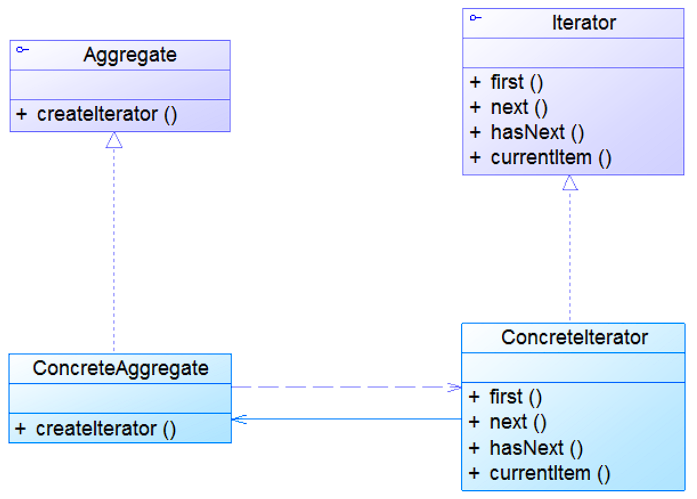
1. 模式定义

迭代器模式（Iterator Pattern)：提供一种方法来访问聚合对象，而不用暴露这个对象的内部表示，其别名为游标（cursor）。迭代器模式是一种对象行为型模式。



**二、**模式动机

一个聚合对象，如一个列表（list）或者一个集合（set)，应该提供一种方法来让别人可以访问它的元素，而又不需要暴露它的内部结构。

针对不同的需要，可能还要以不同的方式遍历整个聚合对象，但是我们并不希望在聚合对象的抽象层接口中充斥着各种不同遍历的操作。

怎样遍历一个聚合对象，又不需要了解聚合对象的内部结构，还能够提供多种不同的遍历方式，这就是迭代器模式所要解决的问题。

在迭代器模式中，提供一个外部的迭代器来对聚合对象进行访问和遍历，迭代器定义了一个访问该聚合元素的接口，并且可以跟踪当前遍历的元素，了解哪些元素己经遍历过而哪些没有。

有了迭代器模式，我们会发现对一个复杂的聚合对象的操作会变得如此简单。

三、例子

电视机遥控器，这个模式一般在C++少用，C++有STL库，支持迭代器，是模板实现编译时多态。

四、优缺点

优点：

·它支持以不同的方式遍历一个聚合对象。

·迭代器简化了聚合类。

·在同一个聚合上可以有多个遍历。

·在迭代器模式中，增加新的聚合类和迭代器类都很方便，无须修改原有代码，满足“开闭原则”的要求。

缺点：

·由于迭代器模式将存储数据和遍历数据的职责分离，增加新的聚合类需要对应增加新的迭代器类，类的个数成对增加，这在一定程度上增加了系统的复杂性。

五、适用环境

·访问一个聚合对象的内容而无须暴露它的内部表示。

·需要为聚合对象提供多种遍历方式。

·为遍历不同的聚合结构提供一个统一的接口。

六、示例代码