Dugga i TDP007
2008-03-12

Ola Leifler

12 mars 2008

Översikt

Duggan utförs under 105 minuter, från 11:15 - 13:00. När ni är klara ska ni lämna kvar filer enligt mönstret `<login>-<uppgift>.[rb|txt]` (se förklaring nedan) som innehåller era lösningar, både på frågor som kräver förklaring i text och sådana som kräver kod.

- Lösning på fråga 1 (ren text) av olale507: `olale507-1.txt`
- Lösning på fråga 2 (rubykod) av roban123: `roban123-2.rb`

Samtliga filer ni skapar ska ligga i katalogen ni startar i, ovanför `given_files`. Enda tillåtna kurslitteratur är *Programming Ruby – A Pragmatic Programmers Guide*.

Kom ihåg att motivera era lösningar så att vi förstår att ni förstått. Om ni behöver göra antaganden, förklara vilka antaganden ni gör.

Hjälp inför datortentan

För att öppna ett terminalfönster: Klicka på arbetsytan (bakgrunden) med höger musknapp och välj Terminal".

För att öppna en Emacs: Klicka på arbetsytan (bakgrunden) med höger musknapp och välj "Emacs".

Följande kommandon gäller i terminalfönstret:

Hur ser man vilka filer som finns i den mapp du står? `ls`

Hur ser man vilka filer som finns i given_files"? `ls given_files`

Hur kopierar man data från given_files"? Om filen man skall kopiera heter `A.TXT` skriver man:

```
cp given_files/A.TXT A.TXT
```

Den nya filen kommer då att heta `A.TXT` i mappen där du står. Kopiera filer från `given_files` till katalogen ovanför (din hemkatalog) innan du börjar modifiera dem.

1 Ruby

Vad gör metoden `Kernel#callcc`? Förklara och ge kodexempel.

2 Domänspecifika språk

Inför seminarie 2 fick ni i uppgift att skapa ett DSL för att antingen kommunicera med en extern, interaktiv process eller för att skapa en koppling mellan ett objekt och rader i en databastabell.

Ge två kodexempel som illustrerar hur man kommunicerar med en extern process från Ruby, där det ena exemplet ska utgöra ett DSL och det andra inte. Förklara också vad ni tycker definierar ett DSL. Ni behöver inte implementera någon tolkning av ert DSL utan behöver endast ange en syntax. Ni kan välja godtycklig extern process att kommunicera med.

3 Implementation av DSL

1. Vilka tekniker använder man sig av i Ruby för att implementera domänspecifika språk? Nämn åtminstone 2.

2. Skulle nedanstående syntax kunna fungera som grund för ett DSL i Ruby? Syntaxen är tänkt att återfinnas i konfigurationsfiler för en webbapplikation. Om ja, ange översiktligt hur man skulle kunna tolka den. Om nej, ange vad som är problematiskt.

```
server = http://localhost:8080
username = root
db = rails
password = rails
```

4 Parsning

I filen `given_files/rdparse.rb` finns den parser ni jobbat med under seminarierna.

Lägg till möjlighet att tilldela en variabel ett värde och slå upp variabelns värde i `DiceRoller`-språket.

5 Constraint networks

1. I filen `given_files/constraint_networks.rb` har metoden för att i `ArithmeticConstraint`-klassen propagera värden mellan `Connector`-objekt ändrats. Fungerar det nya sättet? Om ja, förklara hur det fungerar. Om inte, varför inte?

2. Förklara varför metoden `user_assign` finns i klassen `Connector`. Förklara alltså inte vad den gör för den är löjligt liten, utan varför den behövs. Ge också kodexempel som förklarar.

6 Kompilering

Ändra på `DiceRoller`-klassen i filen `rdparse.rb` så att man skapar ett abstrakt syntaxträd, det vill säga en representation av strukturen hos de uttryck som läses in, istället för att returnera värden. Ni kan välja att använda vilken representation ni vill för de aritmetiska uttrycken.