

Inlämningsuppgifter i Programmering för testare, vecka tre

Veckans inlämningsuppgifter bifogas. Till mig senast måndag 130218 kl. 9:00. Denna veckan har ni möjlighet att arbeta med uppgifterna i par & lämna in dem gemensamt.

Jag föreslår att ni sätter av torsdagen till att arbeta med uppgift 33, som sammanfattar tidigare uppgifter i ett tillämpat exempel. Arbeta alltid efter principen att få något litet att fungera först, för att sedan bygga ut det med fler funktioner. Det gäller inte minst uppgift 33.

Jag har försökt att ange Godkänt eller Väl godkänt -nivå på deluppgifterna. Men, ni måste inte ha gjort alla uppgifter av ett slag för att få G eller VG i betyg på kursen.

Uppgift 31a (G)

Skriv en funktion som läser in ett användarnamn & ett lösenord från tangentbordet, och sparar dessa till disk i en eller flera filer.

Tips: Jämför inlämningsuppgift 1, öppna en fil för skrivning med `file = File.open("namn", "a+")`, skriv till den med `file.puts` eller `file << "utskrift"`. Glöm inte `file.close` när ni är klara.

Uppgift 32b (G)

Skriv en funktion som läser in par av användarnamn & lösenord från disk. Returnera en uppräkningslista (array av objekt eller en hash) som innehåller paren.

Tips: Spara användarnamn & lösenord i en User-klass, så att de "hör ihop". Spara sedan User-objekten i en Array. Alternativt läs in användarnamn & lösenord som nyckel-värde-par i en Hash.

Uppgift 32c (G)

Skriv en funktion som läser in ett användarnamn & ett lösenord från tangentbordet, och jämför dem med de tidigare sparade lösenorden du läste in från disk. Om namn & lösenord matchar, returnera true, annars false.

Tips: Dela upp arbetet i flera delar/funktioner: en funktion som läser in användarnamn & lösenord, en som läser in en de sparade lösenordsparen till en lista du kan iterera över, en funktion som hittar ett lösenordspar i listan givet ett användarnamn, en funktion som jämför de två paren är lika.

Om ni arbetar parallellt med uppgiften, bestäm först var respektive funktion ska ta in för data & vad den skall returnera för data. Skissa först så ni ser helheten, använd papper & penna eller en vit tavla.

Uppgift 32d (VG)

Skriv en funktion som tar in en sträng & använder reguljära uttryck för att se om det finns minst en siffra, en stor bokstav & en liten bokstav i strängen. Beroende på, returnera true eller false. Använd funktionen för att kontrollera lösenorden i uppgift a.

Tips: Det är okej att matcha flera gånger med olika regexp för att avgöra om strängen är okej.

Uppgift 31e (VG)

Modifiera uppgift a & c så att du inte längre sparar lösenorden i klartext. Använd Digest för att kryptera lösenorden, ev. tillsammans med ett salt, t.ex. tiden lösenordet skapades. Saltet måste då också sparas med användaren för att kunna återskapa lösenordet.

Tips: Se <http://ruby.about.com/od/advancedruby/ss/Cryptographic-Hashes-In-Ruby.htm>, det är den genererade hashen ni sparar, och som ni sedan jämför med hashen av användarens inmatade lösen.

Uppgift 32a (G)

Hämta hem `library_example.rb` från `git://github.com/kilskrift/pft-uppgift3.git`. Förklara med egna ord hur `initialize`-funktionen i `Book` & `Film`-klasserna använder `super()` för att skapa unika `id:n` i sina instanser. Hur fungerar funktionen `unique_id` & klassvariabeln `@@previous_id`?

Tips: Om du inte gjort övningarna om klasser & arv på Codecademy track 9, titta på dem först. Du kan kopiera ner en kopia av repot ovan med ”`git clone git://github.com/kilskrift/pft-uppgift3.git`”
Läs även i bokens kapitel 11 & 12 om klasser & arv, samt kapitel 22 om arv (Inheritance)!

Uppgift 32b (G)

Lägg till att `Book`-klassen lagrar sidantal i böckerna. Lägg även till att `Film`-klassen lagrar speltid i minuter. Ändra så att man kan läsa alla attribut som lagras i `Book` & `Film`, men inte sätta om dem.

Uppgift 32c (G)

Skapa en ny `LibraryItem`-klass för ljudböcker. De har en titel, en författare, en inläsare & en speltid i minuter.

Uppgift 32d (VG)

Ändra i `LibraryItem`-klassen så att `unique_id` alltid returnerar strikt ökande löpnummer, även mellan olika körningar av program som använder klassen.

Tips: Du kan anta att det är tillräckligt att lagra senaste löpnummer i en fil mellan körningarna.

Uppgift 32e (VG)

Sortera alla `LibraryItems` i en array efter den tid de tar att konsumera. Anta att du läser 50 sidor i timmen.

Tips: Gör först övningarna i Codecademy 5.4, `Sorting by blocks`, om du inte redan gjort dem.

(Uppgift 33 på nästa sida)

Uppgift 33

Du & din kamrat har fått i uppdrag att bygga ett system för företagets testlitteraturbibliotek.

Minst två slags artiklar finns i biblioteket: böcker & ljudböcker (G). Systemet kan även utökas för att hantera bloggartiklar & webbcasts (VG). Man skall kunna låna böcker & ljudböcker, medan bloggartiklar & webbcasts bara finns listade med webbadress & författare i i registret.

För att få låna en (ljud)bok måste man logga in med ett lösenord (VG), alternativt ange sin e-postadress (G). När en bok är utlånad skall systemet hålla reda på det.

(Artiklar i systemet kan vara hårdkodade i programmet, alternativt läsas in från filer på disk eller kunna matas in i systemet & sparas. Det skall finnas minst tre av varje artikel i systemet.)

a) Valmeny

Registerprogrammet ha en valmeny där man kan välja att gå till följande funktioner

1. Logga in användare
2. Lista artiklar i biblioteket
3. Låna (ljud)bok i biblioteket
4. Sök artiklar i biblioteket

b) Logga in användare

En användare kan logga in genom att ange sin epostadress & sitt lösenord, som matchas mot sparade lösenord från disk. (VG)

c) Lista artiklar i biblioteket

Systemet skriver ut samtliga artiklar, inkl. deras löpnummer. (G)

Systemet skriver även ut om en artikeln är utlånad eller inte. (VG)

Man kan välja att skriva ut en viss kategori av artiklar (böcker, bloggposter osv. (VG)

d) Låna (ljud)bok i biblioteket

För att få låna en bok måste man ange en epostadress. (G)

Systemet kräver att användaren är inloggad för att man skall få låna en bok. Systemet sparar då användarens epostadress med boken för att kunna veta vem som lånat den. (VG)

Användaren matar in löpnummer på en (ljud)bok för att låna den, då markeras den som utlånad. (G)

Systemet skriver ut namnet på artikeln & frågar efter bekräftelse först (VG)

Systemet hanterar bara utlånade böcker i minnet & återställs efter varje omstart av programmet (G)

Systemet sparar utlånade böcker på disk & återställs till tidigare tillstånd efter omstart (VG+)

e) Sök artikel i biblioteket

Skriv in en titel, författare eller url, & systemet söker efter matchande artiklar i systemet, samt visar dem på skärmen (VG).

Lycka till!