TDP007 - Dugga 1

2020-02-01

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. Avvikelser 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 - SAX-parsning (8p)

I den givna filen `dungeoncrawlers.xml` hittar du en xml-fil som extraherats genom en hemsidas API. Detta dokument innehåller alla brädspel som tillhör kategorin dungeoncrawlers från den hemsidan. Varje brädspel representeras av ett element av som heter `link`. Var och en av dessa element innehåller information om brädspelets `id`(unikt id) och `value`(namn). Ditt jobb är att implementera en strömparser (sax-parser) som plockar ut id och value för varje brädspel i filen. När parsern är färdig ska det vara möjligt att plocka ut all denna data i form av en tabell (`Hash`).

I filen `sax_example.rb` finns ett exempel på en påbörjad sax-parser.

Nedan följer ett exempel på hur tabellen kan se ut när parsern är färdig:

```
"1758"=>"Advanced Heroquest",
"270871"=>"Agemonia",
"181495"=>"Alone",
"309721"=>"Alone Against Fear",
"304730"=>"Alone: Avatar Expansion",
"304729"=>"Alone: Deep Expansion",
"273703"=>"Altar Quest",
"324631"=>"Altar Quest: Stretch Goals",
"324632"=>"Altar Quest: The First Four Hero Pack",
"155068"=>"Arcadia Quest",
"156089"=>"Arcadia Quest: Beyond the Grave",
...
```

Uppgift 2 - DOM-parsning (9p)

Gloomhaven är ett synnerligen populärt brädspel och en av de hetaste titlarna på boardgamegeek. En xml-fil som heter `gloomhaven.xml` finns som given fil. Ditt jobb är att skapa en klass som representerar ett brädspel. Denna klass ska vara väl designad enligt de principer som finns för objektorienterad design.

I ditt huvudprogram skall du sedan parsa filen `gloomhaven.xml` med en dom-parser och extrahera följande information:

- Alla spelets namn.
- Rekommenderad minsta ålder för att spela spelet.
- Namnet på spelets designer eller designers

I dagsläget skall objekt av denna typ bara skriva ut det primära namnet (I detta fall Gloomhaven) men vi behöver fortfarande lagra alla alternativa namn för framtida utbyggnad.

Det finns lite information om metoderna i klassen REXML::Element samt några exempel på XPath-uttryck i filen rexml.txt.

Du kan hitta data om rekommenderad ålder för spelare i elementet `poll` som har attributet name som är suggested_playerage. Rekommenderad ålder fastställs genom en omröstning så du behöver välja den ålder som har flest antal röster. För att hitta designers, sök efter attributet `type="boardgamedesigner"`. Din lösning ska självklart vara generell nog för att kunna parsa information om andra brädspel. Testa gärna att köra din lösning på filen `loveletter.xml` också.

```
Game: Gloomhaven
Recommended for ages 14 and up
Designed by: Isaac Childres
```

Uppgift 3 - Strukturerad text (8p)

I den givna filen `3.rb` finns en sträng som sträcker sig över flera rader. Denna sträng representerar en tabell över karaktärer och deras attribut i spelet Gloomhaven i min nuvarande brädspelskampanj. Ditt jobb är att skriva ett program som kan läsa in denna tabell (eller andra liknande tabeller).

Programmet ska sedan samla in data om varje karaktär i form av karaktärens namn och de första 3 kolumnerna med siffror. Med andra ord behöver namn, str, agi och sta samlas in och sparas för varje karaktär och sparas i en lämplig datastruktur.

Databehållaren skall sedan sorteras efter genomsnittet av de tre attributen i stigande ordning och skrivas ut enligt nedanstående format.

Körexempel:

```
Goliath: 10 8 10 - 9.3
Tinker: 9 7 6 - 7.3
Enchantress: 5 4 5 - 4.7
Spellweaver: 4 5 4 - 4.3
```