

Objektorienterad Programmering

Övningsuppgifter

Våren 2009

1. Skriv ett program som utför beräkning av fakultet för heltal. Nedan följer ett exempel på hur programmet skulle kunna fungera.

```
~pem> java factorial 5
120
~pem> java factorial 1
1
~pem>
```

Tips: Metoden `Integer.parseInt (String)` tar en sträng, och om strängen representerar ett heltal returnerar den det.

2. Skriv en klass som beräknar ytan av en cirkel med en given radie.
3. Skriv ett program som avgör om ett ord är ett palindrom¹. Stora och små bokstäver får (men behöver inte) betraktas som olika bokstäver.
4. Skriv ett program som avgör om ett tal är ett primtal (delbart endast med ett och sig själv).
5. Anställda på Lo-Fat Burger Queen har en baslön på 43.20/h. Om de jobbar övertid tjänar de 1.5 ggr baslönen. De tider de betjänar kunder har de dessutom en provision på försäljningen enligt nedanstående tabell. Skriv klasser för att beräkna en dagslön givet ett visst antal arbetade timmar och ett belopp som avser försäljningen.

Försäljningssumma	Provision
1000 - 5000 Kr	5% av den totala försäljningen
5000 - 10000 Kr	10% av den totala försäljningen
≥10000 Kr	15% av den totala försäljningen

6. Skriv en metod `distance` för att beräkna avståndet mellan två punkter i rummet. Beskriv även hur en punkt i rummet bör representeras. D.v.s. vad skall `distance` ta för argument.
7. Skriv en klass för att representera heltal. Klassen skall innehålla metoder för att addera, subtrahera, multiplicera och dividera talen.
8. Lägg till en metod i klassen som skrevs i uppgift 7 för att utföra fakultetsberäkning.

¹Ett palindrom är en teckenföljd där ordningen på tecknen är samma från början som från slutet av följden, t.ex. ordet "alla".

9. Skriv en klass `Complex` som kan användas för att representera komplexa tal och operationer på dessa.
10. Diskutera hur testning bör utföras i uppgift 9.
11. Ett flygföretag behöver simulera verksamheten vid en flygplats och behöver därför en modell över verksamheten vid flygplatsen. I verksamheten behöver man ta hänsyn till att det ankommer och avgår flygplan av olika typer, att dessa skall ha passagerare, besättning, lastas med bagage, fyllas med bränsle och ha en destination. Rita en modell som kan beskriva en sådan verksamhet.
12. Vid en fabrik finns ett antal maskiner som används för att sätta samman detaljer till färdiga produkter. Detaljerna kommer från ett antal olika leverantörer. En produkt består av ett stort antal detaljer och kräver därför behandling av flera maskiner. Varje maskin har en viss kapacitet. Rita en modell som kan beskriva verksamheten.
13. Komplettera klassen `Complex` från uppgift 9 med en metod för konjugering (negering av den imaginära delen av talet). Föreslå lämpliga testdata.
14. Utvidga klasserna för heltal och komplexa tal från tidigare uppgifter, så att det är möjligt att utföra addition mellan de olika typerna. Föreslå testdata, samt rita ett klassdiagram.
15. Skriv en grammatik för enkla aritmetiska uttryck (+, -, *, /).
16. Rita syntaxdiagram för följande grammatik:

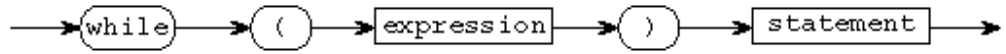
```

declaration ::= type declarator
type        ::= int | char
declarator  ::= name | declarator '[' number ']'
              | declarator '(' type ')'
              | '(' declarator ')'
```

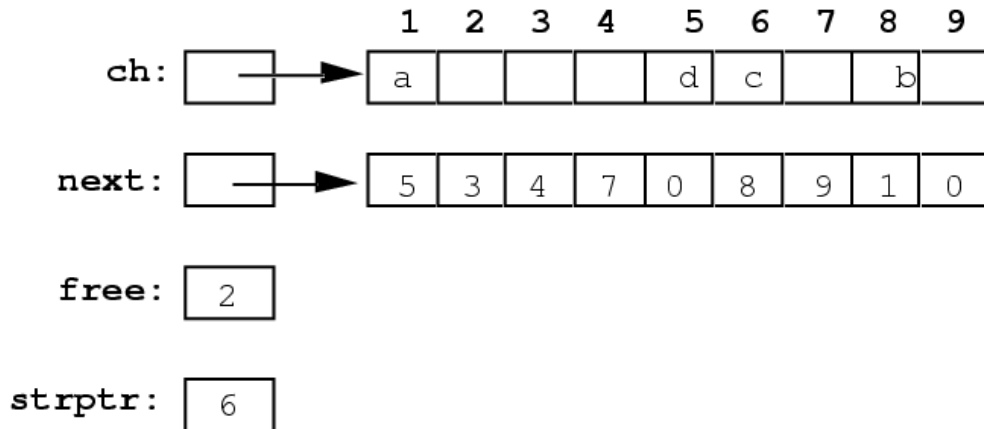
17. Rita ett syntaxträd för följande uttryck:
 - a. $2 + (3 - a)$
 - b. $5 * (2 + 3) - 4/5$
 - c. $a := a + 1$
18. a. Rita ett syntaxdiagram för följande grammatik:

