Förslag till övningar på Map:ar

Dessa uppgifter är inte obligatoriska och du gör dem i mån av tid. De är gjorda så att du får öva på att hantera vissa datastrukturer.

Uppgifterna går ut på att komplettera/modifiera bifogat program WordCounter. Programmet läser in ord från en angiven fil – ordinläsningen är gjord i metoden processFile och behöver inte förändras. För varje ord anropar processFile en metod som heter processWord – det är denna metod som du ska komplettera/modifiera. Dessutom kan du behöva göra andra tillägg och modifieringar – lägga in utskrifter o.s.v.

Tillsammans med denna text ligger det en liten textfil `igelkott.txt` med texten till en dikt av Nils Ferlin samt en stor textfil `bibeln.txt` innehåller texten för hela Bibeln – dessa kan du testa ditt program med (inte för att det skulle vara svårt att hitta textfiler – varje Java-källkodsfil är ju en textfil).

Räkna ord

Klassen WordCounter innehåller kod för att läsa in ord från en textfil. I nuläget läser programmet enbart in ord från en valfri textfil och skriver ut radnummer följt av ordet. Du startar programmet genom att i terminalen skriva:

java WordCounter filnamn

där filnamn ersätts med sökvägen till textfilen. Uppgiften går ut på att komplettera programmet enligt nedanstående utökningar.

1 – Räkna ord

När programmet startar går det igenom alla rader i textfilen, och för varje rad går det igenom alla ord på raden (med ord menas sekvenser av bokstäver). I den första delen av uppgiften är det tänkt att du ska utöka WordCounter så att den klarar av att räkna förekomster av ord. Det finns en metod som heter processWord som tar emot ett ord och ett radnummer, just nu skriver den ut ordet och radnummret men du ska ändra den enligt följande.

Välj en lämplig datastruktur för att hålla reda på hur många gånger varje unikt ord förekommer. Tiden detta tar kommer att vara beroende på hur stor filen är och en viss fördröjning är förväntad.

Efter att ha läst in och räknat alla ord ska programmet sedan föra en terminalbaserad dialog med användaren. Här ska man kunna söka efter ord och få veta hur många gånger ordet förekom. Observera att det inte får uppstå någon **märkbar** fördröjning efter varje sökning. Detta ställer särskilda krav på vilken datastruktur som används och utesluter exempelvis List-baserade samlingar för att spara orden i.

Programmets utskrift och dialog kan exempelvis se ut så här (användarens input i **fet** stil):

```
Läser in ord...

Ange ord: meningen
Ordet 'meningen' förekom 42 gånger

Ange ord: fantasibrist
Ordet 'fantasibrist' förekom 1 gånger
```

2 Index

I den andra utökning ska du bygga ett index från orden i textfilen. Du behöver inte ha någon terminaldialog med användaren. Skriv istället ut orden i **bokstavsordning** följt av vilka **sidor** orden förekommer på. Utskrift ska ske till *System.out*.

För enkelhetens skull antar vi att varje sida rymmer 60 rader text. Detta måste du tänka på när du läser in orden. Utskriften skulle kunna se ut så här:

abdikera 1, 4, 10
cykel 2
monstruös 7, 25, 26
... ..., ...
överkörd 12

Observera att om ordet förekommer flera gånger på en sida ska naturligtvis sidnumret bara nämnas **en** gång i indexet.

Försök att använda datasamlingens inbyggda funktionalitet. Det är t.ex. onödigt att manuellt sortera orden i bokstavsordning, eller att manuellt kontrollera att dubletter inte förekommer.

Normalt sett hamnar allt som skrivs till System.out i kommandofönstret, men om man kör programmet från kommandofönstret så kan man omdirigera System.out till en fil på följande sätt:

java WordCounter filnamn > out_file