Zackarias Ouacha, Michael Lake || zacou776, micla386

TDP007 - Seminarie 2

Hur har ni gått tillväga för att lära er Ruby?
Vi har i stor del använt slidesen från föreläsningarna i kursen för att lösa uppgifterna. Under detta seminariumet så förlitade vi oss dock mer på Rubys dokumentation och generella forumsökningar (stackoverflow, geeksforgeeks) för att lösa uppgifter, till exempel för att hitta hur man delade upp en array i att den ska innehålla arrayer av par som innehåller till exempel lag och målskillnad eller dag och skillnaden mellan max- och mintemperatur.

Vad finns det för nya konstruktioner i Ruby som ni inte sett förut?
För att konvertera ett regex-matchobjekt till en integer måste man först göra om matchobjektet till en string med match.to_s för att sedan kunna göra det till en integer, alltså match.to_s.to_i.

Vad finns det för konstruktioner som ni känner igen men som ser lite annorlunda ut?
När man skriver ut en sträng som ska innehålla information från ett objekt kan man inte bara skriva print “Test” Objekt “Test2” utan allting måste finnas inuti ett set med citattecken, och objekten skrivs ut med specialtecken såhär : “Test #{Objekt} Test2”.

Finns det något som ni irriterar er på eller tycker om i Ruby?
Det är kul att man kan “tänka i andra språk” om man har hållit på med de innan och att det fortfarande funkar i rätt stor grad om man skriver på det sättet i Ruby. Det gör att lösningar kan se väldigt olika ut, vilket gör att man kan hamna i andra tankegångar och bredda synen på hur man kan lösa en uppgift.

Uppgift 1 del A:
Hur har ni tolkat uppgiften?
Vi ska läsa in en .txt fil med en tabell som innehåller resultat från en fotbollsliga. Vi ska med hjälp av tabellen ta fram namnen på lagen och motsvarande målresultat, alltså gjorda samt insläppta mål, och räkna ut vilket lag som har minst målskillnad. Vi ska också göra en uppställning av lagen där laget med minst målskillnad står högst upp och laget med störst målskillnad är längst ner.

Vad var svårt eller lätt med uppgiften?
Denna uppgift bidrog med en hel del utmaningar. Vi behövde först läsa in allt innehåll från filen och sedan “rensa” den med regex. Regex mönstret var något vi kämpade lite med då vi inte riktigt fick fram matchen vi ville, men till slut kom vi på att grupper går att använda sig av.

Vi hade sedan problem med att bestämma vilken metod vi ville använda med regexen. Vi började med att använda *.match* men den returnerade inte riktigt det vi

ville. Det slutade med att vi istället använde oss av *.scan* som endast returnerar det som fångas i grupper vilket passade kanon!

Det vi senare hade problem med var väl egentligen faktumet att vi jobbade med arrayer. Vi kom inte på förens väldigt sent i lösningen att vi borde lägga ihop Namnet och dess tillhörande tal i en array som läggs in i en “teams” array för att enklare hantera målskillnaden som hör till specifika lag. När vi väl löst detta så var det inga större problem som uppstod, bara att det tog ett tag innan vi insåg att det var så vi skulle göra.

Vi fick tidigt problem i uppgiften att vi fick “Local variable undefined” för att vi först försökte använda oss av globala variabler för att lösa uppgiften men märkte snabbt att det innebar några problem. Vi visste inte att vi var tvungna att deklarera en global variabel med $ till att börja med, och när vi väl gjort det insåg vi snabbt att det bryter inkapslingen i koden. Man kommer snabbt bort från objektorienterat tänk att en sak ska göra någonting bra och specifikt och det blir svårt att felsöka en global variabel om den används på flera ställen och kolla när felet har uppstått.

Uppgift 1 del B:

Hur har ni tolkat uppgiften?

