

Inlämningsuppgift 1

Denna uppgift är den första (av två) obligatoriska inlämningsuppgiften på delkursen PROG2 Programmering 2 VT2016. Uppgiften ska lösas enskilt eller i grupper om högst tre personer. Inlämningsinstruktion finns längst bak i det här dokumentet.

Uppgiften är en variant av den uppgift som löstes på kursen OOP fram till hösten 2013, då arv och klasshierarkier ingick i OOP. Den går ut på att skriva ett program för ett register av värdesaker som ska kunna registreras och sedan listas i olika ordning.

Uppgiften beskrivs här i tre delar, så som det är tänkt att man löser uppgiften.

Del 1

Deklarera klasser i en klasshierarki för att representera värdesaker. På sikt ska det kunna finnas många typer av värdesaker, men just nu är tre typer aktuella: smycken, aktier och apparater. Alla värdesaker har ett namn, dessutom har de olika egenskaper som är specifika för typen. För varje värdesak ska man kunna beräkna dess värde, fast det går till på olika sätt beroende på vilken värdesak det är:

Smycken: ett smycke har ett namn (t.ex. "Ring", "Halsband" etc.) och hur många ädelstenar det innehåller, det kan vara av guld eller av silver. Om smycket är av guld så är dess värde 2000, är det av silver så är värdet 700. Dessutom ökar värdet med 500 för varje ädelsten,

Aktier: en aktiepost har ett namn ("Ericsson B", "Asea" etc.), ett antal samt en kurs. Värdet på en aktiepost är antalet * kursen. Kursen hos en aktie ska kunna ändras.

Apparater: har ett namn ("TV", "Dator", "iPhone" etc.), ett inköpspris och ett slitage (ett tal mellan 1 och 10 där 10 betyder helt ny och 1 betyder mycket sliten). Värdet på en apparat beror på slitaget: är slitaget 10 så är värdet lika med inköpspriset, är slitaget 9 så är värdet 90% av inköpspriset, är slitaget 8 så är värdet 80% av inköpspriset osv.

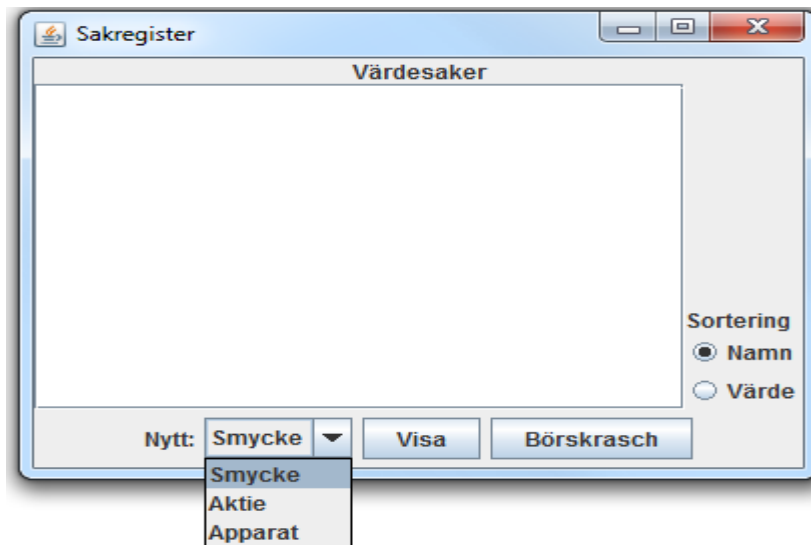
Oberoende av hur värdet beräknas ska man alltid lägga på 25%-ig moms på värdet. Implementera detta gärna så att man inte behöver tänka på detta när man implementerar beräkningen av värde för de specifika typerna av saker.

Alla värdesaker ska kunna presenteras med en sträng med sakens namn, värde och värden på samtliga typspecifika egenskaper.

Testa gärna dina klasser i små program med hådkodat skapande av ett antal objekt av de olika klasser och utskrifter av dem på `System.out`.

Del 2

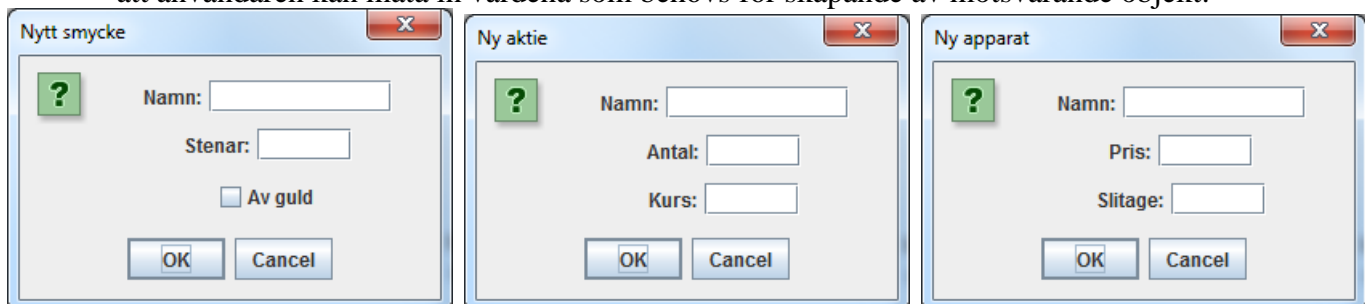
Skriv ett program som låter användaren registrera och lista värdesaker i ett grafiskt användargränssnitt. Fönstret ska se ut så här (man kan ändra detaljer i utseendet):



