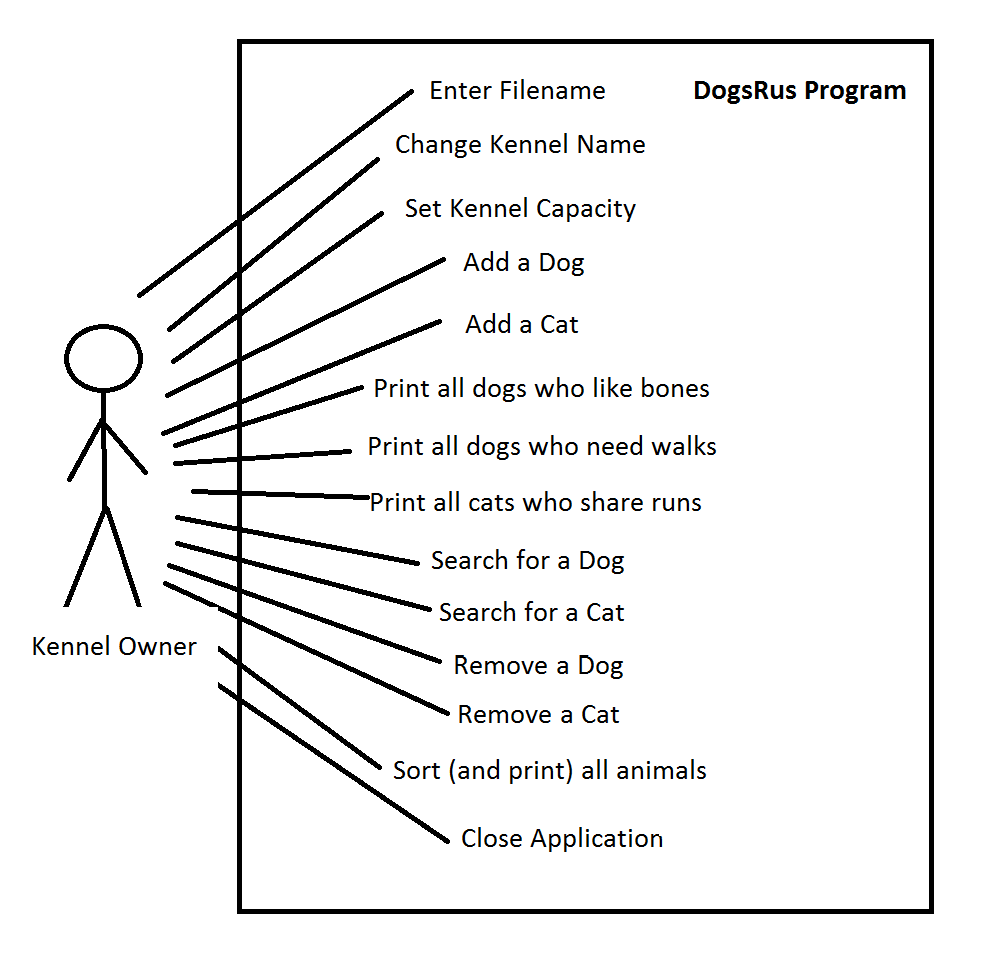
Shankly Richard Cragg

[shc27@aber.ac.uk](mailto:SHC27@ABER.AC.UK)

19/04/2015

DogsRUs

# Use Case Diagram



# Class Diagram

1..1

1..1

1..1

1..\*

1..\*

# Write-up

I began by creating an Animal superclass to be used for dogs and cats. I made this an abstract class as I never intent to create an Animal. I first moved all the methods from dog into this Animal class. I made these methods final as I did not want them edited in the subclasses. I changed the toString to use a stringbuilder as per the request of the pdf. I gave it 500 characters of space, as it in practice was large enough to not have any errors.

The dog class now only contains getters for likesBones and needWalks as there is nothing that edits these values after they have been initialised. The dog class extends Animal as it inherits things used in its constructor. The Cat class is very similar to the dog class.

Using a stringBuilder for my toStrings was very easy to implement, I found it similar to using string concatenation, but it’s a lot more pleasing to work with in its presentation.

The most difficult thing in Kennel was the sorting all animals by name. I eventually took the internet, and followed a tutorial on how to compare value of objects in an arrayList. I had to implement “Comparable” to Animal. From here I adapted some code from the tutorial to compare 2 strings, and called this method from Kennel via “Collections.sort”. As a side effect of implementing “Comparable” to Animal I had to include 2 blank methods in Cat and Dog to compile the code.

In total there are 13 menu options (including quit). I could have theoretically cut this down to many fewer, by compressing “add dog” and “add cat” down to add animal, and inside that asking which animal too add, but I decided that having them all accessible on the menu made it quicker to do what you wanted (which was helpful for running repeated tests). It does make it slightly more work to add extra animal types who want the same functions.

I added an “ID” as a solution to the problem of searching for an animal when two animals had the same name. By giving each Animal a unique ID, it makes it really easy to search for, and remove the intended animal from the arrayList. To give each animal a unique ID, I make the “nextFreeID” equal to the largest ID read in during the “initialise” method plus one.

I used a custom package to show I know how to implement one, it does not serve much purpose other than to demonstrate the implementation of one in my project. I found this relatively easy to do. I simply replaced the default package with my own simply name “CustomPackage”.

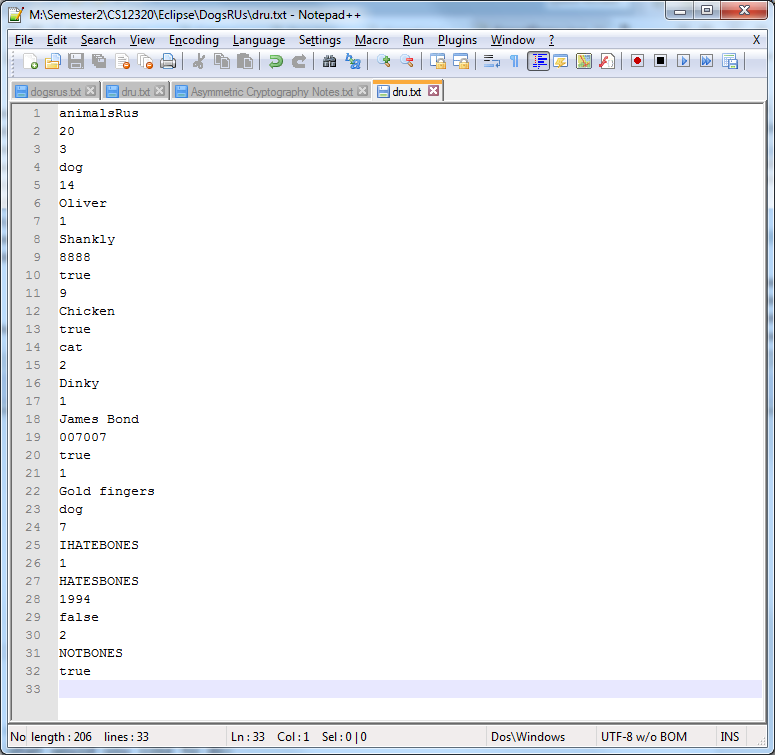
To ensure good error checking, and correct input from the user, I created a few functions called “getStringOfChars”, “getInt” and “getBool”, as these were called multiple times in the code. I included regular expression when I wanted to enforce a specific input (Such as Y, N, yes or no for weather to do something again, like getting another owner).

I believe I deserve a first for achieving all the goals set in the task in a way which has thorough error checking and makes sense given the situation.

