

Inlämningsuppgift för moment 3

Kursen DT1033, höstterminen 2013, provdatum den 13 januari 2014.

Inlämningsuppgiften består dels av obligatoriska uppgifter som du måste klara för att bli godkänd på moment 3 och samt extra uppgifter som du kan lösa för att samla poäng inför momentets skriftliga prov. Det skriftliga provet ingår i tentamen i slutet av kursen. Betyg för moment 3 avgörs av resultatet av den tentamen. Lösningar av extra uppgifter rättas endast om du har klarat de obligatoriska uppgifterna.

Uppgifterna ska du lösa på egen hand utan. Alla hjälpmedel tillåtna. Du får använda kursboken, titta på mina föreläsningar, söka på Internet och så vidare. Du får dock inte kopiera någon annans lösningar eller låta någon annan kopiera dina lösningar och du får inte hjälpa eller ta hjälp av någon annan person.

Otydlig kod, lösningar som inte fungerar, fungerar felaktigt eller enbart fungerar delvis eller under vissa situationer, lösningar som innehåller kod som inte bidrar till lösningen, användargränssnitt som är svåra att förstå, ger poängavdrag.

När du har löst uppgifterna så kopiera alla dina projektmappar och andra dokument till en tom katalog på din dator. Radera alla obj och bin kataloger i samtliga projektmappar. Packa därefter ihop hela katalogen till en zip-fil. Döp zip-filen med ditt användarnamn och sänd in den via Inlämning i kursens rum på Fronter.

När du har packat ihop dina lösningar till zip-filen så kontrollera att allt som du vill sända in kom med i zip-filen genom att packa upp den i någon annan katalog på din dator.

I Fronterummets Inlämning hittar du en mapp med namnet "Inlämningsuppgift 3, måndag den 13 januari 2014". Du ska sända in din zip-fil till den mappen.

Mappen "Inlämningsuppgift 3, måndag den 13 januari 2014" är öppen under den 13 januari från och med klockan 08.00 fram till klockan 23.59. Därefter stängs mappen vilket betyder att du inte längre kan sända in lösningar av inlämningsuppgift 3.

Om någonting skulle krångla eller är otydligt beskrivet så kan du nå mig via telefonnummer 0702-234307.

Obligatoriska uppgifter

De här tre uppgifterna måste du klara för att bli godkänd på provet. Inga poäng räknas för dina lösningar, utan lösningen för varje uppgift blir antingen underkänd eller godkänd.

Observera att samtliga program som du konstruerar ska ha grafiska användargränssnitt.

Uppgift 1

I den här uppgiften ska du visa att du kan skriva och använda en separat metod som returnerar ett värde och har en parameterlista för indata.

Konstruera ett program som omvandlar den totala volymen som ett antal kryddmått, teskedar och matskedar tillsammans representerar till motsvarande volym uttryckt i milliliter.

Programmets användare ska ange hur många kryddmått, teskedar och matskedar som ska ingå i en volym som ska omvandlas till milliliter. Programmets användargränssnitt ska bestå av ett grafiskt fönster.

Omvandlingen av en volym från kokboks mått till milliliter ska ske med hjälp av en separat metod. Antal kryddmått, teskedar och matskedar som utgör en volym ska överföras till metoden via dess parameterlista och metoden returnera motsvarande volym uttryckt i milliliter.

Du kan läsa mera om våra svenska kokboks mått via följande webbsida,
<http://sv.wikipedia.org/wiki/Kokboksm%C3%A5tt>

Uppgift 2

I den här uppgiften ska du visa att du kan lösa problem med hjälp av stegvis förfining, "Top Down design", genom att konstruera ett program med hjälp av den metodiken. Den färdiga lösningen ska innehålla minst två egna definierade metoder förutom de metoder som behövs för att programmets grafiska användargränssnitt ska fungera.

Använd stegvis förfining för att utveckla ett program som bedömer läsbarheten för en textsträng bestående av en eller flera meningar. Programmets användare ska kunna ange en textsträng vars läsbarhet ska analyseras av programmet. Analysen ska bestå i att beräkna hur många ord i genomsnitt som meningarna består av. Till exempel följande textsträng;

"Ett exempel. En mening avslutas med punkt. Punkt och slut."
består av tre meningar och tio ord. Genomsnittligt antal ord per mening blir $10/3 = 3,333333$.

