

Tentamen i TDP007

2008-03-25

Ola Leifler

25 mars 2008

Översikt

Tentamen utförs 08:15 - 12:00. För betyg 3 krävs 50% rätt, för betyg 4 66% och för betyg 5 83%. Varje fråga ger som mest 4 poäng.

0.1 Inlämning av era lösningar

När ni är klara ska ni lämna kvar filer i er hemkatalog där ni loggar in enligt `<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
```

1 Objekt & block

1. Vi vill se till att det endast kan finnas en instans av en klass `Singleton`. Användare ska alltså inte kunna skapa flera instanser av klassen med hjälp av `new`. Använd `Module.private` för att implementera detta. Ge även kodexempel som visar hur klassen ska användas.
2. Kodblock kan använda sig av mer än enbart de argument som skickas till blocket, men hur slås referenser till andra variabler upp? Används den miljö i vilken kodblocket skapades eller den miljö kodblocket anropas ifrån? Skapa testfall (med hjälp av biblioteket `Test::Unit`) som visar vad som gäller.

2 Reguljära uttryck

I filen `adresser.rb` finner du ett antal adresser. Parsa dem med reguljära uttryck för att extrahera information om gata, gatunummer, eventuellt våningsplan, postnummer samt postort. Med informationen som extraherats, skapa adressobjekt som har attribut som motsvararfälten. Verifiera din implementation med testfall (en `Test::Unit::TestCase`-klass).

3 XML

I filen `ruby-lang.html` som är källkoden till hemsidan för språket Ruby finns det tre Ruby-uttryck (leta efter en `div`-tagg med `id=code`) med inbäddad formatering. Extrahera uttrycken med hjälp av REXML för att hitta HTML-elementen med programkod samt använd reguljära uttryck för att ta bort HTML-entiteter såsom ` `; som endast används för formatering. Evaluera slutligen Rubykoden och visa resultatet.

4 DSL

I filen `cloaker.rb` finner ni implementationen av ett DSL för att skapa HTML-kod programmatiskt.

1. Ge ett lite större kodexempel som illustrerar syntaxen och förklara utifrån exemplet hur implementationen fungerar.
2. Förklara varför metoden `cloaker` finns och visa även att metoden går att ersätta med något mycket enklare i kodexemplet.

5 Constraint networks

I filen `constraints.rb` finns det ett enkelriktat constraintnätverk. Vi skulle vilja veta lite mer exakt varför variabler ändrar sina värden i nätverket och vill därför förfina loggningen så att den talar om *varför* en `Wire` får ett nytt värde, det vill säga vilka indata som påverkar utdata. Dessutom vill vi bara veta när utdata *ändrar* sitt värde, inte annars.