

TENTAMEN 961G14, 961G15, 91MAXX, 92MAXX – Del 1

Pontus Haglund, Linköpings universitet

2020/06/03

Uppgift 1 - DSL och relationer - 13p

I denna uppgift kommer du arbeta med den givna filen `dsl.rb`.

Uppgift 1.1 - praktisk

Något som ofta används inom datorvetenskap är relationsdatabaser. Det är ett sätt att spara hur vissa data relaterar till andra data. I den här uppgiften ska du betrakta filen `dsl.rb` som programkod i ett litet internt DSL. Du ska skriva en metod som kan läsa in all information från filen, som om den vore programkod, och spara den i en lämplig datastruktur. Det domänspecifika språket ser ut på följande sätt:

```
dsl.rb
```

```
Descent is_of_genre miniatures_game
Bilehall is_of_genre miniatures_game
Carcassonne is_of_genre euro_game
Flux is_of_genre card_game
```

```
Bilehall is_expansion_to Descent
River is_expansion_to Carcassonne
```

```
Descent is_in_language english
Bilehall is_in_language english
Carcassonne is_in_language english
Carcassonne is_in_language german
Carcassonne is_in_language swedish
Flux is_in_language english
Flux is_in_language german
```

På första raden skapas relationen `is_of_genre` och spelet `Descent` har den relationen till genren `miniature_game`. Lösningen ska klara av vilka namn på relationer som helst, inte bara de som finns i detta exempel eller i denna ordning.

Därefter ska du skriva en metod `find` som kan ta reda på saker med hjälp av informationen som lästs in. Metoden `find` ska ta tre argument: namnet på relationen samt två saker som vi tror kanske har den här relationen. En av sakerna kan vara `nil`, och i så fall hittar vi alla möjliga

matchningar.

```
>> r=Relations.load("dsl.rb")
=> #Relations:0x4b4ace8 ...
>> r.find("is_in_language", nil, "german")
=> [["Carcassonne", "german", ["Flux", "german"]]
>> r.find("is_expansion_to", "Bilehall", nil)
=> [["Bilehall", "Descent"]]
```

Uppgift 1.2 - teoretisk

1. I uppgiften ovan var det domänspecifika språket redan specificerat i form av relationer. Föreställ dig en kund som absolut inte vill arbeta med att spara informationen i form av relationer men ändå vill kunna spara samma typ av data om varje spel samt kunna slå upp alla spel som exempelvis finns på tyska. Föreslå ett annat sätt som det domänspecifika språket kan utformas på.
2. Vilka ändringar skulle behöva göras i din lösning för att din lösning skulle fungera med ditt språkförslag?
3. Hur skulle det fungera att slå upp alla spel som hade språket tyska i ditt föreslagna fall?
4. Vad ser du för för- och nackdelar med det givna domänspecifika språket och det du föreslog ovan?

Uppgift 2 - XML - 14p

I denna uppgift kommer du arbeta med den givna filen `bgg.xml`.

Uppgift 2.1 - praktisk

På hemsidan boardgamegeek samlas folk som gillar brädspel. En funktion sidan har är att alla användare kan skriva ner vilka spel de har. I samlingen är det också möjligt att lämna omdömen om spelen. Boardgamegeek har ett api för att hämta ut information om olika saker som xml-filer. I den givna filen hittar ni en användares samling, de kommentarer hen givit sina spel samt hur hen har betygsatt dessa.

Din uppgift är att med hjälp av en xml-parser (förslagsvis REXML) hämta ut information om alla spelen användaren har i sin samling. Endast spel där användaren faktiskt har lämnat en kommentar är intressanta. Kommentarer finns i taggen `comment`. Du skall på ett valfritt, välstrukturerat format sedan skriva ut följande information om spelen:

- Spelets namn (tips: se taggen `name`)
- när användaren senast uppdaterad spelet i sin samling (tips: se taggen `status`)
- kommentaren användaren lämnat om spelet (tips: se taggen `comment`)
- användarens omdöme (tips: se taggen `rating`)
- det genomsnittliga omdömet (tips: se taggen `average`)

Du får lösa uppgiften med sax- eller dom-parsning.

Uppgift 2.2 - teoretisk

1. Du löste antingen uppgiften med sax- eller dom-parsning. Beskriv vad du hade behövt göra annorlunda för att lösa uppgiften med metoden som du inte använde.
2. Motivera vilken metod som passar bäst för att lösa just detta problem och varför.
3. I denna uppgift använde du dig av en xml-parser för att få ut informationen som efterfrågades, beskriv en annan metod du hade kunnat använda för att lösa problemet.

Uppgift 3 - Parsning - 13p

I denna uppgift kommer du utgå från diceroller parsern. Du hittar den givna koden i `rdparse.rb`.

Uppgift 3.1 - praktisk

Du ska skriva en enkel parser som kan läsa in och hantera träffar för en spelförening. Så här kan det se ut när föreningen skriver ner sina planerade träffar.

```
schema.txt
```

```
| 2020 |  
- January -  
1 Boardgame night  
15 Miniature painting  
  
- February -  
4 Card game bonanza  
14 Terrain building  
28 Strategy night
```

Programmet ska fungera så här:

```
>> c = Schedule.new  
>> c.read_schedule("schema.txt")  
=> [{"Boardgame night", 2020-01-01},  
    ["Miniature painting", 2020-01-15],  
    ["Card game bonanza", 2020-02-04],  
    ["Terrain building", 2020-02-14],  
    ["Strategy night", 2020-02-28]]
```

Metoden `read_schedule()` som anropas ovan använder sig av parsern för att tolka innehållet i filen och returnera en datastruktur som innehåller schemat.

Uppgift 3.2 - teoretisk

1. Beskriv vilka delar av problemet som du löst i:

- (a) Lexern
- (b) Parsern
- (c) Övrigt

2. Om strukturen i filen istället hade haft både året, månaden, dagen och mötesinformationen på samma rad, vilka ändringar hade du behövt göra i din lösning för att den skulle fungera.

3. En struktur som är så enkel som den i uppgiften ovan kanske lämpar sig bättre att lösa på ett annat sätt. Beskriv ett annat tillvägagångssätt för att lösa uppgiften ovan som inte använder sig av `rdparse`.