

TDP007 - Dugga 2

2025

Regler

- All kod efterrättas på denna dugga
- Inga elektroniska hjälpmedel får medtas. Mobiltelefon ska vara avstängd och ligga i jacka eller väska.
- Inga ytterkläder eller väskor vid skrivplatsen.
- Student får lämna salen tidigast en timme efter start.
- Vid toalettbesök ska vakt informeras.
- All form av kontakt mellan studenter under tentamens gång är strängt förbjuden.
- Böcker och anteckningssidor kan komma att granskas av assistent, vakt eller examinator under tentamens gång.
- Frågor om specifika uppgifter eller om tentamen i stort ska ställas via tentasystemets kommunikationsklient.
- Systemfrågor kan ställas till assistent i sal genom att räcka upp handen.
- Endast uppgifter inskickade före tentamenstidens slut rättas.

Hjälpmedel	En Rubybok (exempelvis the pickaxe) Ett A4-ark med egna anteckningar Tillgång till webresurser: ruby-docs, rubular och tidig version av kursboken
------------	--

Information

Betygssättning vid tentamen

Tentamen består av 3 delar. Del 1 är uppgifter på grund-nivå som relaterar till tekniker från kursens första två seminarier, dessa bedöms som G/U. Del 2 är uppgifter på grund-nivå som relaterar till tekniker från kursens andra två seminarier, dessa bedöms som G/U. Del 3 består av mer avancerade uppgifter, där varje uppgift relaterar till någon teknik från kursens fyra seminarier. Uppgifter på del 3 ger upp till 10 poäng. Betyg på tentamen ges enligt följande tabell. För alla betyg så måste förutom de insamlade poängen alla uppgifter på Del 1 och Del 2 lösas med godkänt betyg.

Insamlade poäng Del 3	Betyg
10	3
17	4
24	5

Duggor

Den första duggan består av två delar. Del 1 är uppgifter på grundnivå som relaterar till tekniker från kursens första två seminarier och bedöms som G/U. Del 2 består av en mer avancerad XML-uppgift som ger upp till 10 poäng. Om alla uppgifter på Del 1 har godkänt betyg från denna dugga så har du Del 1 på ordinarie tentamen tillgodo.

Den andra duggan består av två delar. Del 1 är uppgifter på grundnivå som relaterar till tekniker från kursens sista två seminarier och bedöms som G/U. Del 2 består av en mer avancerad parser-uppgift som ger upp till 10 poäng. Om alla uppgifter på Del 1 har godkänt betyg från denna dugga så har du Del 2 på ordinarie tentamen tillgodo.

Om del 1 är avklarad på båda duggorna och du har samlat in totalt 14 poäng från duggorna så har du förutom Del 1 och Del 2 på ordinarie tentamen tillgodo också betyg 3 på tentamensmomentet.

Inloggning

Logga in på datorn i labbsalen med ditt liu-id och lösenord. Detta är samma inloggningsuppgifter som du använder i Lisam.

Skrivbordsmiljön

När du kommit in i tentasystemet har du en normal skrivbordsmiljö (Mate-session i Ubuntu). Efter en stund kommer även din kommunikationsklient att dyka upp automatiskt. Startmenyn är nedskalad till enbart det som examinators bedömt relevant. Andra program kan fortfarande finnas tillgängliga genom att starta dem från en terminal. Observera att en del program som använder nätverkstjänster inte fungerar normalt, eftersom nätverket inte är åtkomligt.

När du är inloggad är det viktigt att du har tentaklienten igång hela tiden. Om den inte dykt upp fem minuter efter inloggning och inte heller när du startar den manuellt från menyn (fisken) tar du kontakt med assistent eller vakt i sal.

Terminalkommandon

`ruby` används för att köra en ruby-fil.

`irb` används för att starta ruby i interaktivt läge.

`ri` används för att komma åt den inbyggda dokumentationen.

Ruby-docs

På tentan har du tillgång till referenssidorna på <https://ruby-doc.org/> (både core och std-lib finns tillgänglig) via webbläsaren Chrome. Starta `chromium-browser` i terminalen eller välj lämpligt alternativ från startmenyn. Observera att allt utom referenssidorna är avstängt. Om du inte kan komma åt en sida du tycker hör till referenssidorna (som kanske blockerats av misstag) kan du skicka ett meddelande via tentaklienten. Tag hjälp av assistent i sal om det inte fungerar.

Rubular

På tentan har du tillgång till sidan <https://rubular.com/>.

