inputs
4. Tích hợp tất cả libraries đã tạo
5. Cho phép người dùng chọn chức năng và thực hiện liên tục

In [ ]:

```
# TODO 3: Viết code ở đây

def menu_toan_hoc():
    """Menu các chức năng toán học"""
    print("\n=== CHỨC NĂNG TOÁN HỌC ===")
    print("1. Kiểm tra số nguyên tố")
    print("2. Tìm ước số")
    print("3. Tính ƯCLN")
    print("4. Tính BCNN")

    lua_chon = input("Chọn chức năng (1-4): ")

    try:
        if lua_chon == "1":
            n = int(input("Nhập số cần kiểm tra: "))
            ket_qua = MathLibrary.la_so_nguyen_to(n)
            print(f"{n} {'là' if ket_qua else 'không phải'} số nguyên tố")
        elif lua_chon == "2":
            n = int(input("Nhập số cần tìm ước: "))
            ket_qua = MathLibrary.tim_uoc_so(n)
            print(f"Ước số của {n}: {ket_qua}")
        elif lua_chon == "3":
            a = int(input("Nhập số thứ nhất: "))
            b = int(input("Nhập số thứ hai: "))
```

```
            ket_qua = MathLibrary.ucln(a, b)
            print(f"ƯCLN({a}, {b}) = {ket_qua}")
        elif lua_chon == "4":
            a = int(input("Nhập số thứ nhất: "))
            b = int(input("Nhập số thứ hai: "))
            ket_qua = MathLibrary.bcnn(a, b)
            print(f"BCNN({a}, {b}) = {ket_qua}")
    except ValueError:
        print("Lỗi: Vui lòng nhập số nguyên hợp lệ!")

def menu_thong_ke():
    """Menu các chức năng thống kê"""
    print("\n=== CHỨC NĂNG THỐNG KÊ ===")
    print("1. Tính trung bình")
    print("2. Tính trung vị")
    print("3. Tính phương sai")
    print("4. Tính độ lệch chuẩn")
    print("5. Phân tích tổng hợp")

    lua_chon = input("Chọn chức năng (1-5): ")

    try:
        # Nhập dữ liệu
        du_lieu = input("Nhập danh sách số (cách nhau bởi dấu phẩy): ")
        danh_sach = [float(x.strip()) for x in du_lieu.split(',')]

        if lua_chon == "1":
            ket_qua = StatsLibrary.trung_binh(danh_sach)
            print(f"Trung bình: {ket_qua:.2f}")
        elif lua_chon == "2":
            ket_qua = StatsLibrary.trung_vi(danh_sach)
            print(f"Trung vị: {ket_qua}")
        elif lua_chon == "3":
            ket_qua = StatsLibrary.phuong_sai(danh_sach)
            print(f"Phương sai: {ket_qua:.2f}")
        elif lua_chon == "4":
            ket_qua = StatsLibrary.do_lech_chuan(danh_sach)
            print(f"Độ lệch chuẩn: {ket_qua:.2f}")
        elif lua_chon == "5":
            print(f"\n=== PHÂN TÍCH TỔNG HỢP ===")
            print(f"Dữ liệu: {danh_sach}")
            print(f"Trung bình: {StatsLibrary.trung_binh(danh_sach):.2f}")
            print(f"Trung vị: {StatsLibrary.trung_vi(danh_sach)}")
            print(f"Phương sai: {StatsLibrary.phuong_sai(danh_sach):.2f}")
            print(f"Độ lệch chuẩn: {StatsLibrary.do_lech_chuan(danh_sach):.2f}")
            print(f"Min: {min(danh_sach)}")
            print(f"Max: {max(danh_sach)}")
    except ValueError:
        print("Lỗi: Vui lòng nhập dữ liệu số hợp lệ!")

def menu_hinh_hoc():
    """Menu các chức năng hình học"""
    print("\n=== CHỨC NĂNG HÌNH HỌC ===")
    print("1. Hình chữ nhật")
    print("2. Hình tròn")
    print("3. Tam giác")
```

```
    lua_chon = input("Chọn hình (1-3): ")

    try:
        if lua_chon == "1":
            dai = float(input("Nhập chiều dài: "))
            rong = float(input("Nhập chiều rộng: "))
            dt = GeometryLibrary.dien_tich_hinh_chu_nhat(dai, rong)
            cv = GeometryLibrary.chu_vi_hinh_chu_nhat(dai, rong)
            print(f"Diện tích: {dt}")
            print(f"Chu vi: {cv}")
        elif lua_chon == "2":
            ban_kinh = float(input("Nhập bán kính: "))
            dt = GeometryLibrary.dien_tich_hinh_tron(ban_kinh)
            cv = GeometryLibrary.chu_vi_hinh_tron(ban_kinh)
            print(f"Diện tích: {dt:.2f}")
            print(f"Chu vi: {cv:.2f}")
        elif lua_chon == "3":
            day = float(input("Nhập độ dài đáy: "))
            cao = float(input("Nhập chiều cao: "))
            dt = GeometryLibrary.dien_tich_tam_giac(day, cao)
            print(f"Diện tích: {dt}")
    except ValueError:
        print("Lỗi: Vui lòng nhập số hợp lệ!")

def calculator_main():
    """Function chính chạy calculator hoàn chỉnh"""

    # Dictionary map menu choices với functions
    menu_functions = {
        '1': menu_phep_tinh_co_ban,
        '2': lambda: print("Chức năng phép tính nâng cao"),  # TODO: implement
        '3': menu_toan_hoc,
        '4': menu_thong_ke,
        '5': menu_hinh_hoc,
        '6': menu_lich_su
    }

    print("🎉 Chào mừng đến với Advanced Calculator!")

    while True:
        try:
            hien_thi_menu()
            lua_chon = input("\nChọn chức năng (0-6): ").strip()

            if lua_chon == '0':
                print("\n👋 Cảm ơn bạn đã sử dụng Advanced Calculator!")
                break
            elif lua_chon in menu_functions:
                menu_functions[lua_chon]()
            else:
                print("❌ Lựa chọn không hợp lệ! Vui lòng chọn từ 0-6.")

            input("\nNhấn Enter để tiếp tục...")

        except KeyboardInterrupt:
```

```
            print("\n\n👋 Tạm biệt!")
            break
        except Exception as e:
            print(f"❌ Lỗi không mong muốn: {e}")

# Uncomment để chạy calculator
# calculator_main()
```