# Test table

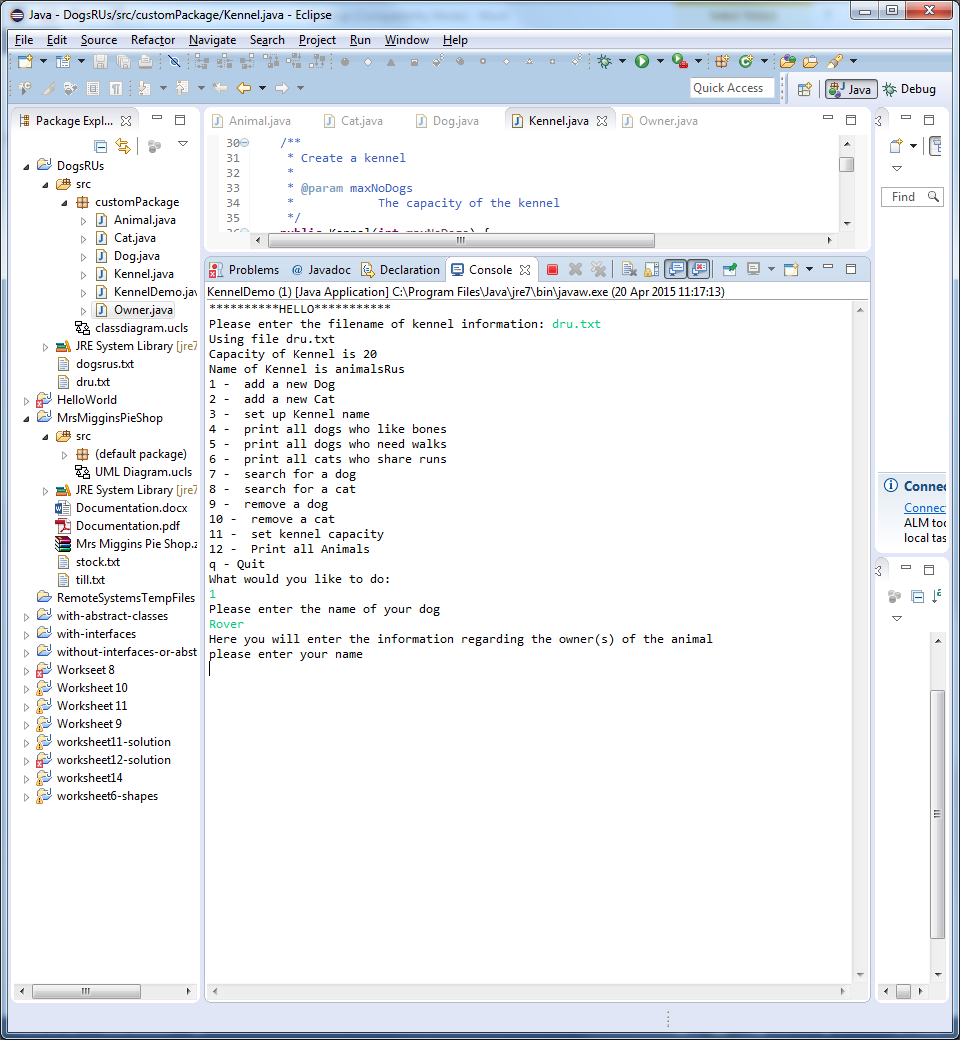
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Menu Option Selected** | **Description** | **Inputs** | **Expected outputs** | **Pass/**  **Fail** | **Comments** |
| 1.1 | 1/2 | Test animal name input | Rover | It is accepted | Pass | I only accept letters of the alphabet as input, so numbers may not be in the name of a dog. |
| Rover234 | It is not accepted | Pass |
| 1.2 | 1/2 | Test animal owner name input | Shankly | It is accepted | Pass | I do not recognise the “ “ space as an accepted input for strings, therefore only a forename is accepted.  All info requiring a string follow the same rules that both this and 1.1 follow, any further tests for string I will refer to these ones |
| Shankly Cragg | It is not accepted | Pass |
| 1.3 | 1/2 | Test animal owner phone number input | 1234 | It is accepted | Pass | I only accept 4 digits in a row as input. |
| 123 | It is not accepted | Pass |
| abcd | It is not accepted | Pass |
| 1.4 | 1/2 | Test add another owner prompt (Y/N) | y | It is accepted | Pass | I accept either “y”, “n”, “yes” or “no” for this selection. Capitalisation does not matter.  All yes or no checks follow this exact same procedure, and when requiring screenshots I will refer to these ones |
| n | It is accepted | Pass |
| Yes | It is accepted | Pass |
| No | It is accepted | Pass |
| gibberish | It is not accepted | Pass |
| 1.5 | 1/2 | Test dog like bones / cat share runs (Y/N) | Mirrored from 1.4 | Mirrored from 1.4 | Pass | For dogs liking bones and cats sharing runs takes place at the same place sequentially. This input/output is mirrored from 1.4 |
| 1.6 | 1/2 | Test animal favourite food | Mirrored from 1.1/1.2 | Mirrored from 1.1/1.2 | Pass | String input follow same guidelines as 1.1/1.2 |
| 1.7 | 1/2 | Test animal number of feeds | 3 | It is accepted | Pass | Only accepts an int larger than 0 |
| 0 | It is not accepted | Pass |
| -3 | It is not accepted | Pass |
| 1.8 | 1/2 | Test ID given to Animal | N/A | It is unique | Pass | When comparing the ID given to those of Animals already in the kennel, we see it is unique. |
| 3.1 | 3 | Test change Kennel name | Mirrored from 1.1/1.2 | Mirrored from 1.1/1.2 | Pass | String input follows same guidelines as 1.1/1.2 User is taken back to main menu and new Kennel name is printed. |
| 4.1 | 4 | Test that all dogs who like bones are printed | N/A | Only dogs who like bones are printed | Pass | The only dog who likes bones is printed |
| 5.1 | 5 | Test that all dogs who need daily walks are printed | N/A | Only dogs who need walks are printed | Pass | Both dogs who need walks are printed |
| 6.1 | 6 | Test that all cats who share runs are printed | N/A | Only cats who share a run are printed | Pass | All cats in Kennel who share runs are printed |
| 7.1 | 7 | Test search dog | 14 | Finds and prints information on dog with ID 14 | Pass | Only the dog with ID 14 is printed, the cat with ID of 2 isn’t printed when searching for a dog |
| 2 | No dog is found | Pass |
| 8.1 | 8 | Test search Cat | 2 | Finds and prints information on cat with ID 2 | Pass | Only the cat with ID 2 is printed, the dog with ID 14 isn’t printed when searching for a cat |
| 14 | No cat is found | Pass |
| 9.1 | 9 | Test remove dog | 14 | Removes dog with ID 14 | Pass | Only the dog with ID 14 is removed, the cat with ID 2 isn’t removed when removing a dog |
| 2 | No dog is found to be removed | Pass |
| 10.1 | 10 | Test remove Cat | 2 | Removes cat with ID 2 | Pass | Only the cat with ID 2 is removed, the dog with ID 14 isn’t removed when removing a cat |
| 14 | No cat is found to be removed | Pass |
| 11.1 | 11 | Test change Kennel size | 2 | Not aloud as too small size | Pass | Kennel size can only be changed to an integer equal too or larger than the number of animals currently in the Kennel. |
| 10 | Kennel size changed to 10 | Pass |
| 12.1 | 12 | Sort by name and print all animals | N/A | All Animals printed in alphabetical order | Pass | Animals are printed in alphabetical order |

