



Laboration 1 1DV404 – Planering och förbättring.

Uppgifterna skall lösas individuellt. Om ni använder er av material/lösningar som andra tagit fram skall detta tydligt framgå av redovisningen(referenser). Vi lägger stor vikt vid att du kan förklara och därmed visa på en förståelse för vad du gjort. Valfritt programmeringsspråk. Förutom källkod skall ni lämna kompilerad kod och dokumentation enligt beskrivningen i uppgifterna.

Ni skall föra en tidslog för hela laborationen, inte enbart för de uppgifter där det specifikt krävs. I loggen anger ni hur mycket tid ni förbrukat, vilken dag, och vad ni arbetat med. Sammanställ dessa siffror och bifoga er slutrapport.

Glöm inte att reflektion är en förutsättning för att man skall dra nytta av erfarenheter.

Redovisning.

Kontakta din handledare för redovisning av laborationen. **All redovisning sker på schemalagd tid!!! Gäller även distansstudenter.**

Paketera allt material i en fil. Namnge filen:
1DV404_HT14_[labgrupp]_[användarnamn]

Alla dokument skall lämnas in i PDF format. Skicka direkt till er laborationshandledare enligt deras instruktion.

Laborationen skall skickas in senast den 24:e november kl. 23.59
Boka redovisning senast i samband med det att ni skickar in!

Uppgift 1 – Tre “enkla” programmeringsuppgifter.

För varje uppgift

1. Planera arbetet i steg/moment, ange din planerade tid med en noggrannhet på 15 min. Exempelvis 3h och 15 min eller 4h och 30min. Beskriv vilken strategi du använder i din planering. Arbeta fram lämpliga formulär för det du skall dokumentera i steg 2.
2. Under tiden du genomför stegen/momenten dokumenterar du
 - a. tiden de tar.
 - b. vilka fel du gör, stora som små.
 - c. eventuella orsaker till avvikelser från planeringen.

3. När du är klar reflekterar du kring din planering och genomförandet.
(ca 1 A4, 1500 ord 12pt)

Uppgift 1a

Skriv ett program `RaknaA` som läser in en textrad från tangentbordet och sedan skriver hur ut många 'a' och 'A' det finns i raden. En körning kan se ut enligt följande:

Mata in en textrad: Alla barn fick högsta betyget A. Antal a: 3 Antal A: 2

Uppgift 1b

Skriv ett program `RaknaSiffror` som för ett godtyckligt positivt heltal N (läses in från tangentbordet) skriver ut antalet nollor, antalet udda siffror och antalet jämna siffror. En körning kan se ut enligt följande:

Mata in ett heltal: 6789500 Nollor: 2 Udda: 3 Jämna: 2

Uppgift 1c

Skriv ett program `NastStorsta` som läser in tio heltal och hittar det näst största talet av dem och därefter skriver ut det. En körning kan se ut enligt följande:

Mata in 10 heltal: 67 -468 36 1345 -7778 0 34 7654 45 -666

Det näst största talet är: 1345

Försök att göra programmet så att det är lätt att ändra antalet inmatade heltal.

Rekommendation: Använd färre än 10 heltal under tiden ni utvecklar programmet.

Notera: Du får inte använda arrayer (eller andra datastrukturer) i den här uppgiften.

Uppgift 2 – Förändring och förbättring

Uppgift kräver ca 1 A4, 1500 ord 12pt

2a. Ge exempel på några alternativa strategier som du kan använda i din planering av programmeringsuppgifterna.

2b. Två av anledningarna till att din planering avviker från verkligheten är dels felanvändningen av resurser, dels alla andra saker som inträffar. Hur kan du ta hänsyn till eller minska konsekvenserna av dessa? Ge konkreta exempel baserat på dina erfarenheter.

2c. Implementera två ”förbättringsåtgärder” i ditt planeringsarbete.

Uppgift 3 – Förbättrad planering av programmering

För varje uppgift

1. Planera arbetet i steg/moment. Beskriv vilken strategi du använder i din planering. Speciellt de förändringar du genomfört enligt *Uppgift 2c*. Modifiera dina formulär eller ta fram nya för din dokumentation.
2. Under tiden du genomför stegen/momenten dokumenterar du
 - a. tiden de tar.

