CẨM NANG CHIẾN THUẬT OOP

(THE ULTIMATE GUIDE FOR FINAL EXAM)

Ôn tập Thực hành OOP

Ngày 30 tháng 12 năm 2025

Mục lục

1 BỘ CÔNG CỤ "VŨ KHÍ"(MODULES) 2

1.1 AllInOne.h (File Nền Tảng - Bắt buộc có) 2

1.2 Adaptations.h (File Biến Thể - Dùng khi gặp đề dị) 2

2 QUY TRÌNH "BẮT BỆNH"TỪ HÀM MAIN() 3

3 BẢN ĐỒ 6 KỊCH BẢN (SCENARIOS MAP) 3

3.1 Dạng 1: Tích Lũy (Cộng dồn) - Dễ nhất 3

3.2 Dạng 2: Trung Bình & Đếm - Bài Nhân sự 3

3.3 Dạng 3: Toán Học Nghịch Đảo - Bài Mạch điện 3

3.4 Dạng 4: Lan Truyền Trạng Thái - Bài Smart Home 4

3.5 Dạng 5: Công Thức Phức Tạp - Bài Chi tiết máy 4

3.6 Dạng 6: Decorator & Strategy - Bài Cafe/Khuyến mãi 4

4 NHỮNG CÁI BẪY "CHẾT NGƯỜI"(PITFALLS) 4

5 LỜI KHUYÊN CUỐI CÙNG 5

1 BỘ CÔNG CỤ "VŨ KHÍ"(MODULES)

Bạn có 2 file header cốt lõi giải quyết 95% các dạng bài thi.

1.1 AllInOne.h (File Nền Tảng - Bắt buộc có)

- **BaseObject**: Lớp cha tối cao của mọi đối tượng.
  - *Chức năng:* Tích hợp sẵn `_name`, `GetValue()` ảo, `GetCategory()` (mới), và `operator«`.
- **Singleton<T>**: Template quản lý đối tượng duy nhất.
  - *Dùng khi:* Thấy `Manager::GetInstance()`, `System::GetInstance()`.
- **CompositeNode**: Quản lý danh sách con (`vector`).
  - *Chức năng:* Tự động `Add`, tự động `Destructor` (xóa bộ nhớ), có `Iterator` (hỗ trợ for-loop), có `SortChildren`.
  - *Dùng khi:* Quan hệ chứa đựng (Thư mục chứa File, Mạch chứa Điện trở, Giỏ hàng).
- **Utils**: Bộ công cụ xử lý chuỗi.
  - *Chức năng:* `Split` (cắt chuỗi), `ToNum` (chuyển số an toàn).
- **Date**: Xử lý ngày tháng.
  - *Chức năng:* Nhập/Xuất, So sánh ngày (`<`, `<=`, `==`).
- **AppError**: Class ngoại lệ (Exception).
- **Macros**: Sinh nhanh Constructor và Operator (`+`, `++`).

1.2 Adaptations.h (File Biến Thể - Dùng khi gặp đề dị)

- **Logic Tính Toán (GetValue variants):**
  - Tìm Max/Min.
  - Tính Nghịch đảo (Mạch song song).
  - Đếm số lượng (Tính trung bình).
- **Logic Kiểm Tra (Singleton variants):**
  - Check trùng mã (Unique ID) dùng `vector` + `find` (hoặc `set`).
- **Logic Parsing (AdvancedParser):**
  - Xử lý format lạ: `Name - Value` hoặc `[Name] (Value)`.

2 QUY TRÌNH "BẮT BỆNH"TỪ HÀM MAIN()

Khi nhận đề, **đừng viết class ngay**. Hãy nhìn vào `main()` để ánh xạ:

Dấu hiệu trong main()	**Kết luận & Hành động**
`Manager::GetInstance()->Set(...)`	**Singleton**. Copy template. Viết hàm xử lý logic bên trong (Check Max hoặc Check Trùng).
`Box->Add(Item)`	**Composite**. Class Box kế thừa `CompositeNode`.
`Factory::Create("A, 10")`	**Factory**. Copy `Utils::Split`. **Lưu ý:** Xem kỹ dấu phân cách là `,` hay `|` hay `-`.
`try { ... } catch (...)`	**Exception**. Các hàm `Add` hoặc `Create` phải có lệnh `throw AppError(...)`.
`cout << *obj`	**Output**. `BaseObject` đã lo, chỉ cần override hàm `Xuat`.
`5 + *obj` hoặc `obj++`	**Operator**. Dùng Macro `ENABLE_OP...` hoặc viết hàm friend.
`for (auto x : *box)`	**Iterator**. `CompositeNode` đã hỗ trợ sẵn.
`list->Sort()` hoặc `Find()`	**Sort/Search**. Dùng `SortChildren` (trong AllInOne mới) hoặc viết hàm đệ quy.