This is the file I used for all of my tests

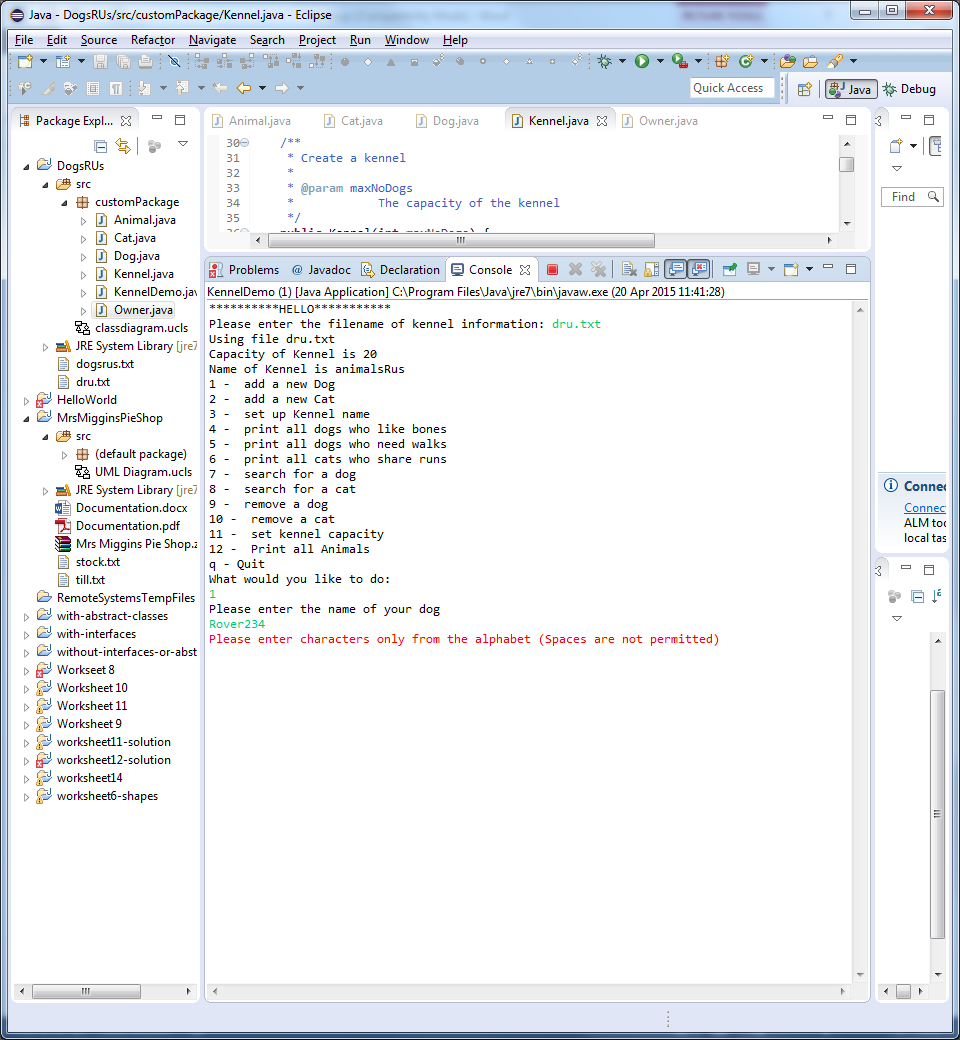


## ID 1.1

Rover being accepted

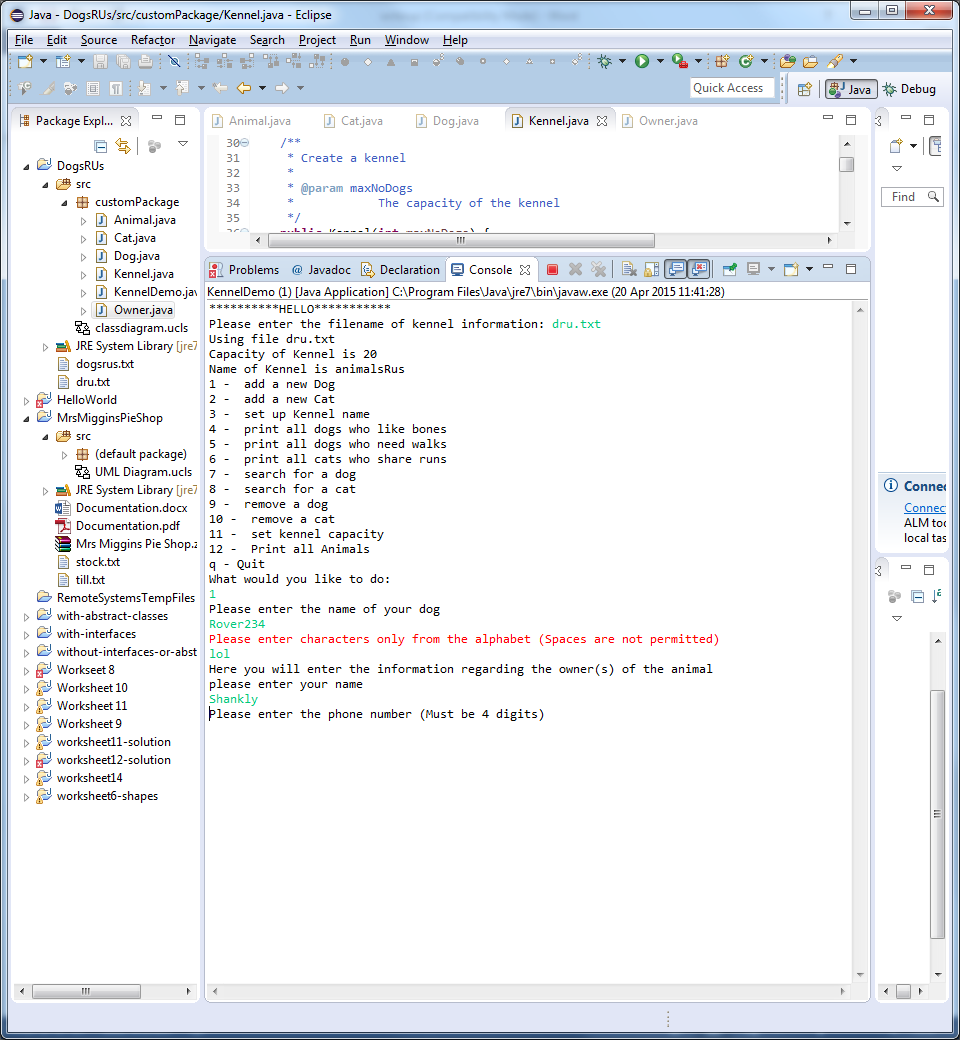


Rover 234 being denied

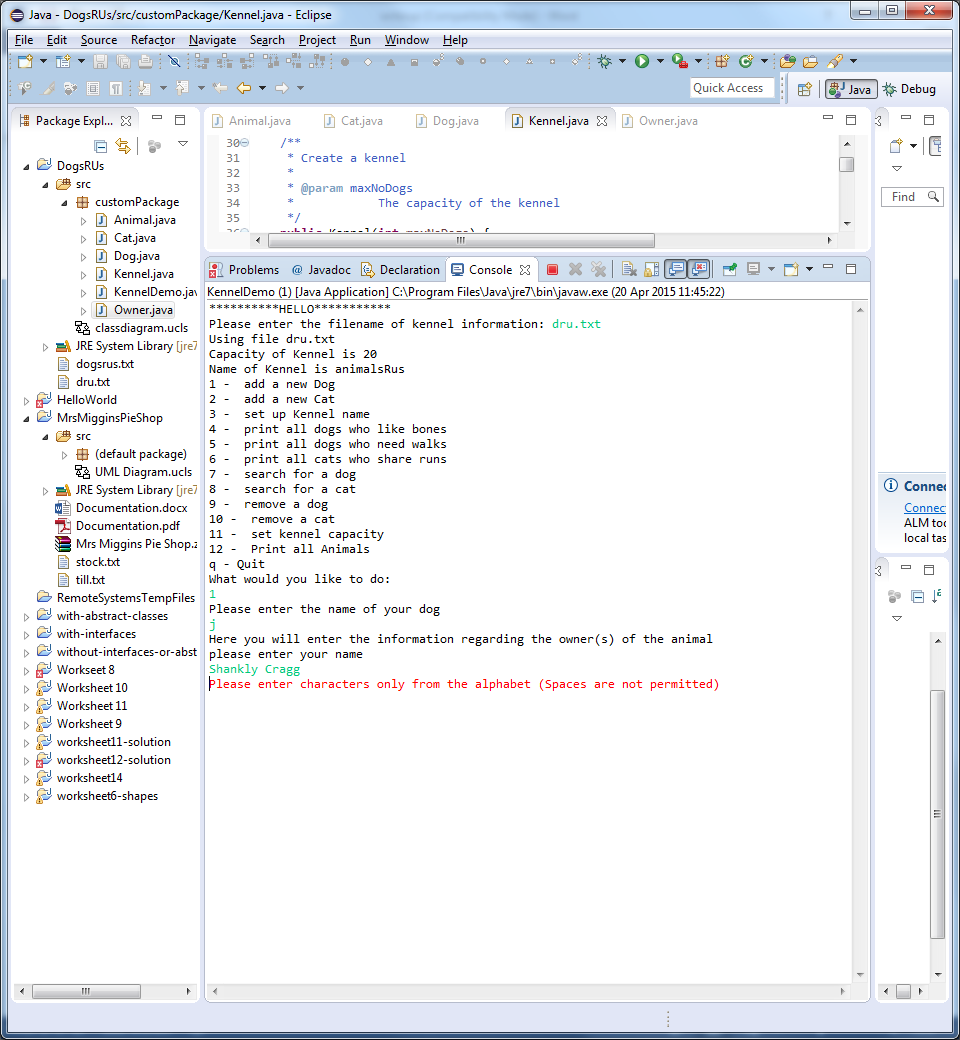