Givna filer

Eventuella givna filer finns i katalogen `given_files`. Denna underkatalog är skrivskyddad, så det är ingen risk du råkar ändra på dessa filer. Skrivskyddet gör dock att du måste kopiera in de givna filer du vill använda till tentakontots hemkatalog. Hur du listar och kopierar filer ska du kunna. Hemkatalogen står du i från början, och du kommer alltid tillbaka till den genom att bara exekvera kommandot `cd` i terminalen.

Avslutning

När du skickat in alla uppgifter och känner dig färdig kan du logga ut och lämna salen. Antalet poäng meddelas i efterhand via webreg.

Avsluta alla öppna program och logga ut ur datorn. Lämna inte datorn innan du ser att du är utloggad.

Del 1: Uppgift 1 - Parser AST

I filen `uppgift1.rb` finns parsern som vi är bekanta med från labbserien. Diceroller-språket har ersatts med ett språk som heter **Arithmetic Parser** som egentligen är en förenklad version av Diceroller. Språket hanterar nu bara addition, multiplikation och parenteser. Ditt jobb är att skriva om detta språk så att ett AST returneras istället för resultatet. När det abstrakta syntaxträdet returnerats ska det evaluera så att resultatet skrivs ut. Var utvärderingen av din AST sker väljer du själv, men hela trädet skall byggas innan du utvärderar det. Du får själv välja hur du representerar det abstrakta syntaxträdet. Utskriften behöver innehålla korrekta resultat av beräkningarna på ett enkelt läsbart sätt, men behöver i övrigt inte stämma överens med de Funktionella kraven.

Funktionella krav:

```
AritParser, evaluating: 1+2 => 3
AritParser, evaluating: 2*3+2 => 8
AritParser, evaluating: (1+2)*3 => 9
AritParser, evaluating: ((1+2)+2)*2 => 10
```

Del 1: Uppgift 2 - DSL

I filen `uppgift2.rb` finns den påbörjade dsl-läsaren vi känner igen från labbserien. I den givna filen `dsl.rb` finns ett litet domänspecifikt språk för konfigurationsfiler. På en rad skriver man kommandot `new_config` och skriver namnet på konfigurationen i form av en sträng. Fram tills att detta kommando skrivits igen tillhör alla rader denna konfiguration. När en ny konfiguration skapats kan man lägga till vilka ord man vill (följer reglerna för variabelnamn i ruby) följt av vilken enskild data man vill (e.g. en sträng eller ett heltal). Du skall lösa så att denna typ av filer kan läsas in och lagras i en associativ behållare (se funktionella krav). Problemet ska lösas med `method_missing` för alla rader. Du får dock skapa en separat funktion för `new_config` om du vill.

Du ska sedan returnera behållaren och skriva ut den. Formatteringen av utskriften behöver inte matcha de funktionella kraven. Det spelar heller ingen roll om nycklarna är av typen symbol eller sträng. Men strukturen av behållaren och datan där i skall stämma med kraven.

Funktionella krav:

```
{
  :development=>{
    :width=>1920,
    :height=>1080,
    :windowname=>"My Game (development)"
  },
  :release=>{
    :width=>800,
    :height=>600,
    :windowname=>"My Game"
  }
}
```

Del 1: Uppgift 3 - Constraints

I den givna filen `uppgift3.rb` hittar du constraint-nätverken vi arbetat med i labbserien. Du kan anta att de aritmetiska constraintsen är fullt fungerande och att du bara behöver använda constraints av typen `Adder` och `Multiplier`.

Formeln för enkel ränta är $I = P \times r \times t$. Där variablerna är:

`I(räntan)`,

`P(investerade pengar)`,

`r(räntesatsen uttryckt i procent/100, 5 procent ränta skrivs alltså som 0.05)`,

`t(tiden i antal år)`.

Ditt jobb är att implementera funktionen `interest_constraint` i den givna filen `uppgift3.rb` så att huvudprogrammet fungerar som förväntat. Funktionen ska sätta upp ett constraint-nätverk som motsvarar ekvationen för enkel ränta. Funktionen tar parametrarna `interestRate` och `years` och `Connectors` som motsvarar `r` och `t` ska sättas till värdena av de parametrarna. Funktionen ska sedan returnera `Connectors` som motsvarar `I` och `P` i ekvationen.

Funktionella krav:

```
i after assigning p=100: 50.0
p after assigning i=45: 90.0
```

Del 2: Uppgift 4 - Parser

Parseruppgifter på denna nivå har varit vanligt förekommande på tidigare tentor i kursen. Se ett urval av dessa för exempel på nivå, börja med nyaste och arbeta dig tillbaka i tiden.