

TDP007 - Dugga 1

2026-02-04

Regler

- All kod liverättas under denna examination
- 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 vakt 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 2 delar. Del 1 är uppgifter på grund-nivå som relaterar till tekniker från kursens seminarier. Del 3 består av mer avancerade uppgifter, där varje uppgift relaterar till någon teknik från kursens seminarier. Alla uppgifter bedöms som G/U. Betygsättning vid tentamen görs enligt följande tabell (för alla betyg 3, 4, 5 krävs att alla uppgifter på grund-nivå är avklarade):

Avklarade uppgifter del 2	Betyg
0	3
1	4
2	5

Duggor

Varje dugga motsvarar halva tentan. Dugga 1 tar upp koncept från kursens första 2 seminarier. Dugga 2 tar upp koncept från de sista 2 seminarierna. För varje avklarad del under en dugga tillgodoräknas motsvarande uppgifter på första ordinarie tentamen efter kursen. Exempel: (1) Som student har jag klarat endast del 1 av dugga 2 i kursen. Om jag skriver ordinarie tentamen behöver jag då inte göra motsvarande hälften av del 1 på tentamen. (2) Som student har jag klarat av del 1 och del 2 på både dugga 1 och dugga 2. Jag har nu betyg 5 på tentamen och det finns ingen anledning för mig att skriva tentan.

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 examinator 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.

Webresurser

På tentan har du tillgång till referenssidorna, rubular och en version av kursboken. Klicka på web access på skrivbordet.

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 är nöjd med din insats (inte tänker arbeta vidare) kan du avsluta.

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

Del 1: Uppgift 1 - Regexp och grunder

I den givna filen **contact.html** hittar du en html-sida som innehåller liu-id:n för personalen i TDP007. Ditt jobb är att skriva ett Ruby-skript som extraherar alla liu-id:n från filen och skriver ut fullständiga e-postadresser för varje **unik** liu-id. Personal (2 siffror i id) har adresser som slutar på @liu.se och studenter (3 siffror i id) har adresser som slutar på @student.liu.se.

Funktionella krav, ordning spelar ingen roll:

```
ponha45@liu.se
aroni22@liu.se
lovar08@liu.se
emili26@liu.se
aleno92@liu.se
casav929@student.liu.se
gorbe161@student.liu.se
jeswa278@student.liu.se
```

Del 1: Uppgift 2 - SAX-parser

Använd SAX-parser för att hämta texten från alla rubriker (h1, h2, h3) i filen **contact.html**. Ja, vi använder en XML-parser för att parse HTML 5 här, men denna sida råkar vara skriven i en (mestadels) well-formed HTML5 som är XML-compliant.

Datan ska sparas i en hash där nycklarna är :h1, :h2 och :h3 och värdena är arrayer med textinnehållet från respektive element. Hämta hashen från din SAX-lyssnare efter du parsat filen och skriv ut hashen på valfritt läsbart sätt.

Funktionella krav:

```
{:h1=>
  [ "IDA - Institutionen för datavetenskap",
    "TDP007 Konstruktion av datorspråk" ],
:h2=>
  [ "Examinator",
    "Kursledare",
    "Kursassistent",
    "Kursadministratör",
    "Övriga lärare",
    "Studierektor" ],
:h3=>
  ["Kontakta"]}
```

Del 1: Uppgift 3 - DOM-parser

Vi jobbar i denna uppgift med den givna filen **contact.html**.

Skapa en klass som representerar en länk, en länk lagrar två datamedlemmar av typen sträng: länktexten och vad länken pekar på. En länk kan också ha övrig lämplig funktionalitet som du behöver.

När klassen är färdig ska du sedan skapa en fristående funktion som tar emot en fil och ett **id_filter**. **id_filter** är en sträng som vi kommer använda i funktionen. Funktionen ska med en DOM-parser hitta alla html-länkar (element `<a>`) som har ett element `<div>` som anfa-der¹(eng: *ancestor*). Men bara där `<div>` har attributet **id** satt till samma sak som parametern **id_filter**. Skapa en instans av din länk-klass för varje länk du hittar och lagra dessa i en lämplig behållare. Funktionen ska returnera denna behållare av länkar.

