Rapport:

Objektivering.

V.Fundin //.

.

Inledning:

Programmet (java) ska hantera data från olika kalkylark/cvm, Json.

Hur jag gjorde.

Plan:

Började med att arbeta med hur man förvandlar varje typ av fil i uppgiften till objekt.
Det gjorde jag genom att först ta del av informationen på Classroom och internet för att få en förståelse för vad som krävdes av programmet. Sedan började jag med att skapa ett projekt, lägga till det i GIT(lokalt), så jag kunde hålla reda på de olika ändringarna(commits) i programmet.
Första målet var att kunna läsa en fil och prompta/printa i terminalen, för att sedan gå vidare till att spara i ett objekt/klass, och sist sortera datan. Från början jobbade jag i 'main'-klassen för att få en förståelse över hur man kan arbeta med de olika filerna.

Printa:

För att printa informationen i filerna använder jag mig av information på internet, där kom jag fram till att alla filtyper i det här fallet fungerar på ungefär samma sätt när de ska printas, där man använder sig av någon typ reader/scanner som till exempel 'BufferedReader' för att läsa varje rad i en fil,

som sedan ‘loopar’ igenom all text i filen tills den gått igenom alla rader i filen. I loopen kan man välja vad man ska göra med informationen, som att visa den i terminalen eller att spara den på något sätt.

BufferedReader är ‘synchronous’ och använder sig av en ‘fileReader’ för att läsa filen, det gör att den kan kommer att prova att läsa filen men kan misslyckas, därför lägger jag den i en ‘Try-catch-block’ som hanterar errors om den inte kan läsa filen korrekt.

Provade att använda mig av olika bibliotek för att läsa av de olika filerna, för JSON fungerade JSON.simple för att läsa av filen, men provade att använda GSON också. ‘JSON.parse’ fungerade för att läsa JSON filen istället för en ‘BufferedReader’ som i CSV-klassen.
För XML använde jag mig av ‘XML.parser.documentBuilderFactory’.

Spara:

I Loopen som går igenom all text i filen så väljer jag att spara den i ett objekt, en dynamisk array (ArrayList) med en typ av String array. Så att varje element i List är en till array, som i sin tur innehåller informationen av filen.
ex. ArrayList: { array: [String : “a: a”, “ ‘b’: b”], [“a: a”, “ ‘b’: b”] }
Sparar informationen på det sättet så att jag senare kan jämföra varje del av datan för att kunna sortera den.

Klasser:

Använder mig av klasser för varje typ av fil, så en fil för CSV, en för XML, och en för JSON. I varje klass finns en konstruktor som tar ett argument, som är namnet på filen, eller filsökvägen i formen av en string. På så sätt kan jag skapa en ny instans av varje klass för varje fil jag vill hantera, datan läses av i konstruktorn och sparas i ett objekt som sedan kan hämtas med en getter.

Sortera:

Eftersom varje rad i filen delas upp och sparas på samma sätt, i en array, så kan jag jämföra varje index/kolumn med varandra.
Då kan jag välja att ändra ordningen med hjälp av att sortera(sort) objektet med en ny 'comparator'. Den jämför varje String vid en viss index i arrayen med 'compareTo'.
Funktionen som sorterar tar även bort den första raden som innehåller rubriker i varje kolumn och sparar den i en ny array. På så sätt när man väljer att sortera datan så ändrar den inte ordning utan är fortfarande längst uppe.

-Hur det sparas, (i=index):
Objekt: { array(0): ["i(0): b", " i(1): b"], array(1): ["i(0): a", " i(1):a"] }

-Väljer t.ex att sortera index(1) i array:
Objekt: { array(1): ["i(0): a", " i(1): a"], array(0): ["i(0): b", " i(1):b"] }

Eftersom:

- Arrayen sorteras med 'compareTo' som jämför värdet av index som vi har valt, i det här fallet den andra. Då jämförs index(1) i den första array mot index(1) i den andra array. I det här fallet är det då värdet 'b' mot 'a', då byts ordningen alfabetiskt med hjälp av 'sort'.

Gui:

En enkel GUI med swing(kod) för att presentera två knappar, en för att välja fil som ska läsas av, och en för att sortera en fil.
se förbättringar nedan.

Förbättringar:

Om jag hade mer tid för att förbättra koden skulle jag:

- Mer funktionalitet i GUI.Swing, t.ex att programmet kan läsa av om en fil är vald och om den kan läsa/sortera. För att sedan bara visa knapparna(JButton) om allt fungerar som det ska.
- Möjlighet för att visa XML i GUI.
- Bättre funktionalitet för sortering. Funktionen för att sortera har möjlighet att välja vilken kolumn som ska sorteras alfabetiskt, men funktionen är inte inlagd i GUI. För att göra det hade jag lagt till en 'dropdown-meny' eller liknande där användaren kan välja vilken kolumn baserad på index, eller bättre, att programmet läser av vilka kolumner som finns och sedan promptar alternativen.
- Fixa indentering för att lättare läsa av filerna.

V.Fundin./