Funktioner i det fullständiga systemet

Henrik Bergström Olle Karlsson

14 december 2018

Innehåll

1 Introduktion

Detta dokument beskriver kraven på de funktioner som ska finnas i den slutgiltiga versionen av hundregistret. Programmet är tänkt att användas av en kennel som säljer sina hundar på auktion och det finns därför kommandon för att hantera kunder och auktionerna. Alla de gamla kommandona från inlämningssteg fyra finns också kvar, även om ett par av dem har lite nya krav i denna version.

Versionshistorik

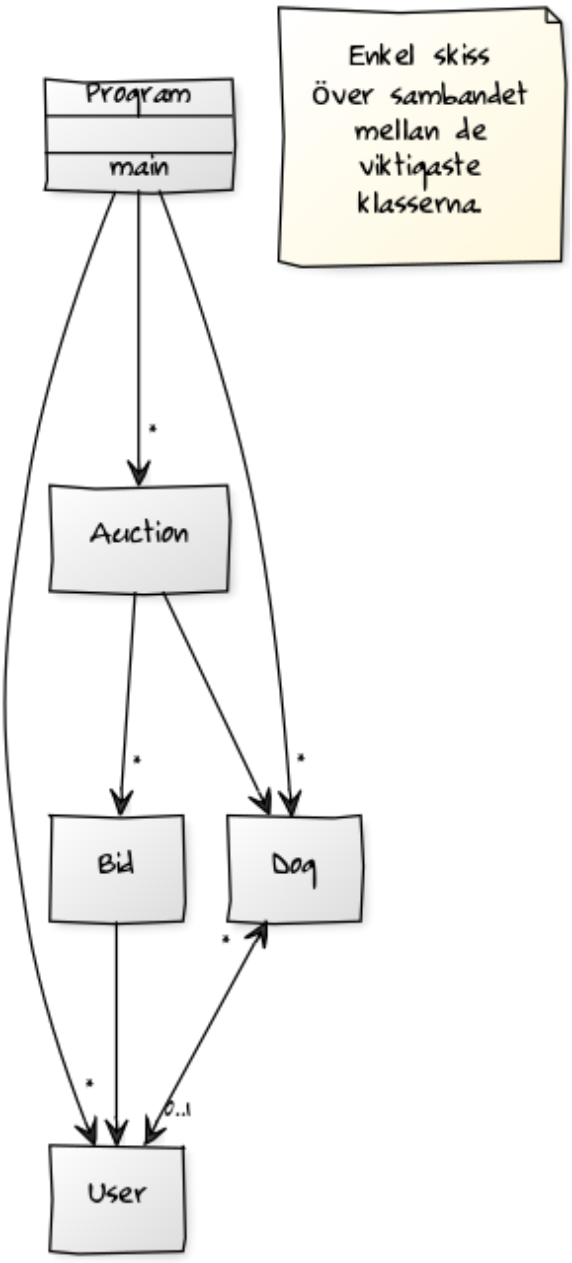
Denna uppgift är ny för i år, och det finns säkert oklarheter och kanske till och med rena felaktigheter i beskrivningarna av de olika kommandona. Om du hittar något du undrar över så skicka ett meddelande om det till handledningsforumet i ILearn. Nya versioner av dokumentet läggs ut löpande då vi uppmärksammas på saker och ting. Skulle det bli frågan om någon större revidering kommer detta att annonseras via kursforumet i ILearn.

2018-12-14 Det frivilliga kommandot `give dog` (sid. 26) lades till för att underlätta testning. En felaktig rubrik på ett exempel rättades.

2018-12-03 Första tentativa versionen.

2 Designskiss

Figur 2.1 på nästa sida visar en skiss över relationerna mellan de viktigaste klasserna i uppgiften. Denna skiss utgör inte något facit, men kan fungera som utgångspunkt för arbetet. Vi kommer att diskutera skissen på ett par av föreläsningarna. Där kommer vi också att titta på var funktionalitet bör ligga.