## ID 1.2

Shankly being accepted

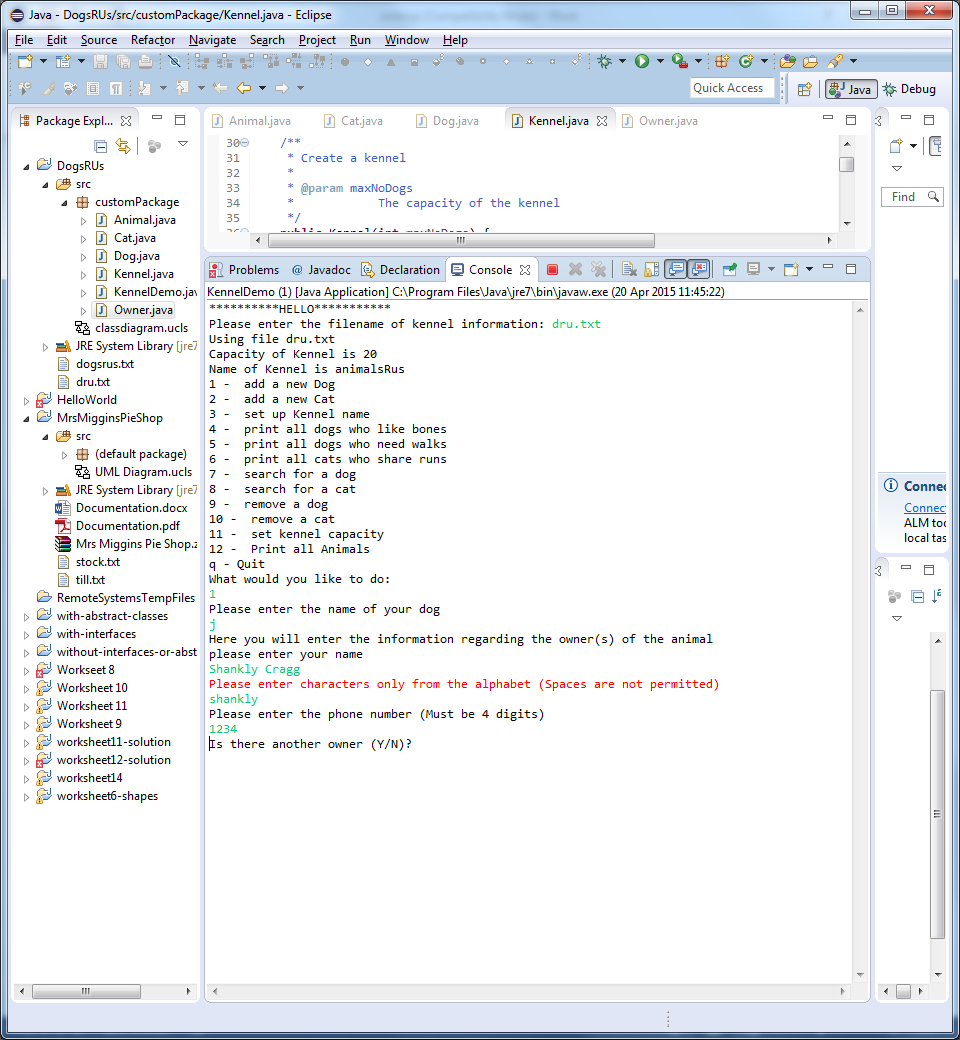


Shankly Cragg being denied

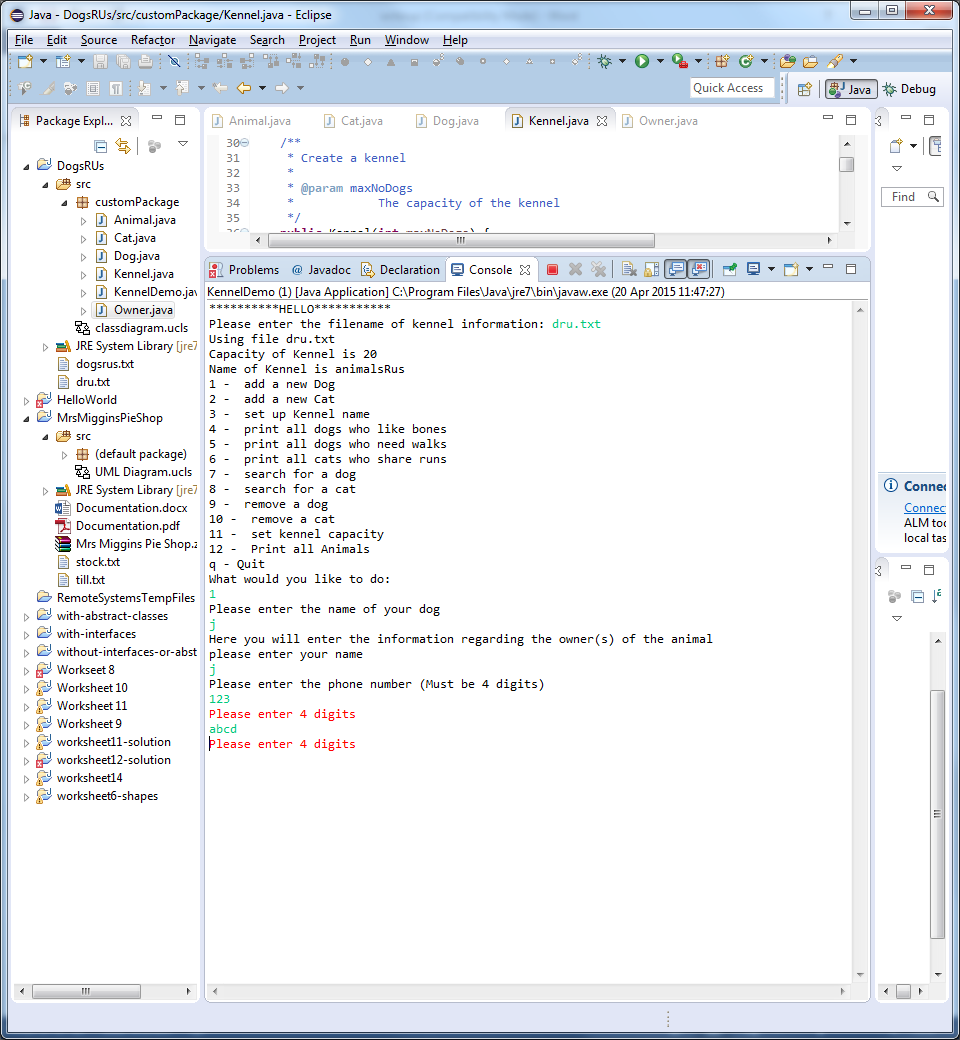


## ID 1.3

1234 being accepted

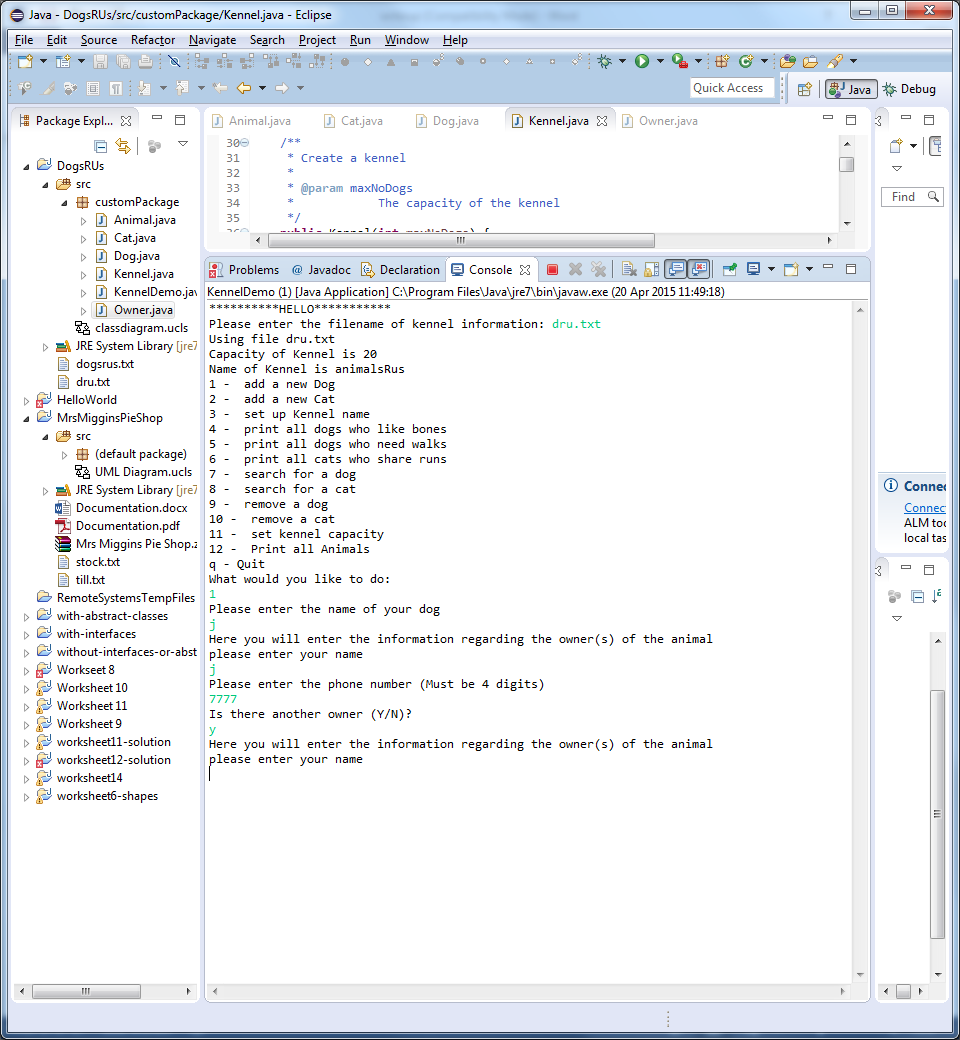


123 and abcd being denied

