

Implementera de klasser som behövs för en resväska enligt Composite. Den klass som kallas Component i mönstrets vanliga beskrivning är en abstrakt klass från vilken klasserna för de enskilda prylarna och klassen för sammansättningarna ärver.

Skriv tre klasser: En abstrakt klass, en klass för löven och en klass som ska vara behållare. Låt ett String-attribut ange vilken pryl som representeras av objektet. Behållarklassens prylnamn blir t.ex. resväska, necessär, påse och exempel på "lövprylar" är jeans, T-shirt, tvål, schampoo.

Minst tio saker ska packas och minst tre olika nivåer av sammansättningar ska användas. T.ex. kan hårspännen ligga i en påse som ligger i en necessär tillsammans med tvål och schampoo och necessären kan ligga i resväskan bredvid större klädesplagg.

Utöver namnet, låt alla komponenter (klassen Component i mönstret) ha en vikt som instansvariabel. Båda attributen ges värden när objekten skapas genom en vanlig konstruktor.

Låt alla komponenter ha operationerna `getWeight()` och `toString()`. `getWeight()` för en liten pryl (löv) returnerar endast prylens vikt medan `getWeight()` för en behållare returnerar hela behållarens vikt (egenvikten plus summan av vikterna av allt den innehåller). `toString()` för en liten pryl returnerar namnet medan `toString()` för en behållare ska returnera en String med behållarens namn följt av namnen på alla saker som finns i behållaren.

Skriv ett testprogram som bygger upp en resväska. Resväskans totala vikt ska beräknas i ett metodanrop, t.ex. `suitcase.getWeight()` och skrivs ut (ingen utskrift i metoden förstås). Tänk på att själva väskan och andra behållare har egen vikt utöver vikten för innehållet! Skriv också ut "hela resväskan". Det ska ge en lista över innehållet. Tag sen bort några komponenter ur innehållet. Skriv ut vikt och innehåll igen och kolla att det är rätt!

`getWeight()` och `toString()` ska implementeras enligt mönstret.

Det innebär att varje anrop på ett Composite-objekt ska gå igenom alla objektets barn.