Vi ska läsa in en .txt fil med en tabell som innehåller dagar i en månad och dess temperaturdata. Vi ska med hjälp av tabellen ta fram numret på dagarna och motsvarande högsta och lägsta temperatur. Vi ska också göra en uppställning av dagarna där dagen med minst skillnad mellan högsta och lägsta temperatur står högst upp och dagen med störst skillnad är längst ner.

Vad var svårt eller lätt med uppgiften?

Denna uppgift innebar inga större utmaningar än förra uppgiften, då den liknar den på många sätt. Dock var regex mönstret på denna något svårare då vi på två ställen behöva ta en “*” i åtanke vilket bidrog till lite problem. Detta var på dagarna 9 och 26.

Skillnader/Likheter mellan denna uppgift och del A:

Skillnaderna mellan uppgifterna är minimal. Den största skillnaden är Regex mönstret i och med att tabellerna ser olika ut. Men i stor omfattning är dem nästan identiska. Filinläsning och uthämtning av data med Regex sker på samma sätt, och filtreringen av att hitta det laget eller den dagen med minst skillnad i antingen målskillnad eller temperatur sker på samma sätt.

Uppgift 2:

Hur har ni tolkat uppgiften?

Vi har tolkat uppgiften som att vi ska ta en kalenderhemsidas HTML-kod, parsa den på valfritt sätt och få ut de inplanerade aktiviteternas information och redovisa dessa på ett godtyckligt sätt. Aktiviteterna har en titel, datum, plats och information.

Zackarias Ouacha, Michael Lake || zacou776, micla386

Vad var svårt eller lätt med uppgiften?

Det svåra med uppgiften var att para ihop informationen, alltså att en aktivitet innehöll allt dess information separat från de andra aktiviteterna. Vi hade inga större problem att läsa in all information om aktiviteterna, men alla titlar kom först, sen alla datum osv. Det var också rätt svårt att komma in i tänket då det inte är något vi gjort sedan tidigare, till exempel själva hierarkin för taggarna, att det finns barn till taggar osv.

Vi började först att lösa uppgiften med Nokogiri. Filinläsning och att få ut den första delen av uppgiften, alltså "vevent"-information, var relativt enkelt, men vi tyckte att metoderna inte var så välbeskrivna och skiljde sig en del från den tillgängliga informationen vi hittade på forum osv. Vi valde därför att använda oss av Rexml istället då vi fann att det var mycket enklare att hitta information kring. Det blev genast mycket enklare att göra det som vi beskrev som en utmaning högre upp med hjälp av Rexml och tillhörande metoder, till exempel *file.elements.each()*.

Vi valde att använda oss av DOM-parser för att bearbeta hela filen i ett svep för att leta efter de relevanta taggarna. De relevanta taggarna hittade vi med hjälp av XPath parsning. W3schools hjälpte oss en hel del med hur xpath-mönster fungerade och hur vi skulle skriva upp våra mönster. De relevanta taggarna (och i följd alla dess sub-information) la vi i en array för att sedan bearbeta arrayen istället.

Vi tycker att vi kom fram till en väldigt smidig lösning som blev relativt kompakt. Programmet blev enkelt att bearbeta då varje funktion gör en sak väldigt bra, så skulle man behöva gå in och ändra i koden, till exempel i vår printfunktion som är baserad på att vi vill skriva ut 5 stycken element per veventtag, så är det enkelt att göra. Det blev väldigt enkelt att arbeta med all data i arrayer också då man kan använda sig av en blandning av "traditionella" kodningstekniker som vi använt oss av sedan tidigare, så som while och for-loopar, men också en hel del Ruby och Rexml specifika metoder.

En sak som vi insåg med vår lösning, som iprincip endast består av arrayer av olika slag, var att vi hade väldigt svårt att komma på olika testfall för den. Man kan kolla hur många element det finns jämfört med hur många det ska finnas och att saker är på rätt index i arrayerna, men mer testfall än så hade vi svårt att komma på. Till nästa gång kanske vi skulle ha tänkt igenom testfallen innan vi började en lösning, för att på ett bra sätt hitta en lösning som både är bra genomförd men också enkel att testa.