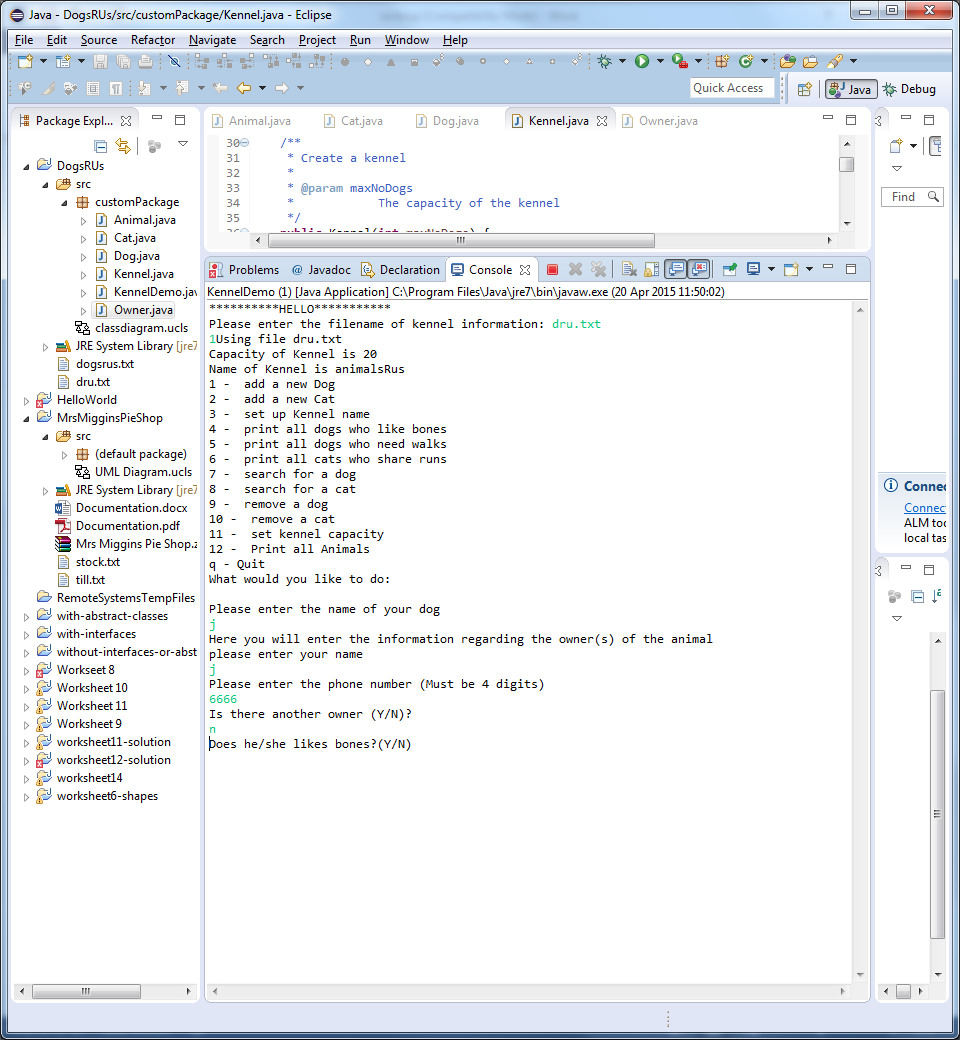


## ID 1.4

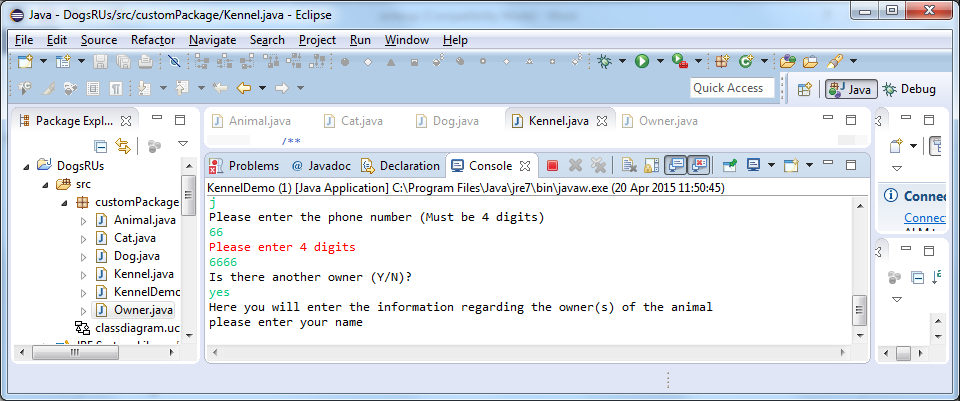
Y being accepted



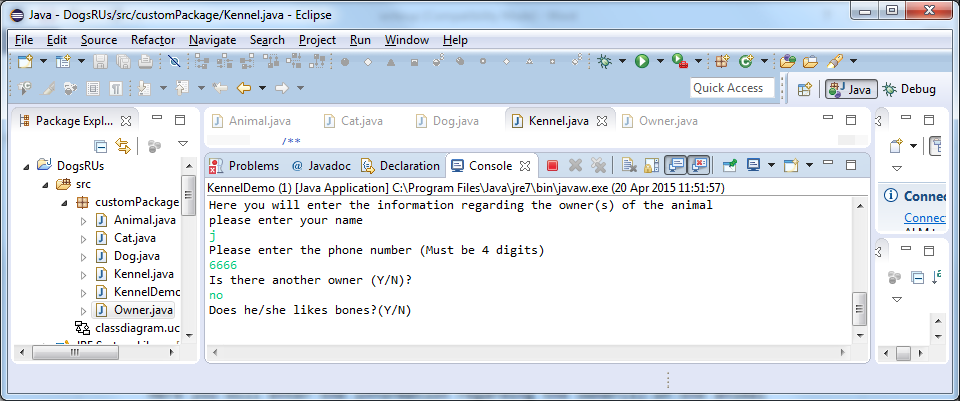
N being accepted



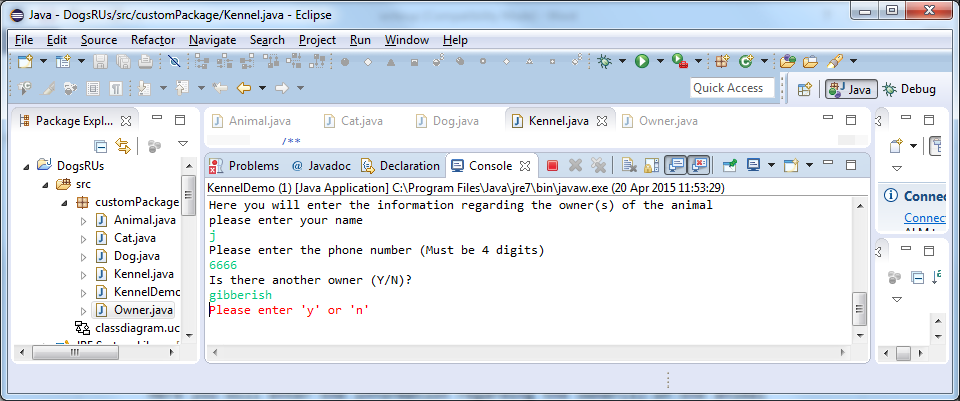
Yes being accepted



