

## Seminarie 2

### Avsnitt 1:

Vår tolkning av uppgiften innebar att vi skulle läsa av värden från en fil för att sedan räkna ut skillnader mellan tal. Det skall finnas funktionalitet för att sortera alla inlästa objekt baserat på skillnaden av objektets värden. Objektet med minst skillnad kommer först i arrayen. Det skall även finnas funktionalitet för att hitta det objekt med minst skillnad mellan sina värden.

Vi tolkade att endast dagar i väderfiler skall räknas och därmed skall ej "mo" (längst ned) ingå.

Vi antar att användaren skriver in rätt information och har därmed inga kontroller.

Vi försökte återanvända så mycket kod som möjligt därför skapar vi en klass som kallas för "Component". Component används för att lätt kunna skapa instanser av fotbollslag eller dagar.

Uppgiften i sig var inte specifikt svår, vårt största problem var ett felmeddelande som uppstod vid kompilering av testfilen. Vi försökte skicka med en specifik array till funktionen `create_teams`, som skall hitta värden för att skapa objekt och placera dessa i en array. Det som förvirrade oss var att i huvudprogrammet uppstod inte några fel, medan när vi använde oss av funktionen i testfilen fick vi felmeddelande. Efter diskussion med handledare insåg vi att vi hade glömt namnge ett av testfallen korrekt vilket resulterade i att vi hade två funktioner med samma namn (`create_teams()` istället för `test_create_teams()` och `create_teams(array)`).

Vi funderade på att konstruera en gemensam funktion som skapar objekt och placerar dessa i en array istället för att ha två funktioner (nuvarande funktionsnamn är `create_teams` och `create_weather`). Vi bestämde oss för att skjuta på det i nuläget och göra det senare om det finns tid. Funktionerna använder sig just nu av två olika strategier för att spara data, vi är osäkra på vilken som är bäst. Spekulerar att det finns en gemensam lösning på problemet som innebär att man använder sig av specifika regex (som man får med hjälp av inparametrarna).

Vi har använt oss av regex för att identifiera värden som används när vi skapar objekt. Dessa värden plockas ut på olika sätt beroende på vilken funktion vi använder (`create_weather` och `create_teams`).

Det fanns ett tal på dag 26 som innehöll ett asterisk tecken, vi funderade på hur det påverkar en omvandling från sträng till heltal (integer). Talet var `97*` och vi använde oss av `"to_i"` på den. Det funkade utan problem.

### Avsnitt 2:

Vi tolkade uppgiften på följande sätt. Man skall med hjälp av en parser identifiera noder som man extraherar information från (t.ex. datum, namn på event, plats o.s.v.). Därefter skall

man spara denna information i ett ruby-objekt (alltså en klass). Klassen skall ha funktionalitet för att skriva ut informationen.

Vi använder oss av en DOM-parser vilket ger oss möjligheten att använda xPath. Vi valde DOM-parser för det kändes enklare att arbeta med om man tog hjälp av xPath.

Det största problemet vi bemötte var hur vi identifierade samt fick ut information från noder. Vi skrev kod som resulterade att vi alltid fick samma resultat över flera iterationer. Istället för att få resultat ett, två, tre och fyra fick vi resultat ett, fyra gånger. Lösningen på problemet var ett inlägg av Pontus på teams forumet. Man behöver ange en punkt i början av sin "matchning" (t.ex. `./span`) för att xPath skulle utgå från det nuvarande elementet, inte det första.

Förutom problemet nämnt i texten ovan var uppgiften lätt. Dessvärre var det väldigt svårt att hitta relevanta sökresultat på webben. När vi försökte identifiera vårt problem hjälpte inte google till mycket. Vi planerade att fråga assisten om hjälp, men Pontus inlägg räckte för att vi skulle klara oss.

I vår erfarenhet är rubys felmeddelanden ganska dålig på att förklara problemet. Ibland står det att koden inte har någon "end" (vi har alltså missat skriva end i någon sats, funktion eller klass). Vi behöver då gå genom hela koden eftersom ingen rad anges. De felmeddelanden som anger rader är ofta inte tydligt på vad problemet är, här tycker vi C++ är betydligt bättre.