Figur 2.1: Designskiss

Del I

Generella krav

3 Namnformat

Programmet som ska skrivas hanterar namn på hundar, raser och användare. Tyvärr är det lätt hänt att användare skriver namn på olika sätt; Henrik en gång, henrik nästa, och med Caps Lock intryckt hENRIK.

Programmet ska acceptera samtliga former på namn och klara av att känna igen det även om det skrivs in på något annat sätt.

Exempel: Namn skrivna på olika format

```
Command> register new user
Name> peter
Peter added to register
Command> make bid
Enter the name of the user> pETER
Enter the name of the dog> rEx
Amount to bid (min 1)> 100
Bid made
```

Programmet ska även ta bort extra mellanslag i börja och slutet av namnen.

Exempel: Namn skrivet med extra mellanslag

```
Command>␣register␣new␣user
Name>␣␣␣␣␣␣␣␣␣␣␣␣␣␣Lotta
Lotta␣added␣to␣register
```

Om namnet är tomt, eller bara består av mellanslag ska användaren få en ny chans att skriva in namnet. Detta ska upprepas tills användaren skriver något.

Tips: det enklaste sättet att uppfylla kravet är att skriva en metod för inläsning av namn som alltid normaliserar namnet till det format du vill ha det på. Detta kan vara bara små bokstäver, bara stora, eller med lite mer jobb det normala skrivsättet med stor begynnelsebokstav.

Exempel: Inget namn angivet

```
Command> register new dog
Name>
Error: the name can't be empty
Name>
Error: the name can't be empty
Name> Ratata
Breed>
Error: the name can't be empty
Breed> No idea
Age> 5
Weight> 7
Ratata added to the register
```

4 De flesta fel återgår till menyn

Med undantag för tomma namn (sid. 5) och alltför låga bud (sid. 23) ska fel som användaren gör omedelbart avbryta den nuvarande funktionen och återgå till menyn. Användaren ska alltså inte få någon möjlighet att rätta till felet utan får ge kommandot igen.

Exempel: Efter de flesta fel återgår programmet till menyn

```
Command> make bid
Enter the name of the user> Slartibartfast
Error: no such user
Command>
```

Du behöver inte tänka på att användaren skriver helt fel typ av data. Testprogrammet kommer alltid att göra rätt på detta. Det finns alltså ingen risk att det, till exempel, kommer en textsträng där programmet förväntar sig ett tal. Om du ändå vill hantera denna typ av fel, något som tas upp på en av extraföreläsningarna, så får du gärna göra det. Du får då själv välja hur ditt program ska hantera felet.

Del II

Kommandon för att hantera hundar

5 Kommando: ”register new dog”

Detta kommando fanns med i programmet du lämnade in i inlämningssteg fyra, men har något förändrade krav. Det som har förändrats är kontrollen av namn och ras som ska göras i enlighet med kraven på sidan 5.

Detta kommando registrerar en ny hund. Användaren ska få frågor om namn, ras, ålder och vikt för hunden i denna ordning. Ett hundobjekt skapas sen och läggs in i registret.

Du kan anta att alla hundnamn är unika och att användaren inte gör något fel med siffrorna.

Exempel: register new dog

```
Command> register new dog
Name> Lassie
Breed> Collie
Age> 78
Weight> 25
Lassie added to the register
```

6 Kommando: ”increase age”

Detta kommando fanns med i programmet du lämnade in i inlämningssteg fyra, men har något förändrade krav. Det som har förändrats är kontrollen av namn som ska göras i enlighet med kraven på sidan 5.

Detta kommando ökar åldern på en hund med ett. Användaren ska få en fråga efter namnet på hunden som ska åldras. Hunden med det angivna namnet ska därefter bli ett år äldre.