No being accepted

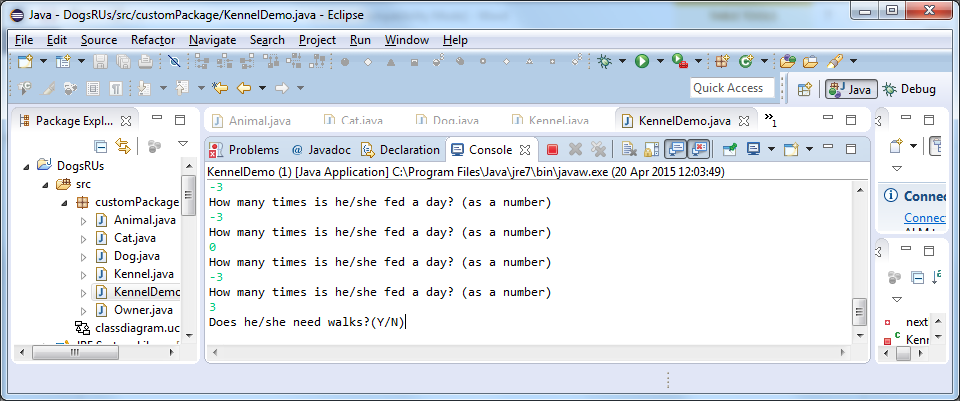


Gibberish being denied

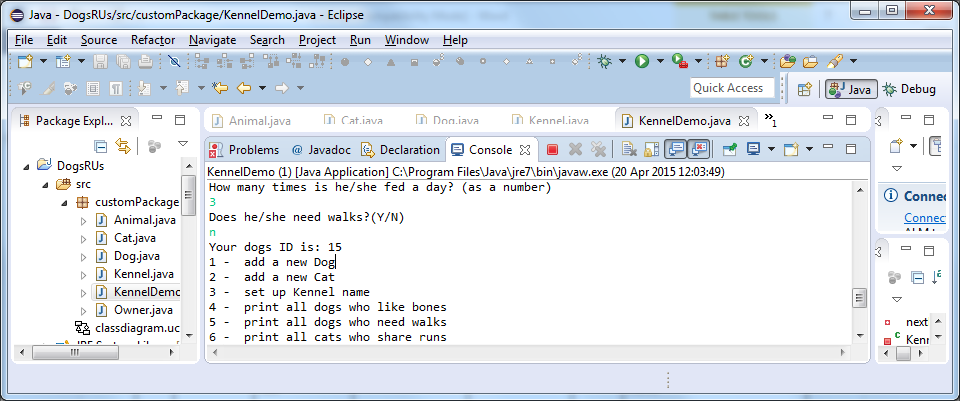


## ID 1.7

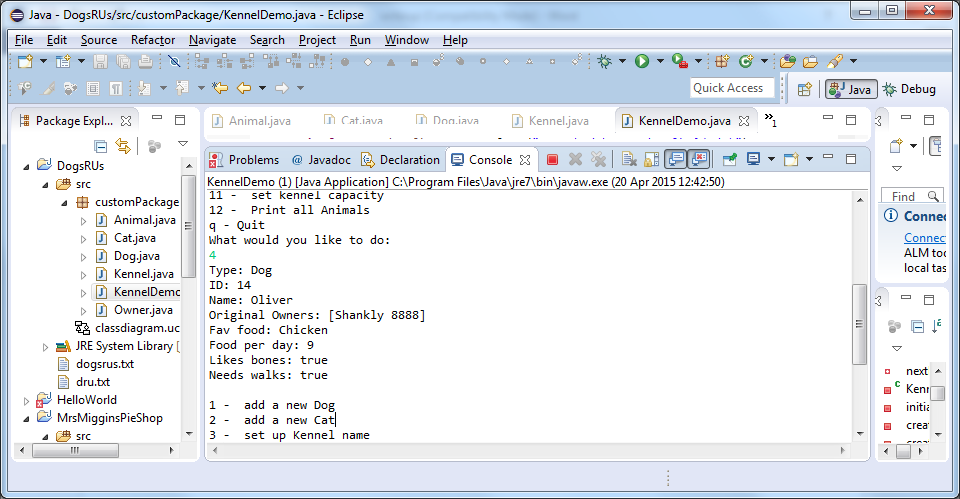
0 and -3 not being accepted. Then 3 being accepted.



