

Vilka ansvarsområden har era klasser?

- Car: Bygga grunden för en bil
- Volvo: Bygga en fungerande Volvo
- Saab: Bygga en fungerande Saab
- Trucks: Grunden till en lastbil
- Scania: Bygga en fungerande lastbil
- Carrier: Bygga en fungerande biltransport
- Carcontroller: Styra bilen och skapa bilarna i simulatorn (ej srp)
- Carrepairshop: Skapar en bilverkstad
- Carview: Skapar alla knappar grafiskt och säger vad ett knapptryck ska resultera i
- Drawpanel: Målar upp det grafiska
- Movable: Bestämmer vilka rörelsefunktioner en bil ska ha

Vilka anledningar har de att förändras?

- carcontrollers inbyggda klass timerlistener borde flyttas ut till sin egna fil
- Carview borde inte skapa knapparna, utan det borde en separat klass göra
- Carview kan ha ett beroende till carcontroller, men carcontroller borde inte vara beroende av Carview

På vilka klasser skulle ni behöva tillämpa dekomposition för att bättre följa soc och srp?

- Carcontroller behöver tillämpas dekomposition på
- Även Carview behöver det

Refaktoriseringsplan

1. Flytta ut TimerListener till sin egna fil
2. Skapa en ny klass CarViewGraphics
3. Lägg in allt grafiskt skapande från CarView till CarViewGraphics
4. Flytta initComponents till CarViewGraphics
5. Skapa en constructor i CarViewGraphics som skapar en ny frame och skapar ett nytt objekt carController som kallar på Main funktionen
6. Ta bort frame objektet i Carcontroller

En person kan göra steg 1 och en annan person steg 2-6

Varför är detta bättre?

1. Detta blir bättre för vi använder oss av funktionell nedbrytning för att få en mer modulär design där strukturen är tydligare
2. Detta ökar till att CarView då enbart får ett ansvarsområde istället för två
3. Samma som ovan
4. Samma som ovan
5. Detta tar bort beroendet som finns från CarController till CarView och skapar enbart ”envägsberoende” vilket vi inte hade innan
6. Samma som ovan