

Observer pattern:

Finns det något ställe i er design där detta används?

Vi använder detta, oavsiktligt, i vår design då vi har actionListeners i vår CarController klass. Detta löser problemet att veta när användaren faktiskt trycker på knappen, och endast då sker något. Vi kan också ändra funktionen på knapparna lättare då actionListeners endast kollar när knappen trycks på, men anropar en annan metod för att något faktiskt ska ske.

Kan detta förbättra er design?

Det hade kanske kunnat användas för att studsas bilarna när de når slutet av rutan, genom att den meddelar när bilen får vissa specifika koordinater och då kalla på en metod som åtgärdar detta. Men det känns som att man gör det mer komplicerat än vad det behöver vara, då man ändå måste hårdkoda en hel del oavsett.

Factory method:

Finns det något ställe i er design där detta används?

Vi använder oss inte av detta

Kan detta förbättra er design?

Vi skulle definitivt kunna använda oss av detta genom att skapa en factory metod i alla subklasser till car som vi tvingar ska finnas genom att skapa en abstrakt klass Creator som innehåller en abstrakt metod som skapar objekten. Detta skulle förbättra vår modell genom att insynen i vår design minskar

State:

Finns det något ställe i er design där detta används?

Vi använder oss inte av detta

Kan detta förbättra er design?

Vi skulle kunna skapa en state metod/interface för plattformen i Carrier, men då detta är en redan väldigt okomplicerad metod som finns så ser vi inte att vi vinner något på det. Vi kan inte skapa ett gemensamt interface som både plattformen i Scania och Carrier använder sig av då de har helt olika metoder som ger olika output.

Composite Pattern:

Finns det något ställe i er design där detta används?

Vi använder oss av detta på flera ställen i koden, bland annat för alla gas, break, turboOn ... metoder i CarController, där vi itererar över en lista och gör samma sak med varje enskilt objekt i hela listan. Vi gör detta oavsiktligt.

Kan detta förbättra er design?

Även om vi använt själva designmönstret oavsiktligt så har vi varit medvetna om att funktionen finns i Java och har därmed redan använt det på alla ställen vi tycker är lämpliga, och ser ingen stans i vår kod där detta hade förbättrat designen där vi i nuläget inte använder det redan.