Anropa den fristående funktionen du skapat i huvudprogrammet med filen **contact.html** och strängen **"foot"** som parametrar, skriv ut alla länkarna som funktionen returnerar på formatet **länktext: länkadress**.

Du behöver använda minst ett xpath-uttryck och ta rimlig hänsyn till att strukturen av HTML-dokumentet kan ändras, så du ska inte hårdkoda hela sökvägen.

Funktionella krav:

```
Kontakta IDA: /department/contact/index.sv.shtml
Kartor: /department/location/index.sv.shtml
Om webbplatsen: https://liu.se/om-webbplatsen
Om LiU: https://liu.se/om-liu
Utbildning: https://liu.se/utbildning
Forskning: https://liu.se/forskning
Samverkan: https://liu.se/samverkan
LiU Student: https://www.student.liu.se/?l=sv
LiU Anställd: https://intranet.liu.se/
LiU Alumni: https://liu.se/samverkan/alumni
Biblioteket: https://liu.se/biblioteket
Lediga jobb: https://liu.se/jobba-pa-liu
Press: https://liu.se/nyheter/presskontakter
LiU Electronic Press: https://www.ep.liu.se/
Grundutbildning: /edu/ugrad/index.sv.shtml
Exjobb: /edu/ugrad/thesis/index.sv.shtml
Forskarutbildning: /edu/grad/index.sv.shtml
Organisation: /department/org/index.sv.shtml
Internwebb: /local/ida/index.sv.shtml
Akut: /department/emergency/index.sv.shtml
Till toppen : #
```

¹Ett elements anfader är alla element som föregår elementet. Exempelvis är elementet `<body>` anfader till alla andra element i **contact.html**, eftersom detta är rot-elementet.

Del 2: Uppgift 4 - DOM-parser

I filen `Imperium-Imperial-Knights_Library.xml` hittar du samma armé-lista som vi arbetat med i labbserien. I dessa armélistor finns profiltyper lagrade som hänvisar vidare till mer information. Imperial Knights har två profiltyper som ligger i `catalogue->profileTypes`. I detta fall är dessa **Deed** och **Quality**. Alla dessa profiltyper har sedan ett id lagrat som ett attribut.

Ditt jobb är att skriva ett Ruby-skript som använder en DOM-parser (du får också använda SAX-parsern, men vi rekommenderar det inte) för att hämta id:t och namnet på de två profiltyperna. Hitta sedan alla element i dokumentet som har attributet **typeId** satt till något av dessa id:n. Lagra dessa i en Hash med namnet på profiltyperna (i just detta fall **Deed** eller **Quality**) associerat med texten (lagrad i attributet **name**) för varje element som matchar profiltyperna.

För andra armélistor kan det finnas fler eller färre profiltyper så skriv en rimligt generell lösning som fungerar oavsett hur många profiltyper som finns i dokumentet. Skriv ut din Hash på ett läsbart sätt. Slå sedan samman varje deed och quality så du skriver ut fullständiga meningar med riddarnas eder (se funktionella kraven, du kan anta att det finns två profiltyper här och att dessa har samma antal texter associerade med sig).

Funktionella krav:

```
{ "Deed"=>
  [ "We pledge to reap a great tally...",
    "We swear to reclaim the realm...",
    "We vow to lay low the tyrant..." ],
  "Quality"=>
  [ "...and we are eager for the challenge.",
    "...with our martial valour risen over all.",
    "...yet shall our legacy be unsullied." ]
}
```

```
We pledge to reap a great tally.....and we are eager for the challenge.
We swear to reclaim the realm.....with our martial valour risen over all.
We vow to lay low the tyrant.....yet shall our legacy be unsullied.
```