

Severance

题目描述

刷新 ↻

在《人生切割术》（Severance）中，Lumon 公司开发出了一种名为“分离”（Severance）的记忆分割手术，接受手术的员工工作和个人生活记忆将被完全切割开，并受空间限制，形成公司人格和日常人格——到达分离层，只有工作的记忆；离开分离层后，只有个人生活的记忆。

分离层员工被分为不同部门，如宏观数据精简部（Macrodata Refinement, MDR）、光学设计部（Optics and Design, O&D）等。每个部门只能访问特定设备，且设备之间存在严格的依赖链（如访问 DataTerminal 前需解锁 SecurityPanel）。员工可通过晋升提升权限等级，但必须满足部门规则。你需要实现一个多员工操作系统，支持动态权限管理和设备依赖验证。

题目要求

给定 main.cpp、_Device.h、_Employee.h 和 Makefile 不可修改，你需要实现 Device.h、Device.cpp、Employee.h 和 Employee.cpp。

文件可以在[这里](/staticdata/problem/2144.YPJirzg555SwkbWz.pub/GjPXA3gTJiyzTmL4.severance.zip/severance.zip) (/staticdata/problem/2144.YPJirzg555SwkbWz.pub/GjPXA3gTJiyzTmL4.severance.zip/severance.zip) 下载。

1. 输入逻辑：

- **第一阶段：**输入 n ，创建 n 个员工，每个员工指定 **员工ID** 和 **部门与级别**：
 - 0：宏观数据精简部初级员工
 - 1：宏观数据精简部高级员工
 - 2：光学设计部初级员工
 - 3：光学设计部高级员工
- **第二阶段：**输入 m ，执行 m 个操作，每个操作指定 **员工ID**、**操作类型** 和 **设备类型**：
 - 操作类型：0（访问设备），1（晋升）
 - 设备类型：0（BaseLock），1（SecurityPanel），2（DataTerminal），3（OpticalTool）为方便读入，当操作类型为 1（晋升）时，仍然会提供设备类型充当占位符。

2. 系统设计：

- **继承与多态：**
 - 基类 Employee 定义虚函数 promote() 和 accessDevice(Device*)。
 - 四个派生类：MDRJunior / MDRSenior（宏观数据部）、OnDJunior / OnDSenior（光学设计部）。
- **组合关系：**
 - 每个员工组合一个 SecurityProfile 对象，包含 **安全等级**（1-Junior, 2-Senior）和 **已访问设备列表**。
- **设备依赖链：**
 - SecurityPanel 依赖 BaseLock，DataTerminal 依赖 SecurityPanel。
 - 设备基类 Device 定义 isUnlocked() 接口，检查依赖是否满足。

3. 权限验证：

- **部门权限：**宏观数据精简部员工只能访问 BaseLock、SecurityPanel、DataTerminal；光学设计部员工只能访问 OpticalTool。
- **安全等级：**Junior 员工只能访问基础设备（BaseLock、SecurityPanel），Senior 可访问全部设备。
- **依赖链：**访问设备前需满足依赖条件（如必须先访问 BaseLock 才能访问 SecurityPanel），每个员工已满足的依赖条件是独立的。

4. 动态晋升：

- Junior 员工晋升为 Senior 后，安全等级提升至 2，可访问高级设备。
- 晋升操作需替换员工对象类型（如 MDRJunior → MDRSenior）。

5. 输出要求：

- 访问成功：输出 [职位]([员工ID]): [设备执行结果]。
 - BaseLock：BaseLock engaged.
 - SecurityPanel：SecurityPanel activated.
 - DataTerminal：Processed [设备容量]TB data.
 - OpticalTool：OpticalTool adjusted.
- 访问失败：输出 [职位]([员工ID]): [失败原因]，原因包括：

- 部门权限或安全等级不足: No permission. ACCESS DENIED.
- 依赖未满足: Dependency not met. ACCESS DENIED.

优先输出靠前的失败原因。

- 晋升成功: 输出 [职位]([员工ID]): Promoted to [新职位]。
- 未尽要求请参见输入输出示例。

输入输出示例

输入:

```
3
0 0 // 员工0 (MDR Junior)
1 0 // 员工1 (MDR Junior)
2 2 // 员工2 (O&D Junior)
8
0 0 0 // 员工0访问BaseLock
0 0 1 // 员工0访问SecurityPanel
0 1 0 // 员工0晋升
1 0 2 // 员工1访问DataTerminal (权限不足)
0 1 0 // 员工0晋升 (已为最高等级)
0 0 2 // 员工0访问DataTerminal
2 1 0 // 员工2晋升
2 0 3 // 员工2访问OpticalTool
```

输出:

```
MDR-Junior(0): BaseLock engaged.
MDR-Junior(0): SecurityPanel activated.
MDR-Junior(0): Promoted to MDR-Senior.
MDR-Junior(1): No permission. ACCESS DENIED.
MDR-Senior(0): Already at the highest level.
MDR-Senior(0): Processed 3TB data.
O&D-Junior(2): Promoted to O&D-Senior.
O&D-Senior(2): OpticalTool adjusted.
```

注: 该样例为第一个测试点。

提示: 如必要, 可使用 if (dynamic_cast<MyClass *>(ptr)) 来判断 ptr 指向的对象是否是 MyClass 或其派生类的实例。

语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	oop_custom	make		1048576 B

递交历史

#	状态	时间
329115	Accepted	2025-04-10 09:13:50
329112	Wrong Answer	2025-04-10 09:10:41
329111	Wrong Answer	2025-04-10 09:10:11
329107	Wrong Answer	2025-04-10 09:04:26
329105	Wrong Answer	2025-04-10 09:01:23

1

当前没有提交权限!