Exempel: increase age för hund som finns

```
Command> increase age
Enter the name of the dog> Karo
Karo is now one year older
```

Om det inte finns någon hund med det angivna namnet så skall programmet skriva ut ett felmeddelande. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: increase age för hund som inte finns

```
Command> increase age
Enter the name of the dog> Ratata
Error: no such dog
```

7 Kommando: ”list dogs”

Detta kommando fanns med i programmet du lämnade in i inlämningssteg fyra, men har något förändrade krav. Det som har förändrats är att listan ska vara sorterad, ägaren ska finnas med vid utskriften, och att det ska komma ett felmeddelande om det inte finns några hundar registrerade.

Detta kommando listar hundarna i registret. Användaren ska få en fråga om minsta svanslängd och programmet ska skriva ut en lista på alla hundar hos kenneln som har samma eller längre svanslängd än denna minsta angivna. Om man anger 0 så kommer alltså alla hundar att skrivas ut. Anger man istället 10 så skrivs bara de hundar vars svanslängd är större eller lika med 10 ut.

Vid utskriften skall alla hundens attribut samt svanslängden skrivas ut. Detta inkluderar i denna version av programmet en eventuell ägare. Listan ska vara sorterad efter svanslängd. Om två hundar har samma svanslängd ska de vara sorterade efter namn.

Exempel: list dogs

```
Command> list dogs
Smallest tail length to display> 1
The following dogs has such a large tail:
* Fido (Dachshund, 1 years, 2 kilo, 3,70 cm tail)
* Karo (Bulldog, 8 years, 7 kilo, 5,60 cm tail, owned by Henrik)
* Milou (Terrier, 7 years, 8 kilo, 5,60 cm tail)
```

Om det inte finns några hundar registrerade ska det inte komma någon fråga om minsta svanslängd. Istället ska det skrivas ut ett felmeddelande. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: list dogs

```
Command> list dogs
Error: no dogs in register
```

8 Kommando: ”remove dog”

Detta kommando fanns med i programmet du lämnade in i inlämningssteg fyra, men har något förändrade krav. Det som har förändrats är kontrollen av namn som ska göras i enlighet med kraven på sidan 5, samt det inte räcker med att bara ta bort hunden ur registret. Den måste också tas bort från sin ägare och från en eventuell auktion.

Detta kommando tar bort en hund ur registret. Användaren ska få en fråga om namnet på den hund som skall tas bort varefter hunden med det angivna namnet tas bort. I den tidigare versionen av programmet räckte det med att ta bort hunden ur `ArrayList`en, men nu måste hunden också tas bort från en eventuell ägare, och från auktionerna om den är till salu.

Du behöver inte hantera komplikationen att det kan finnas flera hundar med samma namn.

Exempel: remove dog när hunden finns

```
Command> remove dog
Enter the name of the dog> Karo
Karo is removed from the register
```

Om det inte finns någon hund med det angivna namnet så skall programmet skriva ut ett felmeddelande. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: remove dog när hunden inte finns

```
Command> remove dog
Enter the name of the dog> Ratata
Error: no such dog
```

Del III

Kommandon för att hantera användare

9 Kommando: ”register new user”

Detta kommando registrerar en ny användare. Användare är ett nytt koncept i denna version av programmet. För tillfäller representerar användarna kunder, men det finns tankar på att i framtiden utveckla systemet så att det stödjer andra typer av verksamheter så som sociala media för hundar, hundtävlingar och hunddagis. Man har därför valt att använda det mer generella ordet användare (user) istället för kund.

När användaren (av programmet) vill registrera en ny användare (kund) så ska denne få en fråga om namnet på den nya användaren som ska skapas. Därefter skapas en användare och läggs in i systemet.

Precis som för hundar kan du anta att alla användare har unika namn.

Exempel: register new user

```
Command> register new user
Name> Henrik
Henrik added to the register
```

10 Kommando: ”list users”

Detta kommando listar samtliga användare i registret. Vid utskriften ska namnet på användaren finnas med tillsammans med namnet på alla hundar denne äger.