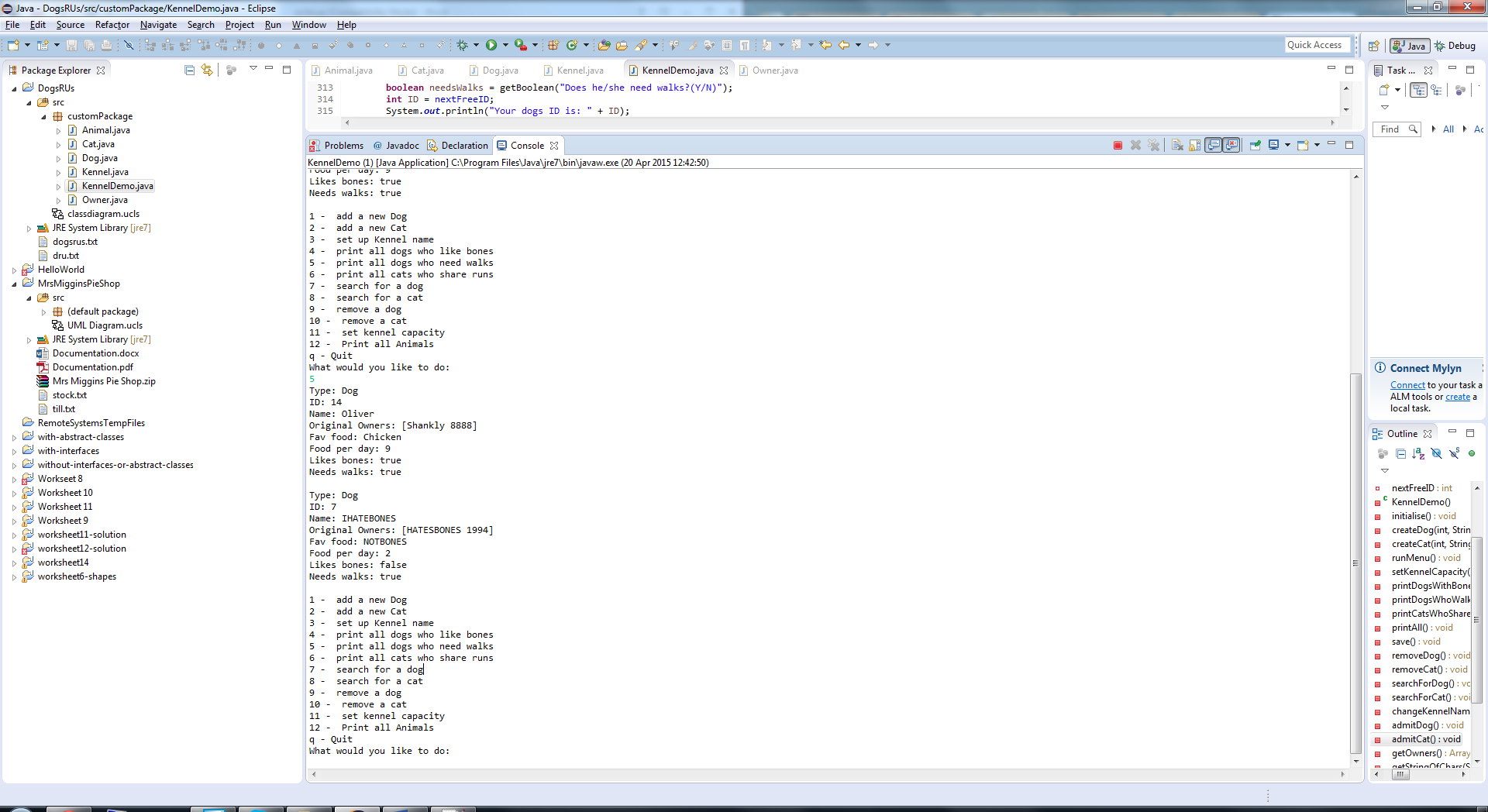
## ID 1.8



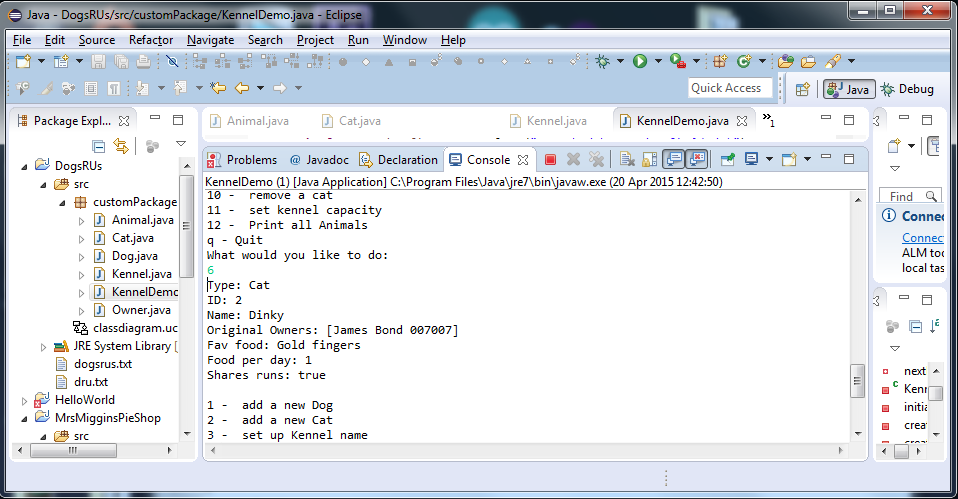
## ID 4.1



## ID 5.1

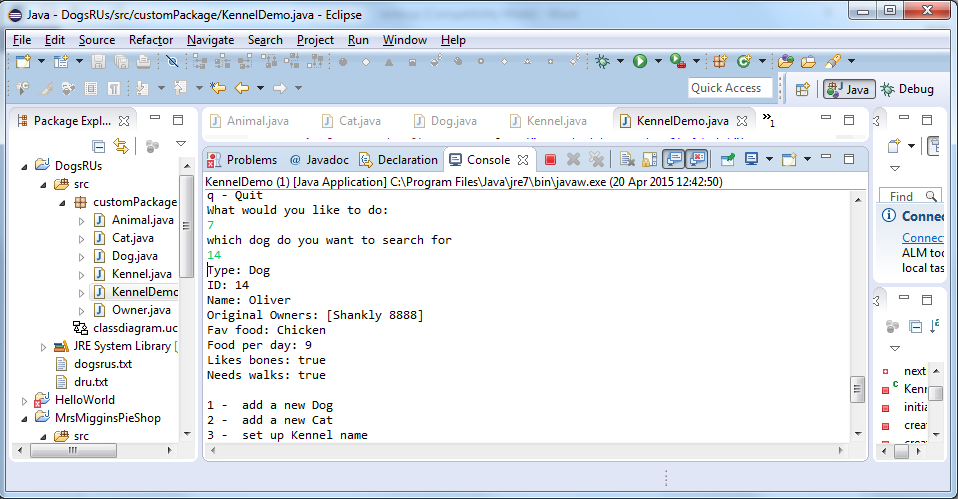


## ID 6.1

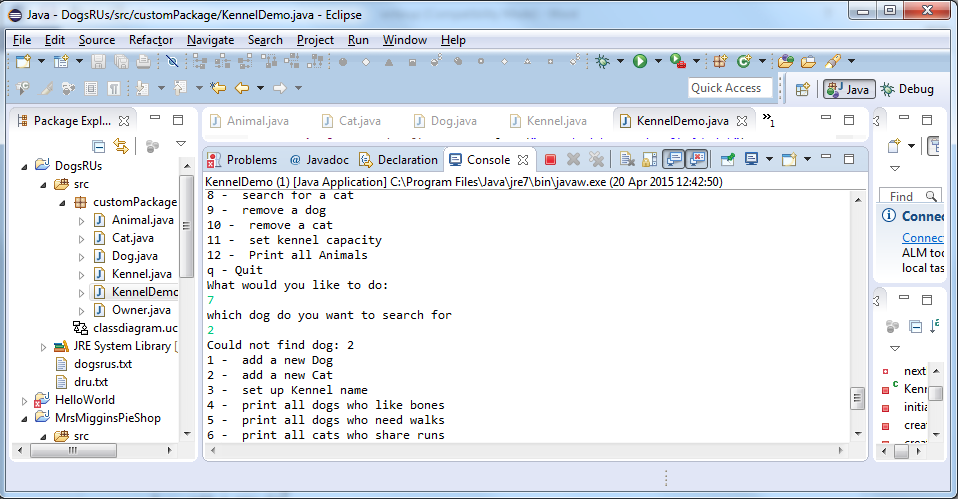


## ID 7.1

Dog ID 14 being searched for

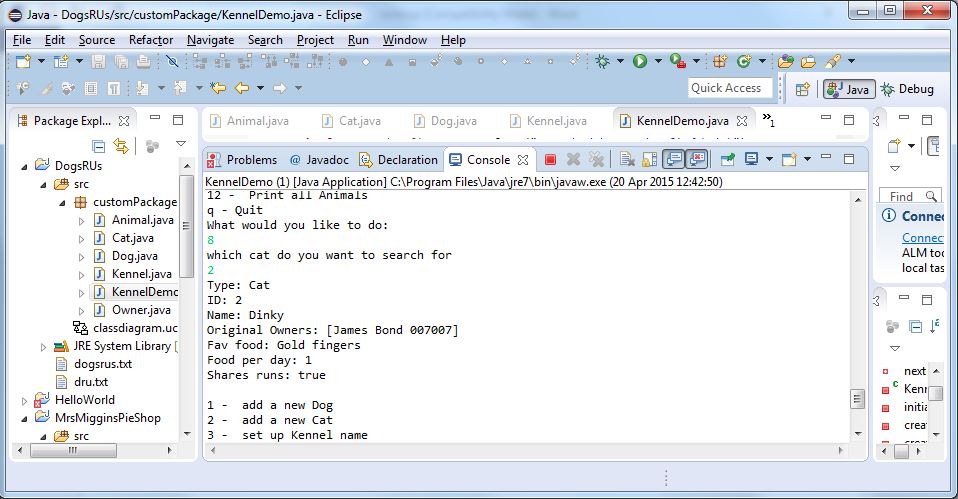


Dog ID 2 being searched for

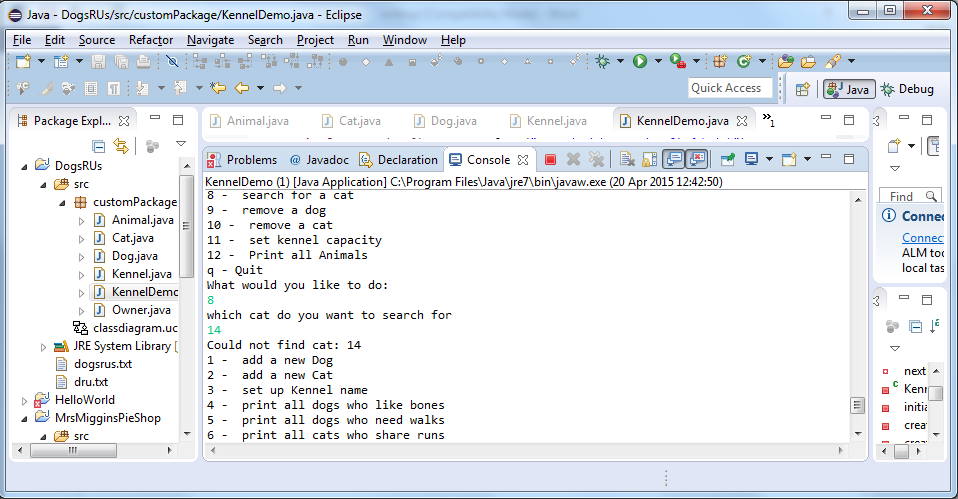


## ID 8.1

Cat with ID 2 being searched for

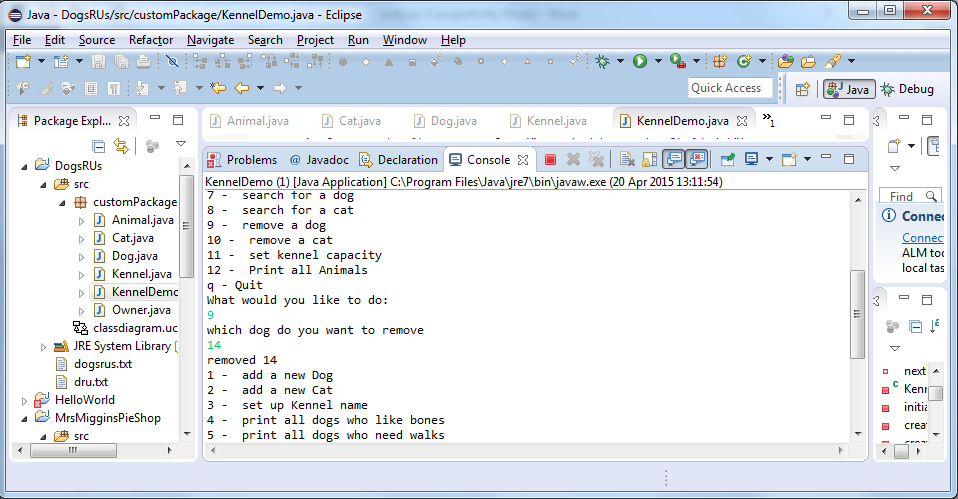


Cat with 14 being searched for