- b. vilka fel du gör, stora som små.
 - c. eventuella orsaker till avvikelser från planeringen.
3. När du är klar reflekterar du kring din planering och genomförandet (ca 1/2 A4, 750 ord 12pt)

Uppgift 3a

Skriv ett program `Palindrome` som testar om en textrad (läses in från tangentbordet, enbart små bokstäver) är ett palindrom, dvs, att textraden har samma sekvens av bokstäver vare sig man läser den framifrån eller bakifrån. Ge några exempel som testar funktionaliteten.

Uppgift 3b

Skapa en klass `Fraction` som representerar ett bråkital på formen T/N där T (täljaren) och N (nämnaren) är heltal. Om nämnaren är noll skall ett felmeddelande lämnas. Det skall finnas följande medlemmar i klassen.

- En konstruktör som skapar och initialiserar ett nytt bråkital.
- Metoderna `getNumerator` och `getDenominator` som returnerar täljaren respektive nämnaren.
- Metoden `isNegative` som ger `true` om det är ett negativt bråkital.
- Metoderna `add` och `multiply` som utför motsvarande bråktalsoperation på två bråk och som returnerar ett nytt bråkital. Bestäm själv ett lämpligt sätt att hantera de fall där någon av de inblandade bråktalen har noll i nämnaren.
- `isEqualTo` som jämför två `Fraction`-instanser och ser om de representerar samma bråkital.
- `toString` som returnerar en strängrepresentation av bråket på form T/N .

Lägg gärna till fler metoder om ni tycker att något fattas. Lämpliga argument- och returtyper för de olika metoderna får ni bestämma själva.

Överkurs för den intresserade: Se till så alla bråkital alltid är maximalt förenklade. T ex, bråktalen $2/4$ och $35/50$ skall internt representeras som $1/2$ och $7/10$. D v s, den interna representationen skall alltid vara de två minsta möjliga heltalen T och N som representerar samma rationella tal T/N . Här kan det vara nyttigt att titta på Euklides algoritm (se Wikipedia).

Uppgift 4 – Planering

Nedan följer delar av ett visionsdokument för en produkt. Detta innebär att mycket information saknas. Detta är en del av utmaningen. Uppgifterna finns efter beskrivningen. Den här uppgiften förväntas ni inte lösa fullständigt utan den skall ses om ett underlag för er reflektion. Vi kommer ändå ställa krav på att ni visar att ni verkligen försökt genomföra en så fullständig planering som möjligt.

CloudPortfolio är ett nystartat företag. De vill erbjuda en produkt som tillåter användare att dela dokument. Alla dokument kommer att lagras i OOXML-formatet. Applikationen fungerar som ett gränssnitt mot ett dokument "moln" där registrerade användare lagrar och delar dokument.

När en användare loggar in i systemet visas filer och mappar. Systemet visar också en lista med de senaste ändringarna i dokumenten. Ändrade dokument färgkodas olika i listan efter typ modifiering och vilken användare som utfört den. Användaren som utfört ändringen måste också visas.

En användare kan ladda upp befintliga OOXML formaterade dokument i sin portfölj eller skapa dokument direkt med en enkel inbyggd textredigerare. Applikationen erbjuder även standardfunktionalitet för att administrera sin portfölj, exempelvis flytta, byta namn och ta bort dokument och mappar.

Dokument har en åtkomstkontrollista kopplad till sig som kontrollerar ägandet och nyttjanderätten. Endast ägare och administratörer kan ändra åtkomsten. En användare kan välja att dela ett dokument eller en mapp med andra CloudPortifolio användare. I så fall kommer ägaren att tilldela rättigheter och en inbjudan skickas till användaren.

Om en användare beslutar att lämna CloudPortifolio så skall alla filer och mappar som hon äger raderas. Om hon har skapat filer eller mappar och delat administrativa rättigheter med andra så kan de tas bort endast om alla användare med administratörsrättigheter är överens. De kommer då att få ett meddelande där de informeras och måste därefter komma överens innan processen kan slutföras. Om några administrativa användare inte samtycker till begäran kommer filen inte att tas bort. När raderingsproceduren är klar lämnar användaren CloudPortifolio och all användarinformation raderas.

4a. Planera arbetet för att släppa en första version av applikationen som beskrivs ovan. Utgå från att ni är 5 personer i gruppen.

- a. Vilka uppgifter måste ni genomföra
- b. Hur lång tid tar detta?
- c. Vem skall utföra det.

4b. Reflektera kring svårigheter i a planera en uppgift som denna (ca 1 A4, 1500 ord 12pt)