

# Refaktorseringsplan

En refaktorisering plan för laborations projekt skriven av grupp 18.

Refaktoriseringplan för vår egna kod:

- Skriv en ny klass "GPS" som ska hantera point klassen och enum direction (**Open/closed**). Då är den öppen för förändring i framtiden.
- Super klassen Car ska implementera GPS istället för både point och direction.
- Ta bort move() från cartransport och uppdatera move() i car klassen med en if-clause för truckload position.
- Skriv om collision, göra mer generell så att vi kan använda den till workshop.
- Flytta move(), turnLeft() och turnRight() till transform klassen.

Refaktoriseringplan för angiven kod:

- Skapa en UI klass som hanterar designen av UI med färger och placering osv. Så hanterar CarView vilken påverkan knapparna har på objekten (SRP/SoC). Lättare att underhålla och uppdatera för att det hålls separerat.
- Flytta moveit() till CarController och skapa workshop objekt i CarController. Drawpanel använder sig aldrig av moveit(). (ISP)
- Flytta gas, brake osv till carview för att carview enbart ska hantera funktionaliteten av knapparna enligt SRP. Då försvinner cirkulära "dependencies" mellan carcontroller och carview. Klassen blir då mer återanvändningsbar och dess beteende är lättare att hänga med i och underhålla.

## KONSEKVENSER

Koden blir tydligare sorterad. Så att potentiella framtida förändringar kommer bli lättare att införa. I och med att vi har färre beroenden och varje klass hanterar en grej. Så om vi förändrar en klass i framtiden kommer det att ge konsekvenser utanför godtycklig klass.