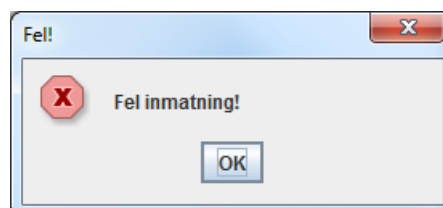
I mitten är det en textarea där programmet skriver ut listor av värdesakerna. Längst ner finns dels en s.k. combobox med en liten drop-down-meny, där användaren kan ange om hen vill registrera ett smycke, en aktie eller en apparat, dels en knapp "Visa" där användaren kan begära en lista över alla registrerade värdesakerna, och dels en knapp "Börskrasch" som sätter kursen för alla aktier till 0 (detta gäller de aktier som är registrerade just nu, aktier som registreras senare kan ha en högre kurs).

Till höger finns radioknappar som blir aktuella i del 3.

När användaren väljer ett alternativ i comboboxen så ska ett litet inmatningsformulär visas, så att användaren kan mata in värdena som behövs för skapande av motsvarande objekt:¹

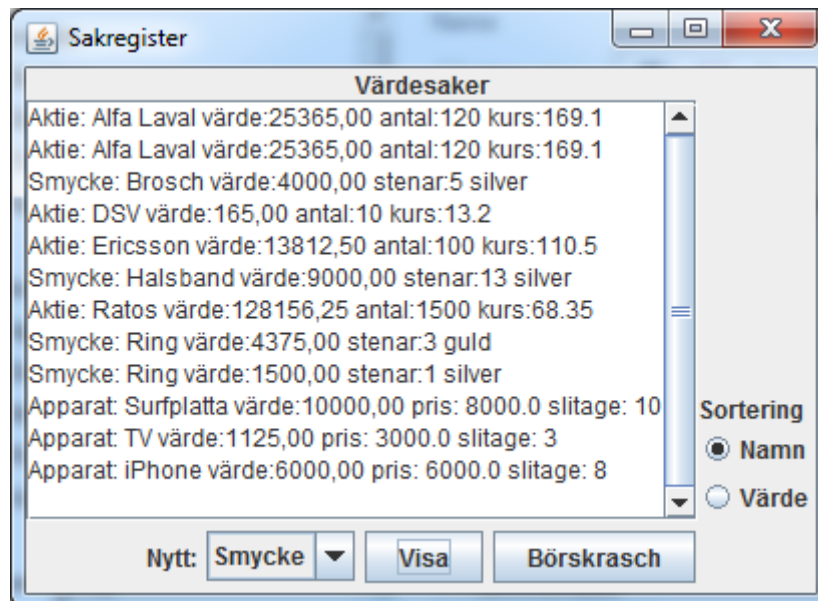


Värdena ska kontrolleras, om användaren lämnar något fält ofyllt eller fyller ett fält som ska vara numeriskt med icke-numeriska tecken så ska ett felmeddelande ges:



¹ Den här delen av programmet kommer nog att kräva en hel del kod för skapande av de tre formulären. Koden blir ganska lik i de tre fallen, men kan vara svår att generalisera så att man slipper upprepa kod.

När användaren tryckt på knappen "Visa" så skrivs listan över alla värdesakerna ut i textarean:



Del 3

Komplettera programmet med att listan vid utskrift ska vara sorterad på antingen namn eller värde. Användaren väljer sorteringsordning med radioknapparna i panelen till höger. Du ska inte implementera någon egen sorteringsalgoritm utan se till att kunna använda sorteringsmetoder som finns i Javas standardbibliotek². Visning av den omsorterade listan behöver inte ske direkt efter valet av sorteringsordning, man kan kräva att användaren klickar på knappen "Visa" för att se den omsorterade listan.

Inlämningsinstruktion

Inlämningen sker på kurssidan i iLearn2 senast 2016-04-08 för rättning under kursen. Uppgiften kan även lämnas in samtidigt med inlupp 2 2016-04-29, eller allra senast vid uppsamlingsinlämningen 2016-08-19.

Om man har jobbat i grupp så är det endast en i gruppen som lämnar in uppgiften, men alla gruppmedlemarnas namn och personnummer ska anges i inlämningskommentaren ("Submission comments")³.

För den här uppgiften ställs det inga krav på inlämningens format (ensklida filer, ZIP-arkiv, JAR-arkiv osv.), men endast källkodsfiler (.java-filer) ska lämnas in, inga .class-filer, projektfiler från IDE e.d.

Däremot ställs det krav på kodutseende: koden ska vara korrekt indenterad, namn på klasser, variabler och metoder ska följa Javas namngivningskonventioner, variabler i de klasser som är tänkta som biblioteksklasser ska vara `private`-deklarerade och metoderna i klasserna ska ha korrekta skyddsnivåer (`public`, `protected` eller `private`).

² Dessa metoder går igenom på föreläsning 9 (enligt preliminär plan)

³ Fältet för "Submission comments" finns på sidan "Submission status" som man får fram efter att ha klickat på inlämningslänken, innan man klickar på "Add submission"