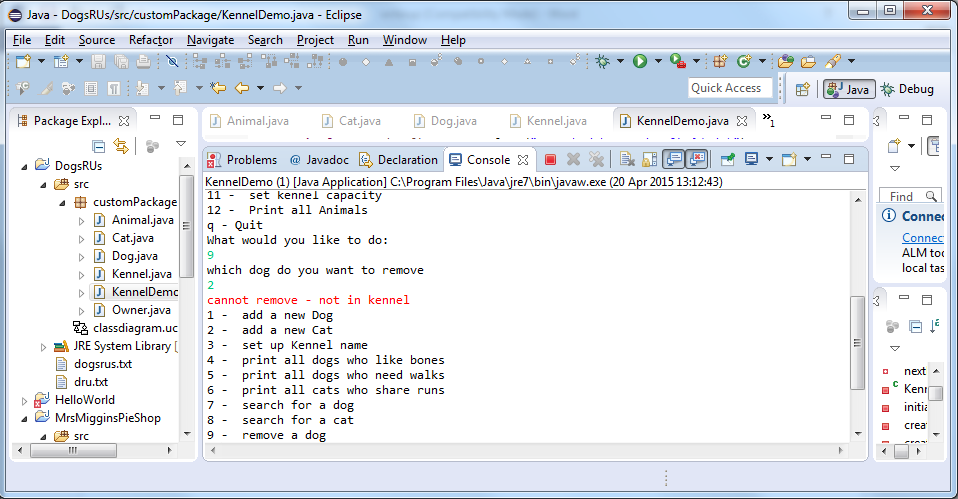


## ID 9.1

Dog with ID 14 being removed

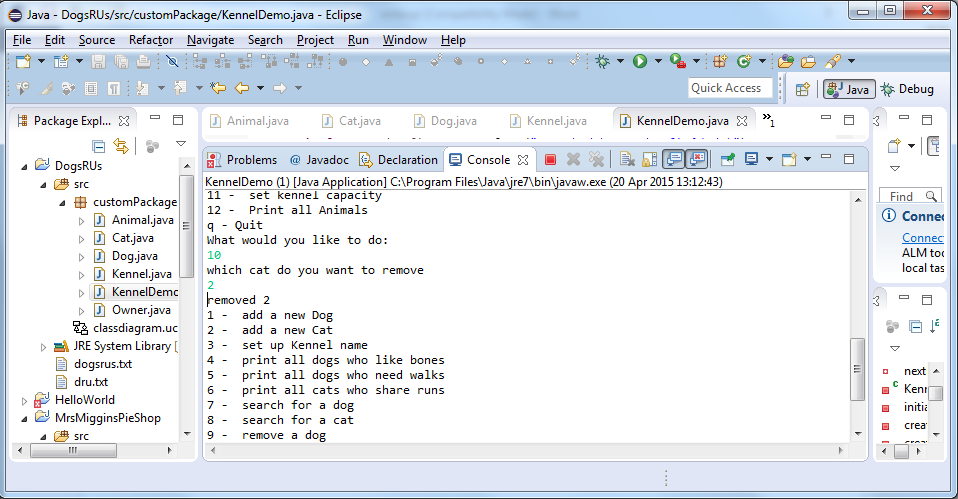


Dog with ID 2 not found

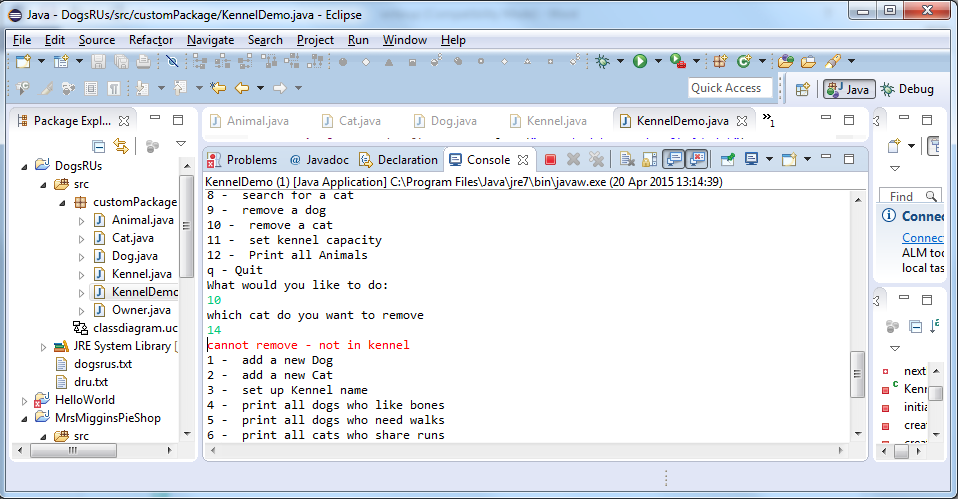


## ID 10.1

Cat with ID 2 removed

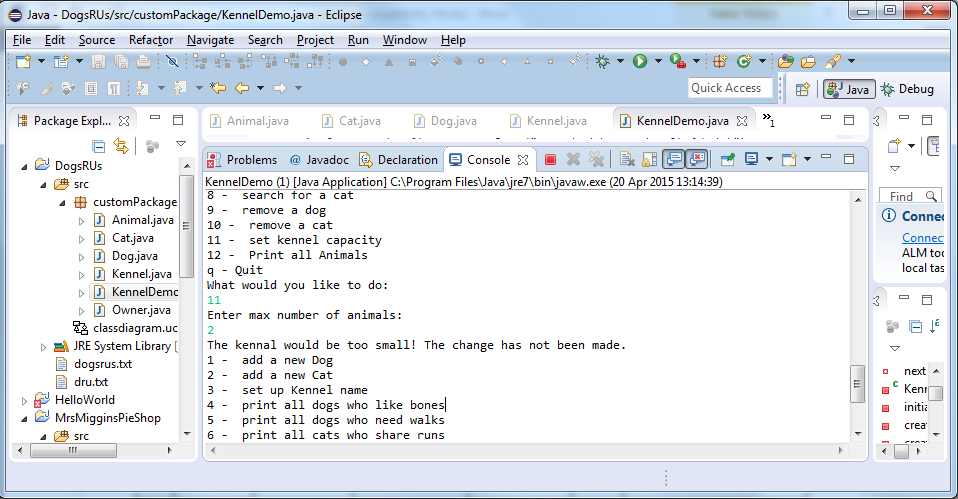


Cat with ID 14 not found to be removed

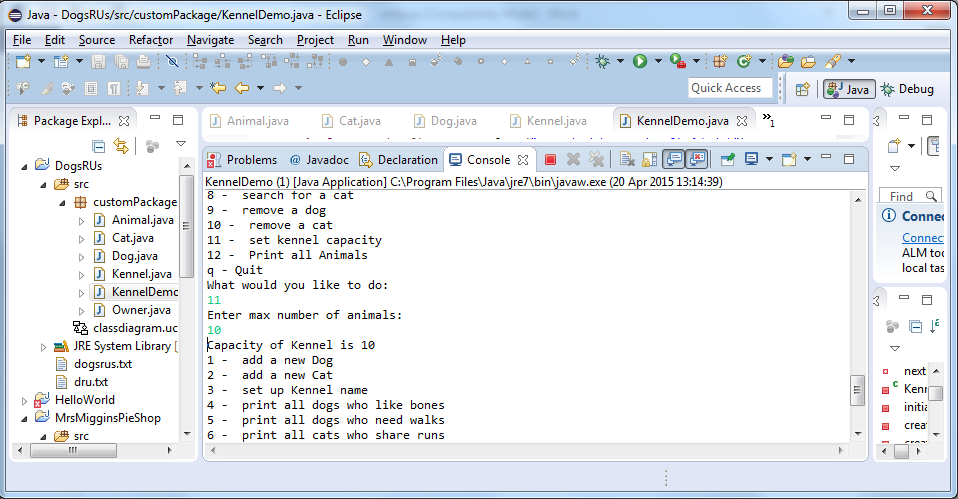


## ID 11.1

Chancing Kennel size to 2 (too small)



Changing Kennel size to 10



## ID 12.1

Sorting all the animals alphabetically and printing them

