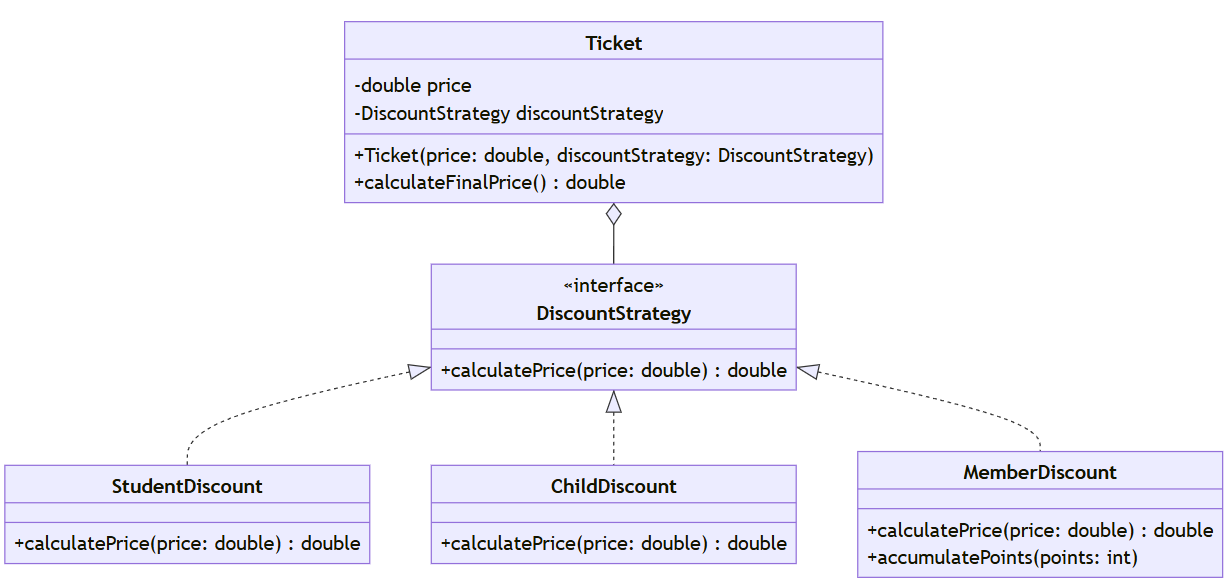
**题目：**

小王正在开发一套电影院售票系统，在该系统中需要为不同类型的用户提供不同的电影票打折方式，具体打折方案如下：(1) 学生凭学生证可享受票价7折优惠；(2) 年龄在10周岁及以下的儿童可享受每张票减免15元的优惠（原始票价需大于等于30元）；(3) 影院会员卡用户除享受票价半价优惠外还可进行积分，积分累计到一定额度可换取电影院赠送的奖品。该系统在将来可能还要根据需要引入新的打折方式。请你针对上面的描述，帮助小王进行设计，给出设计思路及所采用的设计模式，画出类图，并写出关键代码。

**设计思路与设计模式：**

定义一系列算法（折扣策略），并将每个算法封装成独立的策略类。通过**策略模式**，可以让系统在运行时选择合适的折扣策略，而不必修改现有代码。这样设计可以方便地添加新的打折策略。

**类图：**



|  |
| --- |
| **代码** |
| *// 折扣策略接口*  interface DiscountStrategy {      double calculatePrice(double price);  }  *// 学生折扣策略类*  class StudentDiscount implements DiscountStrategy {      @Override      public double calculatePrice(double price) {          System.out.println("Applying student discount: 70%");          return price \* 0.7;      }  }  *// 儿童折扣策略类*  class ChildDiscount implements DiscountStrategy {      @Override      public double calculatePrice(double price) {          if (price >= 30) {              System.out.println("Applying child discount: -15");              return price - 15;          } else {              System.out.println("No child discount available for prices under 30.");              return price;          }      }  }  *// 会员卡折扣策略类*  class MemberDiscount implements DiscountStrategy {      private int points;      public MemberDiscount() {          this.points = 0;      }      @Override      public double calculatePrice(double price) {          System.out.println("Applying member discount: 50%");          accumulatePoints(10);  *// 每次购买增加10点积分*          return price \* 0.5;      }      public void accumulatePoints(int points) {          this.points += points;          System.out.println("Accumulated points: " + this.points);      }  }  *// 电影票类*  class Ticket {      private double price;      private DiscountStrategy discountStrategy;      public Ticket(double price, DiscountStrategy discountStrategy) {          this.price = price;          this.discountStrategy = discountStrategy;      }      public double calculateFinalPrice() {          return discountStrategy.calculatePrice(price);      }  }  *// 测试类*  public class CinemaTicketSystem {      public static void main(String[] args) {  *// 学生票*          Ticket studentTicket = new Ticket(100, new StudentDiscount());          System.out.println("Final price for student ticket: " + studentTicket.calculateFinalPrice());  *// 儿童票*          Ticket childTicket = new Ticket(35, new ChildDiscount());          System.out.println("Final price for child ticket: " + childTicket.calculateFinalPrice());  *// 会员票*          Ticket memberTicket = new Ticket(80, new MemberDiscount());          System.out.println("Final price for member ticket: " + memberTicket.calculateFinalPrice());      }  } |