Vilka avvikelser har vi från MVC i det ursprungliga programmet?
DrawPanel har lagrar just nu data om var alla bilar och verkstaden är. DrawPanel är en del av, enligt MVC, view. View ska inte behöva lagra någon data om de olika komponenterna. DrawPanel bör därför bli dummare då den är beronde av vissa klasser exempelvis Car och indirekt Saab95, Volvo240 etc. Ifall ett program med cyklar och skateboards hade implementerats hade i en ny view hade den inte kunnat kopplas på det befintliga programmet.

CarView är relativt avgränsad i sig men den bör inte ha namnen "View" då det uppenbart tillhör kategorin "Controller" enligt MVC. Denna bör göras tunnare då den enbart bör hantera user input från knapparna.

CarController är den klass med flest problem. Först och främst är det inte en Controller utan en Model enligt MVC. Den har main och en ActionListener vilket gör att den blir beroende av precis allt i hela programmet vilket inte är särskilt åtråvärt. Den lagarar även information om de olika bilarna vilket enligt definitionen av Model i MVC är okej men kan med fördel separeras från logiken (CarController). CarController har även en DrawPanel på grund av Main-funktionen vilket inte är nödvändigt. Denna behövs göras smartare och mer uppdelad.

Vilka av dessa brister åtgärdar vi i vår nya design?
Den nya klassen VehicleData ökar cohesion hoss model-paketet i programmet samt delar upp ansvaret bättre mellan de olika klasserna. Samma sak gäller för RunGame i det att det minskar ansvaret för klassen CarController. Samtidigt tydliggör vi en ny del av MVC villket är applikationer som inte är en del MVC mönstret.

Vad åtgärdar vi inte?
CarView hanterar fortfarane en viss mängd logik. Koppling mellan CarController och DrawPanel med ett interface.