

Objektorienterad reseplanerare

Denna uppgift går ut på att implementera uppgiften "Reseplanerare" från Fas 0 / Sprint 2 men i Java. Uppgiften riktar sig främst till dig som redan implementerat uppgiften i C, men det är inte ett krav.

Poängen med att implementera om reseplaneraren i ett objektorienterat paradigm är att det sätter fingret på skillnaderna i tankesätt i procedurell programmering och OO-programmering, och också på hur de tar sig ut i den faktiska programmeringen.

Genom att lösa om ett problem behöver du inte fokusera så mycket på själva nedbrytningen i delproblem, utan kan fokusera på *mappningen av delproblemen till klasser och objekt*.

Objektorienterad programmering handlar om samverkande objekt. Klasser beskriver hur objekt fungerar, litet analogt med hur en strukt beskriver layouten i minnet för ett sammansatt data i C. En viktig skillnad är dock att klasser inte bara har tillstånd (jmf. poster i strukten) utan också beteende.

Objektorienterad analys går (mycket förenklat) ut på att hitta objekten i domänen. Om jag t.ex. vill simulera en fotbollsmatch hittar jag objekten spelare, domare, fotboll, plan, etc. Objekt är ofta uppbyggda av andra objekt, så om jag analyserar ett spelarobjekt hittar jag förmodligen delobjekten namn, skostorlek, mål, assists, etc. Alternativt (och detta beror ju helt på vad det är för sorts program) en mängd armar och ben, etc.

När man bygger en reseplanerare är vissa av objekten kanske mer abstrakta, t.ex. en bussrutt eller en resa, men kanske även en hållplats eller en buss. (Vad är rimligt? Vad behövs?)

Tänkvärt:

1. Vilka objekt finns i domänen som man modellerar?
2. Vilka objekt finns i programmet?
3. Vilka klasser finns, och hur är de relaterade med avseende på arv och aggregering?
4. Java har stöd för inkapsling som saknas i C – hur kan detta användas fördelaktigt i denna uppgift?
5. Vad skiljer Java-programmet från C-programmet? Vad är bättre/sämre och varför?
6. Vad finns det för hjälpklasser i Javas standardbibliotek som du kan dra nytta av?

OBS! Målet med denna uppgift är att tänka objektorienterat! OBS!

Att översätta C-koden till Java-kod är inte samma sak – det går att programmera procedurellt i Java, på samma sätt som man kan programmera objektorienterat i C.

För dig som redan är bekant med Java: Är du redan bekant med Java är det rimligt att tänka sig att du använder dig av Javas genericitet (parametrisk polymorfism) vid implementationen av de klasser som bygger upp nätverket av busslinjer.