

TENTAMEN TDP007

Pontus Haglund, Linköpings universitet

2020/08/18

Uppgift 1 - XML - 14p

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

Uppgift 1.1 - praktisk

På hemsidan boardgamegeek samlas folk som gillar brädspel. En funktion sidan har är att alla användare kan skriva ner vilka spel de har och vill ha. 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 spel användaren äger och vill köpa, de kommentarer hen givit spelen samt hur hen har betygsatt dessa.

Din uppgift är att med hjälp av en xml-parser (förslagsvis REXML) hämta ut information om spelen i samlingen som användaren vill köpa. Informationen om användaren vill köpa spelet finns i taggen `status`. Du ska sedan på valfritt format skriva ut följande information om spelen användaren vill köpa:

- Spelets namn (tips: se taggen `name`)
- det genomsnittliga omdömet (tips: se taggen `average`)
- antalet spelare som kan vara med, max och minimum (tips: se taggen `stats`)

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

Uppgift 1.2 - 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 2 - DSL och relationer - 13p

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

Uppgift 2.1 - praktisk

I den här uppgiften ska du betrakta filen `schema.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. Detta språk specificerar träffar för en spelförening där datumet och aktiviteten anges. Det domänspecifika språket ser ut på följande sätt:

```
schema.rb
```

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

Programmet ska fungera så här:

```
>> s = Schedule.read_schedule("schema.txt")  
=> #Schedule:0x4b4ace8 ...  
>> s.display  
=> [{"Boardgame night", "Januari", 1},  
    ["Miniature painting", "Januari", 15],  
    ["Card game bonanza", "February", 4],  
    ["Terrain building", "February", 14],  
    ["Strategy night", "February", 28]]
```

Klassmetoden `read_schedule` returnerar alltså ett objekt av typen `Schedule`. Sedan anropas metoden `display` på det objektet vilket skriver ut den sparade informationen..

Uppgift 2.2 - teoretisk

1. I denna uppgift fanns strukturen på det domänspecifika språket given. Beskriv ett annat sätt som detta språk kunde gjorts på som också låter användaren ange året.
2. Vilka ändringar i din lösning skulle behöva göras för att fungera med det föreslagna språket?
3. Föreställ dig en förening som vill använda detta eller ett liknande format för sitt schema men absolut inte vill använda `rdparse` (eller andra parsers). Beskriv kortfattat ett alternativt

sätt att lösa problemet på.

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

Något som ofta används inom datorvetenskap är relationsdatabaser. Det är ett sätt att spara hur vissa data relaterar till andra data. Du ska modifiera filen `rdparse.rb` så att den klarar av att läsa in relationerna i den givna filen `relations.txt`. När filen lästs in med din parser ska man som användare kunna hitta olika data och hur de relaterar till varandra.

```
relations.txt
```

```
Descent is_of_genre miniatures_game
Bilehall is_of_genre miniatures_game
Carcassonne is_of_genre euro_game
```

```
Bilehall is_expansion_to Descent
```

```
Descent is_in_language english
Bilehall is_in_language english
Carcassonne is_in_language english
Carcassonne is_in_language german
```

På första raden skapas relationen `is_of_genre` och spelet `Descent` har den relationen till genren `miniature_game`.

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.

Programmet ska fungera så här:

```
>> r = Relations.new
>> r.read_file("relations.txt")
=> #Relations:0x4b4ace8 ...
>> r.find("is_in_language", nil, "german")
=> ["Carcassonne", "german"]
>> r.find("is_expansion_to", "Bilehall", nil)
=> ["Bilehall", "Descent"]
```

Uppgift 3.2 - teoretisk

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

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

2. Om strukturen i filen istället hade relationen som en rubrik och under denna rubrik listat alla spel som hade denna relation, hur hade du behövt ändra din lösning för att få det att fungera?