

Analysera de beroenden som finns med avseende på cohesion och coupling, och Dependency Inversion Principle.

- Vilka beroenden är nödvändiga?

Car och Truck ärver från Vehicle, vilket är nödvändigt eftersom det är ett "is a"-förhållande

Scania och mercedes ärver från Truck men det behövs för att det är ett "is a" förhållande.

Saab och volvo ärver från car men det behövs för att det är ett "is a" förhållande.

- Vilka klasser är beroende av varandra som inte borde vara det?

Ingen vi behöver allt.

- Finns det starkare beroenden än nödvändigt?

CarController och DrawPanel är väldigt starkt beroende av varandra, vilket är en förbättringspunkt.

- Kan ni identifiera några brott mot övriga designprinciper vi pratat om i kursen?

DIP följs inte helt då vår carController modifierar bilarnas position direkt i DrawPanel

Single Responsibility Principle för att t.ex så gör carController för mycket.

Analysera era klasser med avseende på Separation of Concern (SoC) och Single Responsibility Principle (SRP).

- Vilka ansvarsområden har era klasser?

Vehicle håller attribut som är gemensamma för alla fordon

Alla klasser som ärver från Vehicle håller attribut och metoder för vad det specifika fordonet kan göra

AngularLift, OnOffLift och TransportationBed ansvarar för flakens funktionalitet

CarController ansvarar för billogik, gränssnitt och timern

DrawPanel ansvarar för att rendera grafiken för bilarna, samt lagrar och håller reda på bilarnas position

CarView ansvarar för attribut som knappar och fönster

- Vilka anledningar har de att förändras?

CarController och DrawPanel har båda fler än ett ansvarsområde, vilket strider mot SRP.

- På vilka klasser skulle ni behöva tillämpa dekomposition för att bättre följa SoC och SRP?

I CarController kan vi dela upp ansvarsområdena i helper klasser, som t.ex CarMovementHandler, samma i DrawPanel

Refaktoriseringsplan:

CarController:

1. Flytta rörelse logiken till CarMovementHandler.
2. Skapa CarInputHandler för att hantera input.
3. Skapa CarUpdateHandler för att kunna uppdatera bilarnas position.
4. Skapa en class main
5. Skapa en klass för Carfactory

DrawPanel:

1. Skapa CarPositionTracker för att man ska kunna hantera positionerna.
2. Skapa CarRenderer som målar bilarna.