3 BẢN ĐỒ 6 KỊCH BẢN (SCENARIOS MAP)

3.1 Dạng 1: Tích Lũy (Cộng dồn) - Dễ nhất

- **Ví dụ:** Tổng dung lượng File, Tổng tải trọng, Tổng tiền đơn giản.
- **Logic:** `GetValue` = Tổng `GetValue` của con.
- **Check:** Singleton kiểm tra giới hạn (Max Load).

3.2 Dạng 2: Trung Bình & Đếm - Bài Nhân sự

- **Ví dụ:** Lương trung bình phòng ban.
- **Cần thêm:** Hàm `GetCount()` (ảo) ở `BaseObject` (hoặc lớp trung gian).
- **Logic:** Composite trả về tổng count con. Leaf trả về 1.
- **Công thức:** `Average = GetValue() / GetCount()`.
- **Lưu ý:** Cần `dynamic_cast` khi duyệt con.

3.3 Dạng 3: Toán Học Nghịch Đảo - Bài Mạch điện

- **Ví dụ:** Điện trở song song.
- **Logic:** $1/R_{tong} = 1/R_1 + 1/R_2...$
- **Lưu ý:** Check chia cho 0. Singleton kiểm tra an toàn ($I = U/R$).

3.4 Dạng 4: Lan Truyền Trạng Thái - Bài Smart Home

- **Ví dụ:** Tắt cầu dao tổng → Tất cả thiết bị tắt.
- **Cần thêm:** Biến `_isOn` và hàm `SetState(bool)` trong lớp cha.
- **Logic:** Khi cha `SetState`, vòng lặp gọi con `SetState` theo.
- **GetValue:** `return _isOn ? _value : 0;` (Tắt thì giá trị bằng 0).

3.5 Dạng 5: Công Thức Phức Tạp - Bài Chi tiết máy

- **Ví dụ:** Giá = Tổng giá con + 20% phí lắp ráp.
- **Logic:** Không cộng dồn đơn thuần.

```
    Sum = Sum(child->GetValue());
    return Sum * 1.2;

```

- **Lưu ý:** Singleton (Kho hàng) phải trừ số lượng khi `Add`.

3.6 Dạng 6: Decorator & Strategy - Bài Cafe/Khuyến mãi

- **Ví dụ:** Cafe thêm Topping, Tính tiền theo hạng thẻ VIP.
- **Logic:**
  - *Topping (Decorator):* Lớp con chứa 1 `BaseObject*` bên trong. `GetValue = _inner->GetValue() + _price`.
  - *Khuyến mãi (Strategy):* Class `Bill` chứa con trỏ `Strategy*`. `GetValue` gọi `Strategy` để tính giảm giá.

4 NHỮNG CÁI BẪY "CHẾT NGƯỜI"(PITFALLS)

1. **Quên virtual destructor**:
   - *Hậu quả:* Mất điểm quản lý bộ nhớ (Memory Leak).
   - *Phòng tránh:* `AllInOne.h` đã có, nhưng nếu viết class mới nhớ kiểm tra.
2. **Lỗi ép kiểu (dynamic_cast)**:
   - *Vấn đề:* Gọi hàm của con (VD: `GetCount`) từ con trỏ cha (`BaseObject*`) mà không ép kiểu.
   - *Hậu quả:* Lỗi biên dịch.
   - *Phòng tránh:* Luôn `dynamic_cast<Type*>(child)` trước khi gọi hàm lạ.
3. **Sai dấu phân cách chuỗi**:
   - *Vấn đề:* Đề dùng `|` nhưng code dùng `,`.
   - *Hậu quả:* Crash chương trình ngay khi chạy Factory.
   - *Phòng tránh:* Nhìn kỹ input file hoặc lệnh `Create` trong main.

4. **Nhầm lẫn Logic Singleton**:
    - *Xe tải/File:* Check **Tổng** (`current + new > max`).
    - *Nhân sự:* Check **Trùng** (`find(id)`).
    - *Kho hàng:* Check **Tồn kho** (`inventory[name] > 0`).
5. **Bẫy Factory**:
    - Nếu dữ liệu sai format → Phải `return nullptr` hoặc `throw`. Đừng để code chạy tiếp sẽ crash khi truy cập index mảng không tồn tại.

5 LỜI KHUYÊN CUỐI CÙNG

1. **Bước 1:** Chép `AllInOne.h` vào máy.
2. **Bước 2:** Dọc đề, xác định thuộc **Dạng mấy** (1 đến 6).
3. **Bước 3:** Copy khung code của dạng đó (từ bộ Solutions mẫu).
4. **Bước 4:** Đổi tên Class, đổi tên biến logic.
5. **Bước 5:** Chạy thử và sửa lỗi biên dịch.

CHÚC BẠN BÌNH TĨNH, TỰ TIN VÀ ĐẠT ĐIỂM TỐI ĐA!