

Tentamen

Data- och informationsvetenskap: Objektorienterad programmering och modellering för IA (DA361A)

Teknik och samhälle, Malmö universitet

2018-12-11

Hjälpmedel

Penna och papper.

Anvisningar

Skriv bara **på ena sidan** av bladet och markera varje sida med ditt **personnummer samt sidnummer**. Skriv läsligt och kommentera utförligt vad du gör, det kan ge dig poäng även om resultatet är fel.

Glöm inte att fylla i försättsbladet, inklusive att kryssa i vilka uppgifter du svarat på. Tentamen kan ge totalt 30 poäng

För godkänt krävs 15 poäng, för väl godkänt krävs 23 poäng.

Jag kommer att komma förbi två gånger under tentamenspasset för att svara på eventuella frågor. Den första gången vid cirka 09.00 (så läs igenom alla frågor till dess så att du vet om du har några frågor), den andra vid cirka 10.30.

Viktigt, se bilaga för förtydligande vad som förväntas av er när det kommer till att göra klassdiagram i fråga 1 & 2.

Lycka till!

Fråga 1 (10p)

Ni har fått i uppgift att modellera ett system för ett bageri. Systemet ska hantera både den personal som jobbar på bageriet, befintliga kunder, men även allt som säljs där. Bageriet kommer att ha några heltidsanställda och till och från (under högsäsong) även ha timanställda. De timanställda står endast i kassan, medan de heltidsanställda, förutom att stå i kassan, även sköter bakningen av diverse godsaker. Bageriet har olika typer av godsaker, t.ex. bröd, tårter, kakor samt färdigbredda baguetter. Vid försäljning är det viktigt att tänka på att bröd och kakor säljes styckesvis, medan tårter kan köpas både hela och i bitar – det är viktigt för oss att hålla koll på. Vi vill vi även kunna se vem som bakat vilken godsak i bageriet. Slutligen ska kunderna kunna ha ange sina favoritgoodsaker, så systemet behöver även ha koll på detta.

- Skapa ett **use case-diagram** för att för systemet (2p).
- Skapa ett aktivitetsdiagram för hur en kund kan lägga till en (eller flera) godsaker som favoriter. (2p)
- Modellera systemet med ett **klassdiagram**. Tänk på att inkludera eventuella relationer som t.ex. arv, aggregation, komposition, samt kardinalitet mellan olika klasser. **Motivera** er modellering och klassdiagram och förklara, *där ni anser det förtydligar*, era designval. (6p)

Fråga 2 (10p)

Ni har fått i uppgift att modellera hisssystemet i Niagara. Niagara (byggnaden) har tre huskroppar, där varje huskropp har olika antal våningar. Varje huskropp har ett eget hissystem, bestående av två hissar. Varje hiss har sedan panel bestående av olika knappar med olika funktioner. Det finns knappar för val av våning, knappar för att öppna/stänga dörren och en knapp för hjälp (nödsituationer). För att hissarna ska verka på ett så effektivt sätt som möjligt så ska de även kunna kommunicera med varandra (inom sin huskropp), för att se vilken hiss som är närmst – och när hissen är ledig, skicka den till där användaren vill stiga på hissen. Ta även hänsyn till om hissarna är på väg upp eller ner (om ingen hiss är ledig just nu) för att förminska ev. väntetid för användaren.

- Modellera systemet med ett **klassdiagram**. Tänk på att inkludera eventuella relationer som t.ex. arv, aggregation, komposition, samt kardinalitet mellan olika klasser. **Motivera** er modellering och klassdiagram och förklara, *där ni anser det förtydligar*, era designval. (6p)
- Skapa ett **sekvensdiagram** för scenariot när användaren kallar på en hiss (klickar på knappen för att hen vill åka hissen), från det att knappen trycks in till att hissen anländer hos användaren. (2p)
- Skapa ett **sekvensdiagram** för scenariot när användaren valt våning hen vill åka till i hissen (genom att trycka på knappen för vald våning), från det att knappen trycks in tills att hissen är framme på korrekt våning och dörren öppnas. (2p)

Fråga 3 (10p)

Denna fråga inkluderar att ni skriver mindre kodstycken för att påvisa olika koncept genom programmeringsspråket Python (eller annat programmeringsspråk) i del a, b, c. Förklara er kod och exemplifiera gärna ev. klasser med exempelkod för att skapa instanser och hur dessa används där det är relevant. Utgå från något av de modellerade systemen i fråga 1 eller fråga 2 när ni svarar på frågorna nedan.

- Skriv Python-kod för ett exempel av **aggregation** som finns i fråga 1 eller fråga 2. (2,5p)
- Skriv Python-kod för ett exempel av **komposition** som finns i fråga 1 eller fråga 2. (2,5p)
- Skriv Python-kod för ett exempel av **arv** som finns i fråga 1 eller fråga 2. (2,5p)
- Kan ni hitta något exempel av **polymorfism** i fråga 1 eller fråga 2, om ja vilket? Om inte, ange ett konkret exempel som skulle innebära att polymorfism uppstår (i fråga 1 eller fråga 2)?
(Här behövs **ingen kod** skrivas) (2,5p)

Bilaga

Förtydligande gällande klassdiagram:

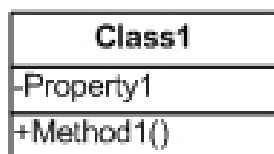
- Endast funktioner som påvisar systemets funktionalitet behövs skrivas ut (alltså inte getters, setters, etc.).
- Bara de viktigaste attributen som påvisar systemets struktur behövs skrivas ut (alltså inte detaljattribut som inte tillför något för att förstå systemet).
- Nedan är en tabell för några olika komponenter i klassdiagram som vi tittat på i kursen. Notera att t.ex. inte kardinalitet är med, vilket ni ska inkludera i era klassdiagram.

UML Cheatsheet

Interface



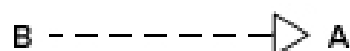
Class



These are both typically drawn vertically:



Inheritance



Generalization



Association



One way Association.



Aggregation



Composition