

DA339A Inlämningsuppgift 1

Syfte

Inlämningsuppgift 1 är en del av de moment i kursen som examinerar (delvis) följande lärandemål:

- Kunskap och förståelse
 - visa förståelse för ett algoritmiskt tankesätt
 - visa förståelse för strukturerad [...] programmeringsteknik
- Färdighet och förmåga
 - kunna använda strukturerad [...] programmeringsteknik vid program-utveckling
 - kunna dokumentera programvara

Inlämningsuppgift 1 utgör provkod 2003 *Inlämningsuppgifter del 1*. Provkoden utgör 1 hp på kursen och har betygsskala UG vilket innebär att studenten kan erhålla betyg U eller G på uppgiften.

Den muntliga redovisningen är en läraktivitet i kursen då studenten får återkoppling på inlämnad lösning. Vid redovisningen kan det vara så att en annan student som också ska redovisa närvarar och studenterna lyssnar på varandras redovisningar och återkoppling. Att se andras kod och andras lösningar är en viktig del i att lära sig programmera och detta är ett tillfälle för detta.

Redovisning

Inlämningsuppgift 1 redovisas genom skriftlig inlämning på kursplatsen på Canvas samt en muntlig redovisning.

Inlämning ska endast göras om du har en komplett lösning som du anser uppfyller kraven för uppgiften.

Du får inte lämna in en lösning som saknar delar och använda redovisningstiden för hjälp. Behöver du hjälp med inlämningsuppgiften nyttjar du handledningstider och/eller labbtider till hjälp för detta. Den som försöker redovisa en lösning som saknar delar kommer att avbrytas och hänvisas till nästa hjälptillfälle för hjälp och nästa omtillfälle för inlämning/redovisning.

Skriftlig inlämning

I den skriftliga redovisningen ska följande lämnas in:

- En fil med källkod: *Assignment1.java*
- En pdf-fil med namn enligt mönstret *da339a_InUpp1_förnamn_efternamn.pdf* (förnamn och efternamn ersätts med ditt namn) som innehåller en bild med ett aktivitetsdiagram. Aktivitetsdiagrammet får inte vara ritat för hand utan ska vara ritat med något verktyg, exempelvis Visual Paradigm (men måste inte nödvändigtvis vara ritat med VP).

Deadline för inlämning till det ordinarie tillfället är **söndag 10/10 23.55**. Inlämningar som lämnats in sent garanteras inte redovisning vid det ordinarie tillfället.

Om man inte lämnat in till det ordinarie tillfället eller blir underkänd vid den muntliga redovisningen för det ordinarie tillfället ges det två omtillfällen med deadline för inlämning:

- Måndag 1/11 23.55
- Augusti 2022, exakt datum ej bokat

Observera att detta är de enda examinationstillfällena vi garanterar för inlämningsuppgift 1.

Muntlig redovisning

För att göra den muntliga redovisningen måste en skriftlig inlämning gjorts inom deadline. Endast kompletta lösningar får redovisas.

Har du gjort en inlämning av den skriftliga delen bokar du en tid för redovisning via kalendern för kursen i Canvas. Se separata instruktioner på kursplatsen för hur du gör detta.

Inlämningsuppgiften redovisas muntligt och genom uppvisande av den inlämnade käll-koden och kompilering och exekvering av motsvarande kod. Studenten ska kunna visa upp sin källkod på dator för lärare/assistent som examinerar och kunna visa hur hen kompilerar och exekverar koden på dator. Studenten ska också kunna svara tillfredställande på frågor om sin kod och de lösningar hen implementerat samt kunna demonstrera förmåga att göra mindre ändringar i koden direkt och kompilera och exekvera igen (exempelvis ändra indata eller något gränsvärde). Studenten ska även kunna visa upp andra lösningar som krävs i inlämningsuppgiften exempelvis diagram, sanningstabeller eller andra skriftliga delar.

Redovisningen är en examination och är fokuserad på att göra en bedömning av den lösning och dokumentation av denna som studenten producerat. Det finns inte tid vid redovisningen att ge utförligare kommentarer om hela implementationen av uppgiften. Det finns vid denna redovisning inte heller tid att studenten beskriver hela sin tankeprocess eller förklarar hela programmet. Den lärare eller assistent som studenten redovisar för kommer att ställa frågor och be studenten demonstrera vissa saker. Diskussioner utöver detta finns det inte utrymme för vid redovisningen.