Ju fler ord som i genomsnitt meningarna består av, desto mera svårläst brukar texten betraktas.

Förutsätt att före varje ord alltid finns ett mellanslags-tecken, ' ', utom för textsträngens första ord. Förutsätt att alla meningar avslutas med tecknet punkt, ' . '. Du behöver inte ta hänsyn till några andra blanktecken, skiljetecken, siffror, förkortningar och annat som kan komplicera analysen.

Redovisa din stegvisa förfining i form ett jsp-diagram eller ett dokument med punktlistor eller pseudokod, minst två steg ska redovisas. Den redovisade lösningen ska överensstämja med det program som du har byggt. Redovisa även projektmappen för det utvecklade programmet.

Uppgift 3

En Vasaloppsåkare vill att du bygger ett program åt honom som hjälper att hålla rätt på hans vikt. Kvällen innan han startar Vasaloppet ska han kunna starta programmet och med hjälp av det spara sin vikt uttryckt i kg till en textfil.

Därefter när han har åkt Vasaloppet ska han åter igen ange sin vikt till programmet för att programmet ska kunna skriva ut procentuell viktförändring i relation till den vikt som han sparade till textfilen före loppet.

Extra frivilliga uppgifter

Om du har löst de första 3 uppgifterna så fortsätt gärna med följande två uppgifter. Uppgifterna är poängsatta. Vid rättningen så kommer poäng dras för brister i dina lösningar. De poäng som du får kan du räkna till godo vid momentets skriftliga prov i den tentamen som kursen avslutas med. De 10 poäng som du totalt kan få för uppgift 4 och 5 utgör 50% av den maximala poäng som du kan få för de betygshöjande uppgifter som ingår i tentamen för moment 3.

Observera att samtliga program som du konstruerar ska ha grafiska användargränssnitt.

Uppgift 4 (5 poäng)

Skapa ett program med ett grafiskt fönster. Via det grafiska fönstret ska det vara möjligt att skapa ett konto, sätta in och ta ut pengar från kontot och kontrollera dess saldo. Det ska också vara möjligt att spara innehållet i kontot till en textfil och läsa tillbaka innehållet från en tidigare sparad konto in till programmet.

Programmet ska vara uppbyggt av minst två klasser, en klass som representerar det grafiska fönstret och en klass som beskriver ett konto. All kommunikation med programmets användare ska ske via det grafiska fönstret. Programmets konto ska representeras av en instans av klassen Konto. Samtliga medlemsvariabler ska vara inkapslade.

I samband med att kontot skapas ska kontot via dess konstruktor initieras med ett kontonummer och en pin-kod. Kontots saldo ska vara tomt.

Det ska gå att se kontonummer och aktuellt saldo i programmets fönster och det ska gå att sätta in pengar till kontot. Kontot ska endast uppdatera sitt saldo om beloppet är rimligt.

För att ta ut pengar från kontot måste man först ange pin-kod till kontot, i annat fall ska kontot inte släppa ifrån sig några pengar. Observera att kontot saknar kredit, i det fall kontot är tomt ska det inte släppa ifrån sig några pengar.

Det ska gå att spara informationen från kontot till en textfil. Överlagra en konstruktor i klassen Konto så att det blir möjligt att återskapa ett konto utifrån tidigare sparad informationen.

Uppgift 5 (5 poäng)

Lös följande problem med hjälp av rekursiv metod.

Skapa en rekursiv metod som räknar hur många filer, med en viss filändelse, som finns i en katalog och den katalogens underkataloger. Exceptions som kan kastas i samband med sökningen av filträdet ska hanteras på lämpligt sätt. Användaren av programmet ska ange den katalog i vilken sökningen ska börja och filändelsen för den typ av filer som ska räknas, till exempel "*.cs".

Ett tips! Klassen `System.IO.Directory` kan innehålla användbara metoder.

Lycka till!

Hans-Edy