

Refaktoriseringsplan

Det som behöver fixas:

- ItemFactory
- Dela upp CarController
- BufferedImage inte del av Car - ny klass som håller koll på bilder
- DrawPanel har beroende på item istället för car
- Lägga till HasBed

Planen bör bestå av en sekvens refaktoriseringssteg som tar er från det nuvarande programmet till ett som implementerar er nya design. Planen behöver inte vara enormt detaljerad.

- Implementera interfaces för att uppnå en högre abstraktionsnivå tex HasTurbo, HasBed. **Dependency inversion principle**
- Dela upp klasser som har för stora ansvarsområden - tex CarView har delats upp till en ControlPanel och en CarView. **Single responsibility principle**
- Skapa en factory (ItemFactory) för att minska coupling (alltså öka de interna beroendena för att minska de externa), och på så sätt få en bättre struktur på programmet. **Low coupling high cohesion, dependency inversion principle**
- Flyttar beroenden, som det mellan car och BufferedImage, så att man skiljer på modulen och det visuella. **Interface segregation principle** - en bil ska inte behöva ha en bild. (det är nu istället ett beroende på imagehandler). Detta möjliggör också att annat än bilar (tex workshop) kan ha bilder. **MVC och dependency inversion principle, open-closed principle**

Tänkte göra ett interface till car så att CarUpdater har ett beroende till ett interface istället, för att skapa ett svagare beroende (beroende mellan klass och interface kontra klass och klass), men eftersom car är en abstrakt klass är detta beroende redan svagt, vilket innebär att det inte finns ett behov för ett interface

Finns det några delar av planen som går att utföra parallellt, av olika utvecklare som arbetar oberoende av varandra? Om inte, finns det något sätt att omformulera planen så att en sådan arbetsdelning är möjlig?

- För att implementera interfaces krävs en förståelse för programmets generella struktur, dock kan utvecklare skapa enskilda interfaces så länge de har förståelse för enskilda klasser, vilket innebär att större delar av refaktoriseringsplanen går att utföra parallellt av olika utvecklare.