`if ::= 'if' '(' expression ')' statement ['else' statement]`

b. Skriv en gramatik i BNF eller EBNF för följande syntaxdiagram:



19. En länkad lista av tecken kan lagras i en teckenvektor `ch` med hjälp av en heltalsvektor `next` där information om var nästa tecken eller fria utrymme finns. Vi har en strängpekare `strptr` som innehåller adressen till det första tecknet och en pekare som pekar på den första lediga platsen `free`. Ex: "cbad" lagras i en tabell med längden 9.



Följande uppgifter skall lösas med hjälp av ovanstående.

- a. Skriv en procedur `writeList ()` som skriver ut tecknen i den länkade listan.
 - b. Skriv en procedur `insert (c, i)` som länkar in tecknet `c` efter det tecken som har index `i`.
 - c. Skriv en procedur `delete (i)` som tar bort det tecken som har index `i` och stoppar tillbaks platsen i den fria listan.
20. Ändra listan så att det blir en enkellänkad struktur I stället för en array.
21. Skriv en metod för att summera alla tal i en lista. Inget krav ställs på vilken typ av lista som används (Tips: använd tex Javas klass `LinkedList` och metoderna `get` och `size`).

22. Skissa en klasshierarki och gänssnitt som tillåter representation av lisp-uttryck. Obs! operander kan vara lisp-uttryck i sig.
23. Implementera klasser för att representera lisp-uttryck bestående av konstanter, aritmetiska operatörer $+$, $-$, $*$, $/$ och lisp-uttryck.
24. Skissa klasser och gänssnitt för en dubbellänkad sorterad lista.
25. Implementera de klasser som behövs för att realisera en dubbellänkad sorterad lista.
26. I samband med textredigering önskar man kunna läsa in och lagra text på så vis att man kan behandla olika nivåer, alltifrån enstaka tecken till hela meningar och stycken. I vissa fall är det önskvärt att kunna komma åt de enstaka bokstäverna, t.ex. vid rättning av stavfel, medan man i andra sammanhang enbart är intresserad av hur många tecken som ingår i en mening, t.ex. vid själva sättningen osv. Skriv de klasser som behövs för att realisera datastrukturen.
27. Skriv de metoder som behövs för att skriva ut en mening i uppgift 26.
28. Skriv en adapter för att använda en stack som kö.
29. Skriv en rekursiv metod för att gå genom en labyrint. Labyrinten representeras av en matris med 0 eller 1 i varje position. I varje steg kan man gå ett steg åt öster, väster, syd eller nord. Inga diagonala förflyttningar är tillåtna. Det är tillåtet att gå från en ruta med värdet 1 till en annan ruta med värdet 1. Alla andra vägar är otillåtna.
30. Skissa en klasshierarki och gänssnitt som tillåter representation av aritmetiska uttryck i form av trädstruktur. Inre noder skall vara operatörer och löven representerar operander.
31. Implementera klasser för att representera trädstrukturen i uppgift 30. Det skall finnas metoder för evaluering och utskrift av uttrycken.
32. Diskutera olika tekniker för att implementera en list-ADT.
33. Diskutera olika möjligheter att implementera komplexa tal, fördelar och nackdelar med direkt implementation, arv samt delegering. Ge kodexempel som stöd i diskussionen.
34. Utöka grammatiken i uppgift 30 och 31, så att unära operationer kan utföras t ex negation ($-$) och sinus.

35. Skriv en generisk kö med felhantering.
36. Skriv en subklass till kön i uppgift 35 så att den fungerar som en prioritetskö. Diskutera för och nackdelar med arv och delegering i detta fall.
37. Bygg en finit automat som kontrollerar att en sträng bestående av morse-tecken är korrekt . Strängen skall accepteras om den bara innehåller ".", "_" och " ". Det får finnas två blanka för att markera att ett ord är komplett och en blank för att markera att ett tecken är komplett. Det längsta tillåtna morsetecknet består av sex tecken.
38. I språket CAB finns bara tre olika bokstäver 'A', 'B' och 'C'. Det finns en enda regel som avgör om en följd av de tre bokstäverna är ett ord i språket. Direkt framför varje förekomst av bokstaven 'A' måste bokstaven 'C' finnas.