4. Bài tập và Thực hành

🔥 TODO 4: Code Organization Challenge

Yêu cầu:

1. Tạo file `my_calculator.py` chứa tất cả code đã viết
2. Chia code thành các modules riêng biệt:
   - `math_lib.py`: MathLibrary
   - `stats_lib.py`: StatsLibrary
   - `geometry_lib.py`: GeometryLibrary
   - `calculator_main.py`: Main program
3. Sử dụng `import` để kết nối các modules
4. Test toàn bộ system

In [ ]:
```
# TODO 4: Organization practice

# Example: math_lib.py
# class MathLibrary:
#     ... (copy từ trên)

# Example: calculator_main.py
# from math_lib import MathLibrary
# from stats_lib import StatsLibrary
# from geometry_lib import GeometryLibrary
#
# def main():
#     calculator_main()
#
# if __name__ == "__main__":
#     main()

print("📝 Hãy tạo các file riêng biệt và tổ chức code!")
```

5. Best Practices Summary

🎯 Key Takeaways từ buổi học:

1. Function Organization:

- Sử dụng default parameters hợp lý

- Error handling cho mọi function
- Docstrings rõ ràng và đầy đủ
- Single responsibility principle

2. Code Structure:

- Menu-driven interface cho user experience tốt
- Dictionary mapping cho code clean và maintainable
- Global variables sử dụng có kiểm soát
- Lambda functions cho operations đơn giản

3. Library Design:

- Static methods cho utility functions
- Logical grouping của related functions
- Consistent naming conventions
- Reusable và modular design

4. Error Handling:

- Try-catch cho user inputs
- Meaningful error messages
- Graceful degradation
- Input validation

🚀 Next Steps:

- File I/O operations (mai học)
- More advanced data structures
- Object-oriented programming
- GUI development với tkinter

7. Homework - Dự án cuối tuần

🏠 Dự án 1: Personal Finance Calculator

Xây dựng máy tính tài chính cá nhân với các chức năng:

- Tính lãi suất kép
- Tính khoản vay và lãi hàng tháng
- Lập kế hoạch tiết kiệm
- Phân tích chi tiêu
- Lưu/đọc dữ liệu từ file

🏠 Dự án 2: Student Grade Management System

Tạo hệ thống quản lý điểm học sinh với:

- Thêm/sửa/xóa thông tin học sinh
- Tính điểm trung bình có trọng số
- Xếp loại học lực (Giỏi, Khá, Trung bình, Yếu)
- Thống kê theo lớp/môn học
- Xuất báo cáo chi tiết
- Import/Export từ CSV

🏠 Dự án 3: Multi-Purpose Converter

Xây dựng bộ chuyển đổi đa năng:

- Đơn vị đo lường (m, km, inch, feet)
- Trọng lượng (kg, pound, ounce)
- Nhiệt độ (°C, °F, Kelvin)
- Tiền tệ (với tỷ giá cố định)
- Thời gian (giây, phút, giờ, ngày)
- Lưu lịch sử chuyển đổi

📋 Yêu cầu chung cho tất cả dự án:

Technical Requirements:

- Sử dụng function scope và global variables hợp lý
- Áp dụng default parameters
- Sử dụng lambda functions với map(), filter(), sorted()
- Tổ chức code theo best practices
- Error handling đầy đủ
- Menu-driven interface

Documentation Requirements:

- Docstrings cho mỗi function
- Comments giải thích logic phức tạp
- README file mô tả cách sử dụng
- Test cases cho các functions chính

Bonus Features (optional):

- Quick command system
- Configuration settings
- Data export/import
- Colorful console output
- Progress tracking

🎯 Tiêu chí đánh giá:

1. **Functionality (40%):** Đầy đủ chức năng theo yêu cầu
2. **Code Quality (30%):** Clean code, organization, best practices
3. **Error Handling (15%):** Xử lý lỗi đầy đủ và user-friendly
4. **Documentation (10%):** Docstrings, comments, README
5. **Innovation (5%):** Tính sáng tạo và features bonus

🗓️ Timeline:

- **Ngày mai (Thứ 4):** Bắt đầu file handling - có thể áp dụng ngay vào dự án
- **Thứ 5:** Hoàn thành core functions
- **Thứ 6:** Testing, debugging, documentation
- **Tuần sau:** Demo và review dự án

💡 Tips for Success:

1. **Start Simple:** Bắt đầu với basic functions trước
2. **Test Early:** Test từng function ngay khi viết xong
3. **Refactor Often:** Cải thiện code organization liên tục
4. **Use Libraries:** Tận dụng các libraries đã học
5. **Ask Questions:** Đừng ngại hỏi khi gặp khó khăn

Chúc các bạn code vui vẻ và thành công! 🚀