

Tentamen 2019-06-04
Objekt-orienterad programmering med Java,
1DL028

Uppsala Universitet
Institutionen för informationsteknologi
Avdelningen för datalogi

Kursansvarig: Sven-Olof Nyström

4 juni 2019

Läs igenom följande instruktioner noggrant:

- Vid tentans start skickar jag frågorna med email till alla som har anmält att de vill skriva. Du som använder flera emailadresser bör ha koll på vilken adress du ger mig när du anmäler dig till tentan.
- Den första dagen (i alla fall till klockan 15.00) finns jag tillgänglig för frågor via email. Jag kommer förstås att besvara email som vanligt även under resten av tentaperioden.
- Om jag skulle upptäcka någon viktig oklarhet i mina frågor kommer jag att försöka kontakta dig. I så fall skickar jag mail.
- Det är OK att söka information på nätet och i böcker. Det är *inte* OK att be andra människor om hjälp.
- Om bitar av ditt svar har plockats från någon källa, tala om det.
- Svar *måste* motiveras och förklaras. Detta är viktigt. Det blir mycket svårt för mig att ge poäng på svar utan motivering eller program som lika gärna kunde vara tagna från webben.
- I uppgifter som handlar om att skriva program ser jag helst att programmen är debuggade och körklara, men en halvfärdig lösning där tankegången är klar kan ge poäng.
- Några av frågorna handlar om ett program som du laddar ner här:
`http://user.it.uu.se/~svenolof/t-190604-cruund`. Kontrollera att du kommer åt programmet.
- Vid tentamensperiodens slut (eller när du är klar) laddar du upp svaren och programmen till Studentportalen.

Maxpoäng: 60.

Läs igenom hela tentan innan du börjar. Om du fastnar på en uppgift, gå vidare till nästa—uppgifterna är inte nödvändigtvis ordnade i svårighetsgrad. Om en uppgift består av flera deluppgifter och du fastnar på en av dem, gå vidare till nästa deluppgift—det kan hända att efterföljande deluppgifter kan lösas utan det fullständiga svaret på tidigare uppgifter.

1. Fel (6p)

Förklara skillnaden mellan kompileringsfel och fel vid körning.

Ge exempel.

Försök att ge allmänna principer för när ett fel uppträder vid kompilering och när det uppträder vid körning.

Kan du tänka dig någon typ av fel som skulle kunna visa sig i båda situationerna?

2. Referenssemantik (6p)

Se gärna anteckningarna, avsnitt 2.

(a) Förklara begreppet referenssemantik med egna ord.

(b) Beskriv en situation där ett objekt inte längre kan refereras och kommer att avallokeras.

Ge ett exempel på ett program där detta inträffar.

(c) Beskriv en situation där ett objekt refereras av två eller flera variabler.

Ge ett exempel på ett program där detta inträffar.

(d) Beskriv en situation där ett objekt kopieras.

Ge ett exempel på ett program där detta inträffar.

3. Autocomplete (12p)

Implementera ett program som tar emot början på ett svenskt ord och ger en lista av alla ord som börjar på detta sätt.

Exempel: Indata "avst" ger resultaten `avstanna`, `avstamp`, `avstava`, ...

Se anteckningarna, avsnitt 5 för exempel på ett program som läser från fil. Där finns även en svensk ordlista.

Budd's solitaire

Denna och de följande frågorna handlar om ett program som du kan ladda ner här:

<http://user.it.uu.se/~svenolof/t-190604-cruund>. Notera att en pdf-fil som beskriver programmet ingår.

Några av frågorna går ut på att modifiera programmet. Min tanke var att du skulle göra alla ändringar i samma program, men om du vill får du lämna in separata program för de olika frågorna. Berätta hur du gör.

4. Budd's solitaire i stora drag (7p)

Beskriv programmet med egna ord.

Beskriv klasserna och deras ansvar, och hur olika klasser är relaterade. Vilka klasser är beroende av vilka?

Påpeka bättre och sämre aspekter av programmet.

5. Städa upp (7p)

Nå, det finns några saker som bör städas upp och moderniseras.

- (a) Lägg varje klass i en egen fil.
- (b) Ser du andra saker du vill städa upp?
- (c) Vid kompilering får jag diverse varningar. Vilka är varningarna och vad beror de på?
- (d) Åtgärda varningarna.
- (e) Modernisera programmet. Ersätt användning av AWT med Swing.

Du ska bland annat ersätta `Frame` med `JFrame`, `Button` med `JButton` och metoden `paint` med `paintComponent`. Det sista för med sig andra ändringar eftersom `JFrame` inte är en komponent och metoden måste definieras för någon annan klass.

6. Statiska variabler (5p)

Det finns ett antal variabler som är deklarerade `static` i klassen `Solitaire`.

Vad representerar de?

Kan du tänka dig någon orsak till att Budd valde att göra dem statiska?

Skissa på hur en lösning *utan* statiska variabler skulle se ut. Finns det några fördelar eller nackdelar med en sådan lösning?

7. Eliminera statiska variabler (5p)

Implementera en lösning utan statiska variabler.

8. Duplicerad kod i select (6p)

Metoderna `select` i klasserna `DeckPile`, `TablePile` och `DiscardPile` innehåller en del likartad kod.

- (a) Beskriv den duplicerade koden och vad den representerar.
- (b) Hur skulle man kunna eliminera den duplicerade koden? Skissa på en lösning—eventuellt kan det vara lämpligt att introducera någon ny metod.
- (c) Modifiera programmet för att eliminera den duplicerade koden.

9. Roligare patiens (6p)

När man har sökt igenom högen med kort (talongen) en gång är spelet slut.

- (a) Vad måste ändras för att man ska kunna vända upp högen med kort och söka igenom den en gång till?
- (b) Implementera ändringarna.

LYCKA TILL!

Sven-Olof