

TDP007 - Dugga

2020-03-10

Regler

- All kod efterrättas på denna dugga
- 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 via Zoom
- 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 ett antal uppgifter på varierande nivå. Uppgifter som uppfyller specifikationen samt följer god sed och konventioner krävs för maxpoäng på en uppgift. Avvikelse från ovanstående ger avdrag.

Duggan består av uppgifter (några indelade i deluppgifter) som totalt kan ge 25 poäng. Dessa poäng summeras ihop med poängen från den andra duggan.

Ruby-docs

På tentan har du tillgång till referenssidorna på <https://ruby-doc.org/> (både core och std-lib finns tillgänglig) via webbläsaren.

Rubular

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

Givna filer

Eventuella givna filer finns länkade från kurssidan.

Avslutning

När du skickat in alla uppgifter och känner dig färdig kan du stänga tetaklienten, lämna Zoom och stänga thinlinc.

Uppgift 1 - DSL (8p)

I den givna filen `dsl.rb` hittar du ett domnänspecifikt språk. Här är ett utdrag ur filen.

```
pizza consists_of sauce
pizza consists_of dough
pizza can_consist_of cheese

dough consists_of flour
dough consists_of yeast
```

I detta DSL kan man specificera delar av maträtter genom att exempelvis skriva att pizza består av sås, deg och eventuellt ost. Deg består i sin tur av mjöl och jäst. Ingen av dessa består dock själva av några delar. Du ska nu implementera en läsare för detta språk med hjälp av `method_missing`. I den givna filen `1.rb` finns en påbörjad implementation och du behöver lägga till och/eller ändra de medlemsfunktioner som krävs för att få det hela att fungera.

Resultatet av att läsa in en fil ska vara en datorstruktur. Varje ingrediens som består av andra ingredienser ska sparas som en nyckel i en hash. Värdet är sedan associerat med ingredienser som måste finnas för att kunna framställa nyckeln och sådant som man kan använda om man vill. Se exemplet nedan för hur en del av hashen ska se ut vid utskrift efter programmet kört:

```
{
  "pizza" : {
    "required" : [
      "sauce",
      "dough"
    ],
    "optional" : [
      "cheese"
    ]
  },
  ...
  "dough" : {
    "required" : [
      "flour",
      "yeast"
    ]
  },
  ...
}
```

För full poäng ska godtyckliga ingredienser kunna läggas till utan att din läsare behöver modifieras. För full poäng ska inte det givna domänspecifika språkets struktur modifieras.

Uppgift 2 - Icke-deterministisk programmering (6+2p)

Praktisk (6p)

Jonna försöker gå ut och handla så lite som möjligt nu under pandemin. Men hon tycker det är svårt att komma på vad hon kan laga för mat med ingredienserna som hon har hemma. Du har därför fått i uppgift att skriva ett program som utifrån en hash av recept och en array av ingredienser kan komma fram till vilka maträtter som går att tillaga. I den givna filen `2.rb` finns det kod för en problemlösare som löser problem med hjälp av `calcc`. Ditt jobb är att skriva funktionen `what_can_i_cook` som med hjälp av denna problemlösare skriver ut alla recept som Jonna kan tillaga med sina tillgängliga ingredienser.

Programmet ska fungera som förväntat oavsett vilken hash av recept och array av ingredienser som stoppas in. Men du kan anta att de är på samma format som i den givna filen. Utskriften av programmet bör se ut så här om det körs med datan i den givna filen:

```
You can cook the following recipes
pizza
borsjtj
bread
Total of 3 possibilities.
```

Teoretisk (2p)

Du kan svara på denna fråga i form av en separat text-fil. Den teoretiska och praktiska delen skickas in som samma uppgift i tentasystemet.

Besvara följande fråga, var kortfattad, informerande och motivera ditt svar.

- Efter du löst den praktiska delen av uppgift 2 så har du använt problemlösaren för att ta fram vilka rätter som kan tillagas med de tillgängliga ingredienserna. Vilka förändringar behöver du göra (i hela filen) för att också kunna hantera mängder av ingredienser i programmet. Det spelar ingen roll att jag har lök hemma om jag har 2 lökar och receptet kräver 4.

Uppgift 3 - Parser (6+3p)

Praktisk (6p)

I den givna filen `3.rb` finns `rdparse` och `dicerollerspråket`. Ditt jobb är att modifiera denna parser för att tolka recept istället för tärningar. Användaren av språket kan antingen lägga till en ingrediens till receptet genom att skriva `ingrediens: mängd` eller slå upp hur mycket av en existerande ingrediens som finns genom att skriva `ingrediens`. En lite speciell del är att man även ska kunna använda existerande ingredienser för att uttrycka mängden av nya ingredienser.

För full poäng ska parserna ska hantera addition, subtraktion, division och multiplikation. Dessa ska fungera med korrekt associativitet och prioritet. För full poäng ska det även gå att styra prioritet med parenteser, mycket av detta finns dock löst i `diceroller-språket`.

Körexempel:

```
[recipeparser] flour: 3
[recipeparser] yeast: 1
[recipeparser] dough: flour + yeast
[recipeparser] tomatoes: flour * 2
[recipeparser] dough
4
[recipeparser] tomatoes
6
[recipeparser] sauce: flour * (2 + 1)
[recipeparser] sauce
9
```

Teoretisk (3p)

Du kan svara på denna fråga i form av en separat text-fil. Den teoretiska och praktiska delen skickas in som samma uppgift i tentasystemet.

Besvara följande fråga, var kortfattad, informerande och motivera ditt svar.

- I det nuvarande programspråket kan användaren antingen spara undan ingredienser eller slå upp existerande ingredienser. Enheten för mängden av dessa ingredienser framgår dock inte. Förklara vilka ändringar du skulle behöva göra för att kunna ange mängden. Jag vill kunna ange mängden på följande sätt: `flour: 3dl`. Även med andra vanliga enheter. Förklara vilka utmaningar du föreställer dig finns och hur du tolkar beteendet. Använd gärna korta exempel för att motivera ditt svar.