

Ansvarsområden som klasserna har:

CarController – Kontrollera bilens simulering, initialisering av bilarna

DrawPanel – Ritar upp panelen. Hanterar bilder av bilar, samt deras position.

CarView – Skapar UI och kopplar knapparna till CarController

CommonBaseCar – Biltypernas gemensamma egenskaper. Hanterar rörelse, riktning och motorns funktion.

Workshop, VolvoWorkshop – Tar emot och returnerar fordon.

CarTransport-, GeneralFlap-, Flap – Hanterar flaket och dess funktion på lastbilar.

Saab95, Scania, SmallCar, Volvo240 – Representerar specifika bilmodeller med sina funktioner

Förändringar:

- CarView & CarController är beroende av varandra (Aggregation) och bör därför delas upp i mindre klasser. Detta kan man lösa genom att skapa en ny klass som skapar objekt av CarView & CarController så att dessa inte beror på varandra (Dependency Inversion principle)
- CarView, CarController & DrawPanel har i tillfället väldigt många ansvarsområden och därför bör ansvaret delas upp till mindre klasser. (Separation of Concern principle) (Dekomposition)
- CarTransport – loadCar och isInVicinity kan hanteras i separata klasser som sedan används av CarTransport, Scania och Workshop. (Single Responsibility Principle)

Refaktoriseringsplan:

- Skapade en ny klass CreateNewGraphics som gör instanser av DrawPanel, CarView & CarController. Detta minskar beroenden mellan dessa klasser och gör att de istället är beroende av CreateNewGraphics.
- Loading, RadiusCheck: Flyttat ut radius-, samt load-metoder från klasser som loadar bilar. Detta minskar kodduplicering då flera klasser tidigare använde liknande metoder.
- Skapade en ny klass CreateNewCar som gör instanser av alla typer av bilar.
- Delade upp GeneralFlap i två nya klasser, TwoStatePlatform & AngleFlap