

象来调用对应的公式计算算法。

流程图可以让我们了解对象之间的互动流程。如图 10-4 所示为敌方阵营 IEnemy 攻击玩家阵营 ISoldier 时的攻击流程。

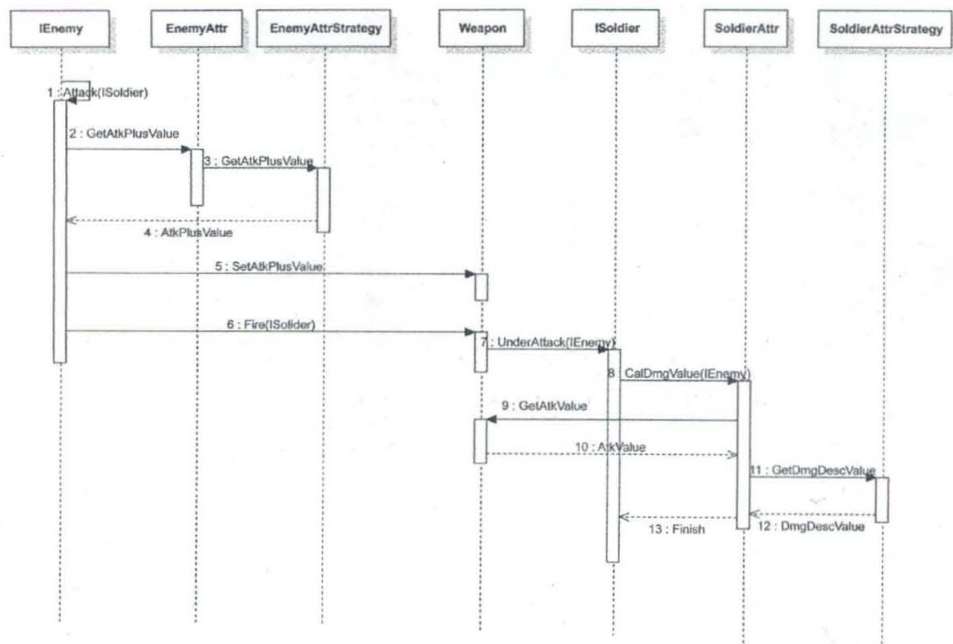


图 10-4 敌方阵营 IEnemy 攻击玩家阵营 ISoldier 时的攻击流程

10.3.3 使用策略模式 (Strategy) 的优点

将角色属性计算运用策略模式 (Strategy) 有下列优点:

让角色属性变得好维护

对于改进后的角色类 ICharacter 来说, 将角色属性有关的属性以专属类 ICharacterAttr 来取代, 可以使以后角色属性变动时, 不会影响到角色类 ICharacter。此外, 随着游戏需求的复杂化, 加入更多的角色属性是可预期的, 所以让角色属性集中在同一个类下管理, 将有助于后续游戏项目的维护, 也可以减少角色类 ICharacter 的更改及降低复杂度。

不必再针对角色类型编写程序代码

通过 ICharacterAttr 与其子类的分工, 将双方阵营的属性放置于不同的类中。对于角色类 ICharacter 而言, 使用 ICharacterAttr 的对象引用时, 完全不用考虑将使用哪一个子类对象, 避免了使用 switch case 语句的编写方式及后续可能产生的维护问题。当有新的阵营类产生时, 角色类 ICharacter 并不需要有任何改动。