Nhận xét sơ qua: Nhóm đã bám theo sơ đồ thiết kế chặt chẽ và đã cố gắng code theo SOLID ngay từ đầu nên căn bản code gần như không phải sửa mấy.

1. Thêm các loại thẻ vé
   1. Thẻ tháng:

Bước thêm:

* Trong package certificate
  1. Thêm class MonthlyCard, extends class Hour24Ticket, override hàm getStatus() để phù hợp với cách kiểm tra thẻ còn hay hết hạn.
  2. Thêm helper MonthlyCardHelper implements interface Helper.
* Trong package checkValid, thêm class AFCCheckValidOfMonthlyCard, và implement interface AFCCheckValidOfCertificate.

Giải thích: Bản chất của thẻ tháng không khác gì với vé 24h về mặt properties nhưng lại khá khác nhau về cách kiểm tra tính hợp lệ, nên ta sẽ kế thừa cái có sẵn là class Hour24Ticket và tạo mới các class còn lại cần thiết.

* 1. Vé cuối tuần:

Bước thêm:

* Trong package certificate
  1. Thêm class WeeklyTicket, extends class Hour24Ticket, override hàm getStatus() để phù hợp với cách kiểm tra vé còn hay hết hạn.
  2. Thêm helper WeeklyTicket Helper implements interface Helper.
* Trong package checkValid, thêm class AFCCheckValidOfWeeklyTicket, và implement interface AFCCheckValidOfCertificate.

Giải thích: Một lần nữa, bản chất của Vé cuối tuần không khác gì với vé 24h mặt properties nhưng lại khá khác nhau về cách kiểm tra tính hợp lệ, nên ta sẽ kế thừa cái có sẵn là class Hour24Ticket và tạo mới các class còn lại cần thiết.

1. Thay đổi cách tính phí:
   1. Thay đổi các mốc, hạn mức giá trị:

Cách làm: Vào file Config trong class certificate và thay đổi thông số tương ứng:

***PREPAID\_MINIMUM\_BALANCE***

***BASED\_FARE***

***BASED\_DISTANCE***

***DEFAULT\_ADDITION\_DISTANCE***

***ADDITIONAL\_FARE***

* 1. Không tính phí đi lại trên khoảng cách, mà tính dựa trên zone của hai nhà ga:

Bước thêm:

* Trong package station
  1. Thêm class StationHelper2, extends class StationSystem và implements interface SystemHelper. (Khác nhau duy nhất ở hàm này và hàm calculateDistance vì cơ chế tính tiền khác nhau)
* Trong package transactions
  1. Người mua chương trình muốn dùng cách tính tiền nào thì thay bằng class tương tự, thay thành helper tương ứng trong class TransactionHistory của package transactions

StationHelper helper = **new** StationHelper();

1. Có nhiều tuyến tàu (line). Các tuyến này có thể cắt nhau.
   1. Tạo tuyến tàu (StationSystem1, StationSystem2, …) trong package stations.
      * Vì StationSystem hiện tại giống như là một tuyến tàu vip (đi qua mọi trạm), các tuyến khác thì cũng tương tự, chẳng qua là ít số lượng ga đi, nên là hàm constructor của các hàm system phải trên phải sửa sao cho đúng với loại ga đi qua.
      * Các hàm trong class sẽ là: kiểm tra xem tuyến có đi qua ga X không.
      * Sau đó tự tạo thêm class Helper implements interface Helper cho mỗi tuyến đó. (Mỗi tuyến có cách tính tiền khác nhau)
   2. Các tuyến có thể cắt nhau.
      * Các tuyến có chung điểm dừng thì có thể cắt nhau.
      * Trong các helper giờ phải có thêm function kiểm tra có qua trạm X không.
      * AFCController giờ sẽ là fat data aggregation, tụ hội lại chỗ helper trên và cho người dùng chọn một trong các helper đó để đi. (như là chọn tuyến để mà đi).