

Inlämningsuppgift 7 – UML & Applet

Denna inlämningsuppgift består av tre deluppgifter. Du ska få prova på att skapa ett klassdiagram utifrån befintlig kod, skapa kod från ett klassdiagram samt konvertera en tidigare applikation till en applet. För att skapa klassdiagram kan modelleringsverktyget Violet (<http://violet.sourceforge.net>) användas. Se separat dokument i Moodle (kursbeskrivning) för en kort genomgång hur Violet används.

Spara din lösning för varje deluppgift i separata kataloger på din hårddisk (förlagsvis döpta till deluppgift 1, deluppgift 2 och deluppgift 3). När du sen skickar in din lösning paketerar du alla tre kataloger i samma zip-fil och döper den enligt tidigare instruktioner.

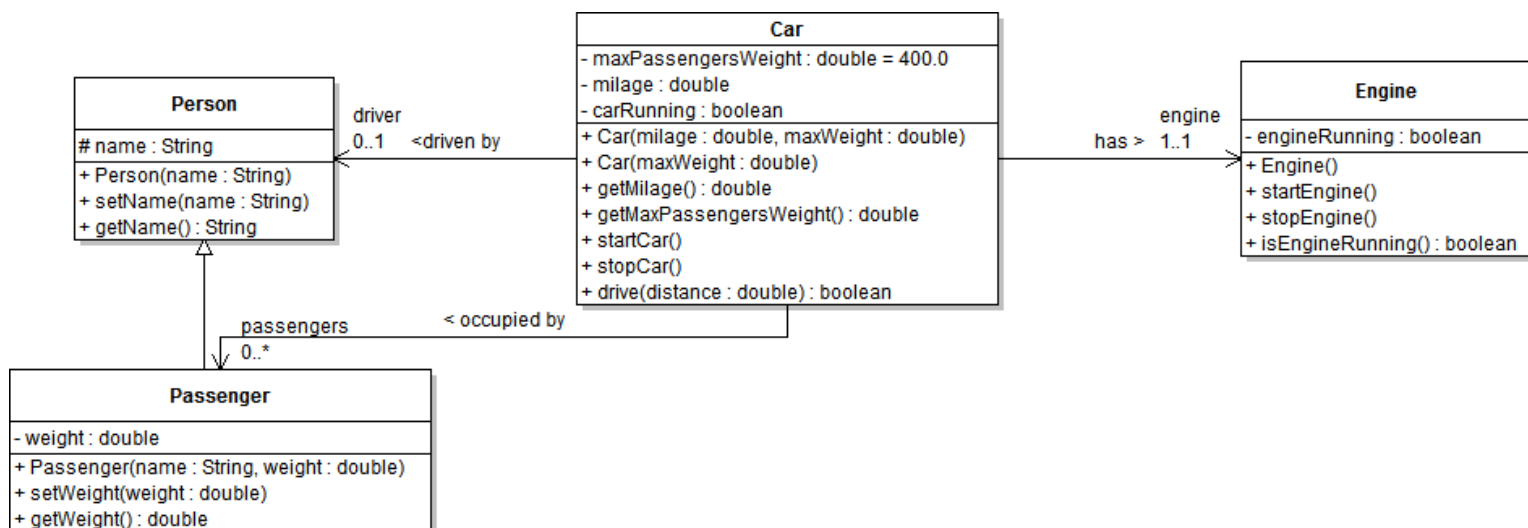
Deluppgift 1

Här ska du skapa ett klassdiagram över de klasser som utgör din lösning på inlämningsuppgift 1 (bankuppgiften). Klassdiagrammet ska minst innehålla klasser (och eventuella gränssnitt) för kund och alla kontotyper. Det är valfritt om du även vill ta med övriga klasser i din lösning. Tänk på att använda rätt typ av relation mellan klasserna samt ange korrekta multipliciteter. Alla instansvariabler, konstruktorer och metoder ska finnas med i klassdiagrammet. Även returvärden (som inte är void) och parametrar till metoder/konstruktorer ska finnas med.

Skicka in din lösning som en bild-fil (.png eller .jpg) eller som ett dokument (.pdf eller .doc) där du klistrat in ditt klassdiagram. Använder du Violet får du gärna skicka med din sparade fil i Violet. Skicka även med källkod för de klasser du modellerat i klassdiagrammet (det vill säga dina .java från inlämningsuppgift 1).

Deluppgift 2

Här ska du utifrån klassdiagrammet nedan skapa motsvarande kod i Java.



Koden ska vara fullständig och körbar. Det är mycket upp till dig hur du väljer att implementera all funktionalitet. Till exempel att när bilen startas eller stoppas ska även bilens motor startas och stoppas. För att köra bilen måste bilen vara igång och det måste finnas en person som är förare. När bilen körs en viss sträcka (distance) ska bilens mätarställning uppdateras (milage). När passagerare

läggs till ska passageraren inte läggas till om det innebär att den totala vikten på alla passagerare överstiger bilens maximala vikt. Var extra observant så att du översätter multipliciteterna på korrekt sätt. Skapa en tesklass som kort demonstrerar användningen av klasserna i klassdiagrammet. Om du anser det saknas metoder i någon klass för att få en fungerande lösning, lägg då till dessa. Olika utskrifter från körning av testklassen väljer du själv.

Skicka in all din källkod. Om du gjort speciella antaganden eller tillägg som du vill förklara, skriv dessa i ett separat dokument (.doc eller .pdf) och skicka in den tillsammans med källkoden.

Delövning 3

Här ska du ta ditt ritprogram från inlämningsuppgift 6 och göra om den till en applet. Paketera alla class-filer i en Jar-fil som html-filen sen använder (i applet-taggen). När du skickar in din lösning kom ihåg att lägga till html-filen, jar-filen samt alla källkodsfiler.