Exempel: list users

```
Command> list users
Henrik [Karo, Rex]
Olle []
```

Om det inte finns några användare registrerade ska det istället skrivas ut ett felmeddelande. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: list users när det inte finns några användare

```
Command> list users
Error: no users in register
```

11 Kommando: ”remove user”

Detta kommando tar bort en användare ur registret. Användaren (av programmet) ska få en fråga om namnet på användaren som skall tas bort varefter användaren med det angivna namnet tas bort.

När en användare tas bort ska användarens hundar tas bort samt eventuella bud användaren har lagt i alla auktioner.

Du behöver inte tänka på komplikationen att det kan finnas flera användare med samma namn.

Exempel: remove user när användaren finns

```
Command> remove user
Enter the name of the user> Henrik
Henrik is removed from the register
```

Om det inte finns någon användare med det angivna namnet så skall programmet skriva ut ett felmeddelande. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: remove user när användaren inte finns

```
Command> remove user
Enter the name of the user> Henrik
Error: no such user
```

Del IV

Kommandon för att hantera auktioner

12 Kommando: ”start auction”

Detta kommando startar en ny auktion för att sälja en hund. Användaren ska få frågan om vilken hund som ska säljas varefter auktionen skapas. När en auktion skapas ska den automatiskt få ett löpnummer. Den första auktionen som skapas ska få nummer 1, nästa nummer 2, osv.

Exempel: start auction

```
Command> start auction
Enter the name of the dog> Fido
Fido has been put up for auction in auction #5
```

Om hunden som ska säljas inte finns ska ett felmeddelande ges och ingen auktion skapas. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: start auction när hunden inte finns

```
Command> start auction
Enter the name of the dog> Karo
Error: no such dog
```

Om hunden som ska säljas redan finns utlagd för försäljning ska ett felmeddelande ges och ingen auktion skapas. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: start auction när hunden redan är utlagd till försäljning

```
Command> start auction
Enter the name of the dog> Fido
Error: this dog is already up for auction
```

Om hunden som ska säljas redan har en ägare ska ett felmeddelande ges och ingen auktion skapas. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: start auction när hunden redan har en ägare

```
Command> start auction
Enter the name of the dog> Fido
Error: this dog already has an owner
```

I exemplen ovan ges tre olika felmeddelanden för de olika feltillstånden. Detta är bra ur användarsynvinkel, men inte ett krav. Det kan vara samma felmeddelande i alla tre fallen.

13 Kommando: ”list auctions”

Detta kommando listar samtliga pågående auktioner. Vid utskriften ska löpnumret på auktionen finnas med, namnet på hunden som ska säljas, samt de tre högsta buden. Buden ska vara sorterade med det högsta budet först.

Tänk på att det inte behöver finnas tre bud registrerade än.

Exempel: list auctions

```
Command> list auctions
Auction #1: Milou. Top bids: [Anna 200 kr, Henrik 100 kr]
Auction #2: Rex. Top bids: []
```

Om det inte pågår några auktioner ska istället ett felmeddelande ges. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: list auctions när det inte finns några pågående auktioner

```
Command> list auctions
Error: no auctions in progress
```

14 Kommando: ”list bids”

Detta kommando ska lista samtliga bud i en viss auktion. Vid utskriften ska buden vara sorterade med det högsta budet först. Listan ska innehålla namnet på den som gav budet och beloppet.

Exempel: list bids

```
Command> list bids
Enter the name of the dog> Milou
Here are the bids for this auction
Anna 200 kr
Henrik 100 kr
```

Om inga bud angetts än ska detta skrivas ut. Observera att detta inte är ett fel, och alltså inte ska betraktas som ett felmeddelande.

Exempel: list bids när inga bud finns

```
Command> list bids
Enter the name of the dog> Rex
No bids registred yet for this auction
```

Om hunden som anges inte ligger ute till försäljning ska däremot ett felmeddelande ges. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: list bids när hunden inte är till försäljning

```
Command> list bids
Enter the name of the dog> Karo
Error: this dog is not up for auction
```

15 Kommando: ”make bid”

Detta kommando registrerar ett nytt bud på en hund. Användaren ska få frågor om vem som vill lägga budet, vilken hund som ska bjudas på, och hur mycket som bjuds i denna ordning.