Önskar studenten återkoppling i större utsträckning på en lösning ges detta som vanlig hjälp vid laborationer.

Förberedelser för redovisning

Studenten förväntas vara förberedd när redovisningen startar och ha följande förberett innan redovisningen startar:

- Ha kommandotolk startad och redo att kompilera och exekvera källkoden för uppgiften som kan redovisas (det vill säga ha navigerat till den katalog du har dina filer i innan din redovisningstid startar).
- Ha källkodsfiler öppnade och tillgängliga i editor för att kunna visa upp koden och kunna göra ändringar i denna vid förfrågan.
- Ha övriga filer med efterfrågad dokumentation tillgänglig och redo att visas upp.

Generella krav vid redovisning

För godkänd redovisning krävs:

- Väl formaterad källkod
- Källkod ska gå att kompilera och exekvera med ett resultat som efterfrågas för uppgiften via kommandotolk
- Att uppgiften är löst på rimligt sätt oavsett om det är i kod eller annan form av dokumentation av lösning som efterfrågas
- Studenten ska muntligen kunna förklara sin lösning och beskriva varför lösningen ger det efterfrågade resultatet
- Studenten ska kunna visa förmåga att göra mindre ändringar i koden och kompilera och exekvera den ändrade koden med efterfrågat resultat via kommandotolk
- De lösningar som visas upp överensstämmer med de lösningar som lämnats in skriftligt.

Studenten ska alltid vara beredd på att kunna visa legitimation vid redovisning.

Redovisning i Zoom

Vid redovisning via Zoom så behöver du

- kunna kommunicera via mikrofon/högtalare
- ha kameran påslagen så att den som tar redovisningen kan se dig
- kunna uppvisa legitimation via kamera
- skärmdela den dator du sitter vid så att du kan visa dina lösningar, din kod och kompilering och exekvering

Har du inte möjlighet till detta vid en privat dator kontakta IT-supporten för studenter för att få hjälp med en lämplig plats för din redovisning.

Förberedelser

Inlämningsuppgift 1 omfattar kursinnehåll som gåtts igenom till och med föreläsning F11 och laboration L10. I Inlämningsuppgift 1 används flera olika byggstenar för problemlösning som tidigare används i laborationer och exempel. Du bör ha genomfört alla laborationerna L1-L10 innan du tar dig an Inlämningsuppgift 1 då dessa innehåller lösningsmetoder som du har nytta av till den här uppgiften.

Uppgift

Övergripande beskrivning

I uppgiften ska du implementera ett menystyrt program för kommandotolken. Programmet ska hantera en gästlista för ett event där man bjuder på mat (exempelvis ett bröllop eller någon annan fest). Då man bjuder på mat vill man utöver namn på gästerna hålla reda på ålder. Ålder behöver man veta för att kunna beräkna hur mycket mat som ska beställas. Man räknar alla gäster som är 13 år eller yngre som barn i portionsstorlek.

I programmet ska användaren kunna göra val i en meny där man kan välja saker som att lägga till, ta bort eller redigera gäster. Man ska också kunna se hur den aktuella gästlistan ser ut och få fram viss statistik för gästerna. Exakt funktionalitet presenteras i lista med krav på programmets funktionalitet nedan.

Gästlistan i ditt program behöver i det här fallet endast hantera 10 gäster för att testning ska bli rimlig. Programmet ska däremot vara skrivet så att man enkelt skulle kunna byta ut gästlistan mot en mycket större array. Det vill säga: när du behöver veta antalet element i gästlistan använd `.length` och inte en variabel som anger antalet element i listan.

Krav på funktionalitet

Det program du skapar ska uppfylla följande krav på funktionalitet i programmet:

Id för krav	Kravbeskrivning
IU1F1	Menyval: Användaren ska kunna skriva ut gästlistan med namn och ålder för gästerna.
IU1F2	Menyval: Användaren ska kunna skriva ut statistik för gästlistan. Statistiken ska visa: <ul style="list-style-type: none"> IU1F2.1: Det totala antalet gäster IU1F2.2: Antalet vuxna gäster IU1F2.3: Antalet gäster som är barn (max 13 år gamla gäster) IU1F2.4: Vem som är den äldsta gästen och dennes ålder IU1F2.5: Vem som är den yngsta gästen och dennes ålder
IU1F3	Menyval: Användaren ska kunna lägga till en ny gäst med namn och ålder till gästlistan om det finns en ledig plats.
IU1F3.1	Om ingen ledig plats finns för att lägga till en gäst ska användaren få ett meningsfullt felmeddelande som talar om detta.
IU1K4	Menyval: Användaren ska kunna ändra en gästs namn i gästlistan.
IU1F5	Menyval: Användaren ska kunna ändra en gästs ålder i gästlistan.
IU1F6	Menyval: Användaren ska kunna byta plats på två gäster i gästlistan. Detta ska ske oavsett om platserna man byter innehåller en annan gäst eller inte. I praktiken innebär detta att man kan flytta en gäst till en tom plats i listan.
IU1F7	Menyval: Användaren ska kunna ta bort en gäst ur gästlistan. Detta innebär att sätta tomma strängar istället för gästens namn och ålder.
IU1F8	Menyn ska skrivas ut i början av programmet och efter varje utfört val så att användaren vet vad hen kan välja att göra.
IU1F9	Menyval: Användaren ska kunna välja att avsluta programmet i menyn.

Krav på implementation

Följande krav ställs på din implementation av uppgiften:

Id för krav	Kravbeskrivning
IU1I1	Källkoden ska vara väl formaterad med lämplig indentering som presenterats på kursen.
IU1I2	Högst upp i filen Assignment1.java ska det finnas en kommentar som innehåller: <ul style="list-style-type: none"> Förnamn och efternamn Datorid Vilket program du läser
IU1I3	Källkoden ska gå att kompilera och exekvera.
IU1I4	Uppgiften ska vara löst på rimligt sätt utifrån de verktyg för problemlösning som presenterats på kursen till och med F11.
IU1I5	Lösningen ska följa den struktur med metoder som givits i filen Assignment1.java. Metoderna får ändras så att de har fler eller färre parametrar eller ger returvärden eller inte.
IU1I6	Gästlistan ska representeras av en array i koden och programmet ska skrivas så att man kan använda en array med olika antal rader utan att behöva ändra något annat än initieringen av arrayen.
IU1I7	Element i arrayen som inte innehåller information om någon gäst ska innehålla tomma strängar.
IU1I8	Koden ska delas upp i metoder där det som minst finns en metod för respektive menyval, undantaget att avsluta programmet. Koden får delas upp ytterligare i metoder om du finner det lämpligt.

Aktivitetsdiagram

Din lösning ska dokumenteras med ett aktivitetsdiagram som visar vad som sker när programmet körs. Du kan givetvis göra aktivitetsdiagrammet innan du börjar koda för att göra en design för din lösning. Säkerställ dock att diagrammet du lämnar in stämmer överens med den kod du lämnar in.

Följande krav ställs på aktivitetsdiagrammet:

Id för krav	Kravbeskrivning
IU1A1	Aktivitetsdiagrammet ska följa den UML-notation som presenterats på kursen.
IU1A2	Aktivitetsdiagrammet ska visa hur iterationen och selektionerna fungerar för den övergripande menyn.
IU1A3	Aktivitetsdiagrammet behöver inte visa vad som sker inuti metoder för respektive menyval (se undantag IU1A4) men ska visa när metoder för menyvalen anropas.
IU1A4	Alla aktiviteter som innebär någon form av inläsning av input/indata från användaren ska visas i aktivitetsdiagrammet, oavsett om denna finns i huvudmenyn eller i metoder för specifika menyval.
IU1A5	Aktivitetsdiagrammet ska vara ritat med något verktyg som Visual Paradigm eller liknande och får inte vara en handritad skiss på papper som sedan skannats in.

Tips och ledning

Här kan du hitta lösningar och förslag till hur man löser de olika delarna i inlämningsuppgift 1.

- Att läsa in värden som tal eller sträng från användaren hittar du i flera uppgifter i L5.
- Hitta största och minsta talet i en array se L9 och L10 Uppgift 2a och b.
- För att plocka ut ett sträng-värde och konvertera till tal se L10 filen `Test_parseInte.java` och Uppgift 3b
- 2-dimensionella arrayer och hur du traverserar dessa presenteras i föreläsning F9 och F10.
- Att strukturera data i form av strängar i en array se F10 och L10 Uppgift 3a-b och L11 Uppgift 3a-b.
- Hjälp hur metoder, parametrar och returvärden fungerar finns i föreläsning F8 och L8.
- Konstruera menyer hittar du i L4 Uppgift 7, L6 Uppgift 10, L8 Uppgift 1
- För att formatera strängar snyggare se Deitel 2.4, 14.3.6 och online kapitel I Formatted Output (läggs också som pdf på kursplatsen)