

TDP007 - Dugga 1

2026

Regler

På duggor och tentor som går i kursen kommer de inledande sidorna innehålla regler, dessa förmedlar bara det som presenteras på kurssidan så vi hänvisar dit. Vill du se hur detta ser ut kan du titta på en tidigare dugga/tenta som har gått på kurssidan.

Del 1: Uppgift 1 - Regexp

Skriv en funktion i den givna filen `uppgift1.rb`, som tar emot en godtyckling sträng som argument, och använder Regex för att plocka ut alla e-postadresser från strängen i fråga. Funktionen ska returnera en hash där varje nyckel är en godtycklig e-postdomän (t.ex. `@liu.se`), och värdet är hur många e-poster som hade denna domän. Du kan utgå från att alla domäner slutar på `.se` för denna uppgift. Det finns en given sträng i filen som du kan anropa funktionen med och se att du får samma svar som i de funktionella kraven.

Funktionella krav:

```
{"luddigt.se"=>2, "blocket.se"=>3, "hittahit.se"=>2,
"foobar.se"=>3, "somewhere.se"=>1}
```

Del 1: Uppgift 2 - SAX-parser

Utgå från den givna filen `boardgames.xml` använd en sax-parser för att läsa ut varje spels namn, genre och minsta antal spelare. Datan lagras i valfri struktur och ska skrivas ut på valfritt tydligt format. Observera att det är viktigt att utskriften sker efter datan för alla spelen lästs in. Ett exempel på utskrift finns i de funktionella kraven, din utskrift behöver inte se ut på samma sätt eller använda samma datatyper, men varje spel ska skrivas ut på en enskild rad.

Funktionella krav:

```
Games:
{:name=>"Terraforming Mars", :genre=>"Strategy", :min_players=>"1"}
{:name=>"Catan", :genre=>"Family", :min_players=>"3"}
{:name=>"Gloomhaven", :genre=>"Cooperative", :min_players=>"1"}
{:name=>"Codenames", :genre=>"Party", :min_players=>"2"}
{:name=>"Dominion", :genre=>"Deck-Building", :min_players=>"2"}
{:name=>"Azul", :genre=>"Abstract", :min_players=>"2"}
{:name=>"Arkham Horror: The Card Game", :genre=>"Thematic", :min_players=>"1"}
{:name=>"Viticulture", :genre=>"Eurogame", :min_players=>"1"}
{:name=>"Descent: Legends of the Dark", :genre=>"Adventure", :min_players=>"1"}
```

Del 1: Uppgift 3 - DOM-parser

Utgå från den givna filen `library.xml` och använd en dom-parser för att lösa denna uppgift. Lösningen ska fungera för en godtycklig xml-fil på samma format. Du kan exempelvis inte anta att vartannat element är en bok. Skriv ut titeln (`title`) på alla inventarier (`item`) som har attributet `type` satt till `book`. Du behöver använda minst ett x-path-uttryck i lösningen av denna uppgift.

Funktionella krav:

```
The Great Gatsby
A Brief History of Time
```

Del 2: Uppgift 4 - DOM-parser

Utgå från filen `boardgames.xml`. Skapa en klass som kan representera brädspel, ett brädspel behöver lagra två datamedlemmar: `name` och `publisher`. Du ska läsa ut `name` och `publisher` från brädspelen i filen med hjälp av en `dom-parser`, men vi är bara intresserade av brädspel som har minst 2 expansioner och en rating som är minst 8. För varje sådant spel ska du instansiera ett objekt av din klasstyp och lagra det objektet i en **Array**. I slutet av programmet ska spelen du hittat skrivas ut (se funktionella krav för formattering). Din lösning behöver ta hänsyn till att spelen kan ändras över tid (du kan alltså inte hårdkoda att du letar efter Gloomhaven och Descent). Du behöver använda minst ett x-path-uttryck i lösningen av denna uppgift.

Funktionella krav:

Gloomhaven - Cephalofair Games Descent: Legends of the Dark - Fantasy Flight Games