Exempel på ord som ingår i språket är:

CAB
C
CBBB
BCA
CACCA

Exempel på ord som inte ingår i språket är:

CCAA
ACA
CBABB
A

Skriv en gramatik för språket i BNF eller EBNF.

39. *a.* Konstruera en finit automat för språket som beskrivs i uppgift 38.
b. Skriv ett program som avgör om ett ord tillhör språket som beskrivs i uppgift 38.
40. Givet ett binärt träd, skriv metoder för att
a. räkna antalet noder i trädet samt
b. summera alla värden som lagrats i trädet.

41. Skriv en subklass till det binära trädet så att man kan
 - a. beräkna den interna väglängden i trädet
 - b. kontrollera om trädet är komplett
 - c. räkna upp trädets noder i "nivå"-ordning från vänster till höger
 - d. kontrollera om två träd är strukturellt lika
 - e. balansera trädet

42. Stavningskontroll kan utföras på följande sätt: Med hjälp av ett lexikon bygger man ett träd. Man har en startnod. Utgående från det första tecknets värde väljer man en av "barn"-noderna (numrerade a-z). Därifrån går man vidare med hjälp av andra tecknet, o.s.v. Om sökvägen tar slut innan ordet gör det så finns ordet inte i lexikonet och kontrollen visar att ordet är felstavat. Om ordet tar slut och man befinner sig på en nod som är en möjlig slutnod (inte nödvändigtvis ett löv) så är ordet rätt stavat. "alba" är rätt stavat, men sökvägen innehåller även "albatross". I alla andra fall är ordet felstavat.
 - a. Gör en modell över ett sådant lexikon.
 - b. Visa representation av de olika nödvändiga noderna (d.v.s. skriv klasserna, men inte alla metoder).
 - c. Skriv de metoder som krävs för att kontrollera om ett ord är rätt stavat.

43. Skriv ett program som kopierar en fil som innehåller ett Java-program men hoppar över alla kommentarer.

Tips: Det finns två olika typer av kommentarer i Java. Nästlade kommentarer tillåts ej.

44. Skriv ett program som går igenom en Java-programfil och extraherar gränssnittet hos den yttersta klassen.

45. Skriv ett program som läser en fil och ersätter alla priser på formen "xx kr" eller "xx.xx kr" med motsvarande moderna form "SEK xx" respektive "SEK xx.xx".

46. MPEG Layer 3 är ett format för att lagra ljud, detta format sparas ofta i filer med ändelsen "mp3". mp3-filer används i många sammanhang för att lagra musik. I många mp3-filer lagras även tillhörande information

om artist, låtnamn o.s.v. Det finns en standard som heter ID3² som går ut på att de sista 128 byten i en mp3-fil innehåller denna typ av information. Nedan följer en tabell som beskriver vad de 128 byten innehåller.

fält	längd	offset	innehåll
tag	3	0	Detta fält innehåller alltid "TAG".
titel	30	3	Låtens titel.
artist	30	33	Artistens namn.
album	30	63	Namn på skivan från vilken låten kom.
år	4	93	Produktions år.
kommentar	30	97	En kommentar utan bestämt innehåll.
genre	1	127	Musikgenre.

Skriv ett program som listar denna typ av information för en mp3-fil. För att slippa läsa hela mp3-filen är det lämpligt att använda sig av en Randomaccessfile. Nedan följer ett exempel på hur programmet skulle kunna fungera.

```
~pem> java ID3 track02.mp3
Title: Soak Up The Sun
Artist: Sheryl Crow
Album: C'mon, C'mon
Year: 2002
~pem>
```

I katalogen /info/suooop08, oop09/Exempel/mp3/ finns några mp3-filer som kan användas för att testa att programmet fungerar. För den intresserade finns även en lista över genrer.

47. Skriv ett program som uppdaterar en RandomAccess-fil där varje post har formatet `Varunamn<10 pos>varunummer<5 pos>pris<på formen xxx.xx kr>sort<st eller kg>`, och alla priser < 1000 kr, så att alla priser justeras nedåt med 10 procentenheter.
48. På en RandomAccess-fil finns en "telefonbok" vars poster har formatet `Namn<25 pos>Telefonnummer<10 pos>` och en sekvensiell fil bestående av poster med information på formatet `<Postnummer><Fältnummer><Värde>`. Denna information används för att utföra uppdateringar t.ex. 17 2

²Det finns även en nyare standard som heter ID3v2 med vilken man kan lagra mycket mer tillhörande information än med ID3, dock är dess specifikation betydligt mer komplicerad.

