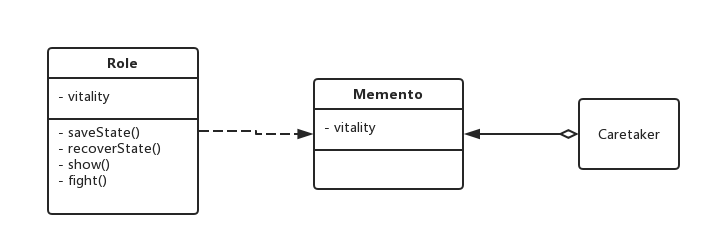
系统要求

1. 现在假设游戏角色中有多个个属性，角色名称和角色生命力，攻击力，和防御力，现在要进行的保存游戏角色的生命力，攻击力，和防御力。初始三种力量都是100，和大BOSS打完以后，损耗严重，各项指标下降到0；因此需要进行备份恢复到初始状态；请从封装性的角度考虑，实现上述的整个过程。

UML类图



类图使用PrcessOn（https://www.processon.com）在线生成工具完成。

程序采用备忘录模式进行设计。

主要包含三个类，Role, Memento。各类之间关系见上图，其中Caretaker为简单类。

Role类用于对游戏角色的抽象，包含四个方法，一个属性：需要注意的是，由于题目中的三个属性逻辑上相同，因此这里我们只使用一个属性；

一个属性为：

1. vitality：标定其角色的生命力；

四个方法为：

1. show()：显示角色状态；
2. fight(): 模拟角色打斗过程，状态归零；
3. saveState(): 存储进度；
4. recoverState(): 读取进度；

Memento类为角色备忘录的抽象，，一个属性；

一个属性为：

（1）vitality：标定该进度中角色的生命力；

系统实现

该版本的系统使用C++实现。工程包含五个文件（main.cpp memento.cpp memento.h role.cpp role.h）。

其中：

1. main.cpp: 包含游戏过程简单实现，包括存储进度、读取进度、战斗开始；
2. role.cpp & role.h: 包含游戏角色类的实现，包括角色状态的设定、状态显示、战斗模拟、进度的读写；
3. memento.cpp & memento.h：包含游戏进度类的实现，保存某一状态下的角色信息;

游戏存档系统实现过程：游戏开始后显示角色的初始状态，此时存储进度保存角色当前状态，后角色开始打斗，状态清零并显示，随后角色读取进度，打印出角色读取进度后的状态。

由于时间有限，需要注意的实现细节如下：

1. 由于逻辑上的相似性，实现过程中我们将题目中描述的角色的三个属性选择一个，即生命力进行实现；
2. 本版本没有将设计类图中的Caretaker这一进度管理类进行封装实现，这是由于题目实现过程中不涉及到复杂的进度管理，因此我们进行了相应的简化；

系统运行结果