Exempel: make bid

```
Command> make bid
Enter the name of the user> Henrik
Enter the name of the dog> Milou
Amount to bid (min 201)> 300
Bid made
```

Om användaren redan avgett ett bud ersätter det nya budet det gamla. En användare kan alltså bara ha ett bud per auktion.

Exempel: make bid uppdaterat bud

```
Command> make bid
Enter the name of the user> Henrik
Enter the name of the dog> Rex
Amount to bid (min 1)> 100
Bid made
Command> make bid
Enter the name of the user> Henrik
Enter the name of the dog> Rex
Amount to bid (min 101)> 200
Bid made
Command> list bids
Enter the name of the dog> Rex
Here are the bids for this auction
Henrik 200 kr
```

Om personen som avger budet inte existerar ska ett felmeddelande ges. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: make bid när användaren inte finns

```
Command> make bid
Enter the name of the user> Rolf
Error: no such user
```

Om hunden inte är till salu ska ett felmeddelande ges. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: make bid hunden inte är till salu

```
Enter the name of the user> Henrik
Enter the name of the dog> Ratata
Error: this dog is not up for auction
```

Om budet är för lågt, alltså lägre eller lika med det nuvarande högsta budet, ska ett felmeddelande ges. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande. Till skillnad från de flesta övriga felmeddelandena ska detta inte avbryta inmatningen utan användaren ska få en ny chans att lägga ett bud. Detta ska fortsätta tills användaren lägger ett godkänt bud.

Det lägsta möjliga budet om det inte tidigare finns något bud är en krona.

Exempel: make bid när budet är för lågt

```
Command> make bid
Enter the name of the user> Henrik
Enter the name of the dog> Milou
Amount to bid (min 201)> 100
Error: too low bid!
Amount to bid (min 201)> 200
Error: too low bid!
Amount to bid (min 201)> 300
Bid made
```

16 Kommando: ”close auction”

Detta kommando stänger en pågående auktion. Användaren ska få frågan om vilken hunds auktion som ska stängas och därefter ska hundens ägare sättas till den som gav det högsta budet. Utksriften från programmet ska innehålla namet på den som vann auktionen och beloppet.

Exempel: close auction

```
Command> close auction
Enter the name of the dog> Milou
The auction is closed. The winning bid was 200kr and was made
by Anna
```

Om inget bud avgetts ska auktionen ändå stängas, men hunden får ingen ny ägare. Detta räknas inte som ett fel.

Exempel: close auction när inget bud lagst

```
Command> close auction
Enter the name of the dog> Rex
The auction is closed. No bids where made for Rex
```

Om hunden inte är till salu eller inte finns ska ett felmeddelande ges. Var noga med att felmeddelandet innehåller något av orden *fel* eller *error*, annars kan testprogrammet inte känna igen att det rör sig om ett felmeddelande.

Exempel: close auction när hunden inte är till salu eller inte finns

```
Command> close auction
Enter the name of the dog> Fido
Error: this dog is not up for auction
```

Del V

Övriga kommandon

17 Kommando: give dog

Detta kommando är inget krav, och testprogrammet kommer inte att köra det, men det kommer att underlätta din egen testning. Mycket av koden kommer att vara identisk med koden för `close auction`*, så metoder kommer att vara till hjälp.*

Användaren får frågan om vilken hund som ska ges bort och vilken användare som ska få den.

18 Kommando: ”exit”

Detta kommando är detsamma som det var i programmet du lämnade in i inlämningssteg fyra. Det finns med här för att göra kommandolistan komplett.

Detta kommando skriver ut ett meddelande om att programmet avslutas och sen avslutas det. Detta ska göras genom att kommandoloopen avslutas, och inte med `System.exit` som inte får användas.

Exempel: exit

```
Command> exit
Goodbye!
```