7906276 betyder att 17:e postens andra fält skall ändras till 7906276. Skriv ett program som utför uppdateringen.

49. Givet ett binärt träd skriv metoder, ev i en subklass till det binära trädet, som
 - a. beräknar ett trädets höjd
 - b. räknar löven
 - c. kontrollerar om två träd är innehållsmässigt lika
 - d. kopiera ett träd
50. Som statistiskt mått på en mängd tal används ibland typvärdet, d.v.s. det värde som är mest frekvent (oftast förekommande). Skriv en procedur som beräknar typvärdet för data i en array med heltal. Om flera värden är lika vanliga skall proceduren ge det minsta av dem.
51. För att kunna hitta snabbt i en informationsstruktur använder man oftast hashing, men då kan man inte enkelt räkna upp posterna i nyckelordning. Man kan använda "container"- element som ordnas i ett binärt träd (för uppräknig i ordning) och i en hashtabell (för snabb åtkomst). en klass som realiserar ett register med möjlighet till insättning, borttagning, sökning och uppräknig i ordning enligt denna idé.
52. Antag att vi inte är intresserade av att sortera, utan endast av att finna den k:te nyckeln i storleksordning. Visa hur man kan skriva om quicksort för att lösa problemet.
53. Antag att alla nycklar i en lista är tal. En annorlunda metod för att finna en pivot är att ta medelvärdet av alla nycklar som pivot. Metoden kallas meansort.
 - a. Implementera meansort.
 - b. Kan meansort spåra ur och bli riktigt dålig?
54. Skriv en metod för att en lista med positiva och negativa tal ordnas så att de negativa talen kommer först och sedan de positiva. Negativa respektive positiva tal behöver inte vara sorterade, du behöver endast samla alla negativa tal för sig. Använd inte en extra array, problemet kan lösas ändå.

55. Antag att det finns objekt som lagrar 0 - 1000 element (av någon sort). Dessa objekt innehåller metoder för att sortera de lagrade objekten och för att ta en lista av objekt av samma sort och generera en lista av objekt som representerar en enda sorterad följd. Skriv en klass som realiserar denna idé.
56. Implementera en sorteringsmetod, för 10-siffriga heltal, som använder facksortering .
57. Generalisera föregående så att godtyckliga element, med diskreta nycklar, kan sorteras.
58. Skriv en effektiv variant av quicksort.
59. I binärsökning delas vektorn i två delar och sökningen fortsätter i den del där elementet kan finnas. Skriv en sökmetod där vektorn delas i tre delar.
60. Skriv en metod för insättning i ett B-träd.
61. Skriv en metod för insättning i ett B+-träd
62. Skriv en metod som, utan att sortera, undersöker om en lista är sorterad (oberoende av sorteringsordning).
63. Skriv en heapsort som delar vektorn i tre delar i stället för två.
64. Gör ett program dir för att lista innehållet i en katalog. Programmet skall ta ett godtyckligt antal argument varav alla utom eventuellt det sista är väljare. Programmet skall köras på följande sätt:

```
java dir {-a | -l | -F | -s | -t , -r} [biblioteksnamn]
```

om biblioteksnamnet inte anges så skall det vara ”.”. Väljarna har följande betydelse:

- a alla filer, även de med . i början
- l lång form d v s accessrättigheter och tidsstämplar skall anges
- F skriv / efter filnamnet om det är en katalog
- s sortera efter filnamn
- t sortera efter tidsstämpel
- r lista rekursivt, varje fil som i sig är ett bibliotek listas för sig.

65. Skriv en listhanterare som kan lagra alla objekt som är subclass till klassen `Element` (som också skall redovisas). Listan skall arbeta med `ContainerElement` så att ett `element` kan stå i flera listor samtidigt. Listan skall vara ordnad.
66. Skriv en metod `copy` i listan från uppgift 65, så att listan kan kopiera sig själv (med innehåll)
67. Skriv en iterator, till listan från uppgift 65, som är så generell att man kan göra nästan vad som helst med hjälp av den.
68. Skriv en metod för borttagning av ett nyckelvärde i ett B-träd.
69. Skriv om metoderna för insättning i ett B-träd så att de fungerar i ett B+-träd
70. Skriv en metod för att balansera ett binärt träd enligt följande, mindre snygga, metod: lägg alla noder i en vektor och organisera om trädet med hjälp av denna. Rot till trädet bör vara den nod som finns i ett index som indikerar vektorns halva längd.