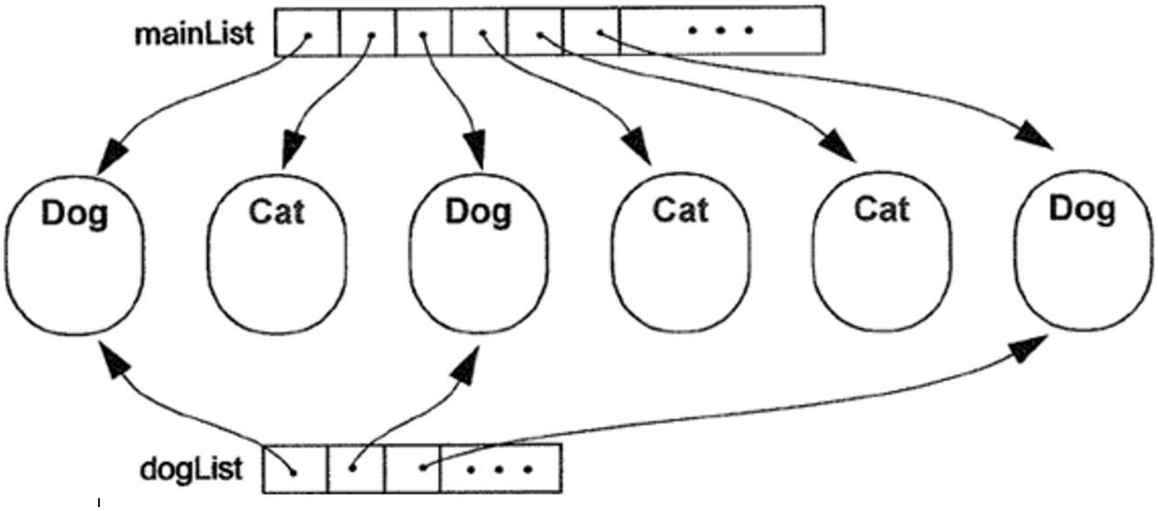
1. Write a program that creates an ArrayList of pets. An item in the list is either a Dog or a Cat. For each pet, enter its name and type ('c' for cat and 'd' for dog). Stop the input when the string STOP is entered for the name. After the input is complete, output the name and type for each pet in the list.

2. Repeat Exercise 1, but this time group the output by printing out the names of all cats first and then the names of all the dogs.

3. Modify the Dog class to include a new instance variable weight (double) and the Cat class to include a new instance variable coatColor (String). Add the corresponding accessors and mutators for the new instance variables. Modify Exercise 1 by inputting additional information appropriate for the type. First you input name and type, as before. If the type is a cat, then input its coat color. If the type is a dog, then input its weight. After the input is complete, output the name, type, and coat color for the cats and the name, type, and weight for the dogs.

4. Suppose you have a list of Dog and Cat objects from Exercise 3 and want to find the average, minimum, and maximum weight of dogs. To compute these values, you must scan the whole list. It would be more efficient if you could get the results by traversing only Dog objects in the list. One approach to achieve this improvement is to create another list that includes only Dog objects (actually references to Dog objects). Here's an example:



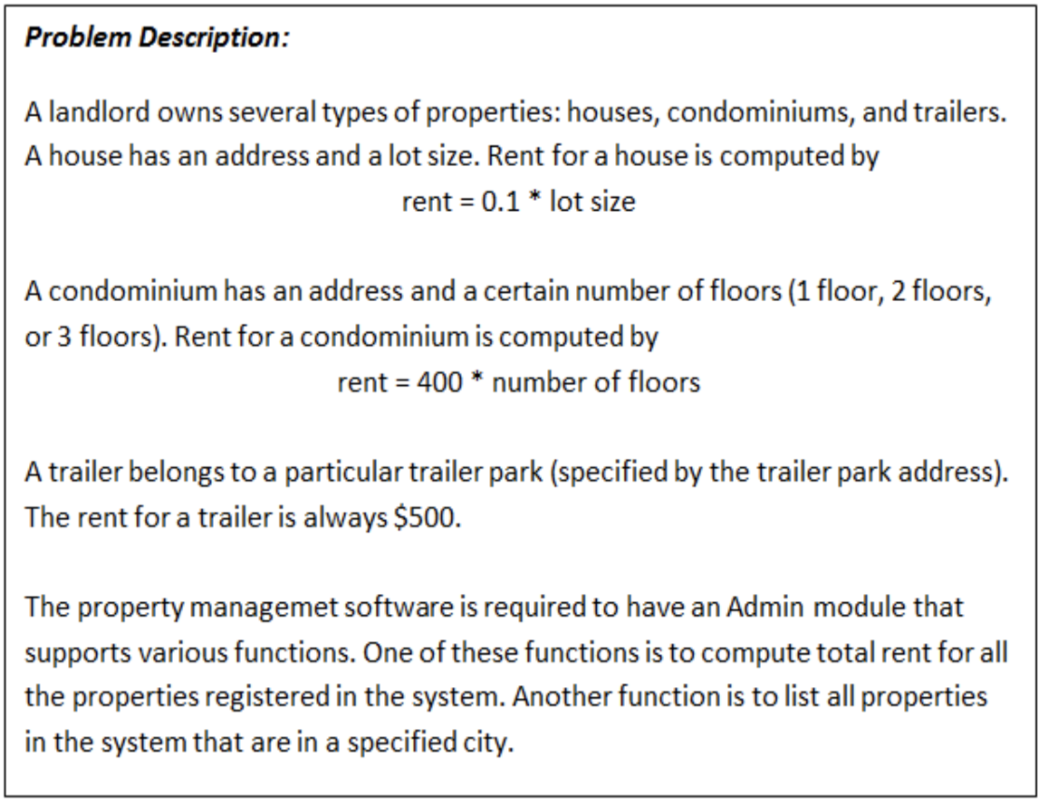
Repeat Exercise 3, but this time create the additional dog list. Then find the average, minimum, and maximum weight of dogs by traversing the dog list.

5. Perform the input routine for the Dog and Cat information as specified in Exercise 3. In addition to creating a main list, create separate cat and dog lists. After the lists are created, allow the user to add or remove information. Display the following menu choices:

```
1. Add Cat
2. Add Dog
3. Remove Cat
4. Remove Dog
0. Quit
```

Allow the user to remove information by specifying the name (assume there are no duplicate names). When adding a new cat or a dog, input the corresponding data values. Make sure to update the lists accordingly when adding or removing a pet. Repeat the operation until the user wants to quit.

6.



Problem Description:

A landlord owns several types of properties: houses, condominiums, and trailers. A house has an address and a lot size. Rent for a house is computed by

rent = 0.1 * lot size

A condominium has an address and a certain number of floors (1 floor, 2 floors, or 3 floors). Rent for a condominium is computed by

rent = 400 * number of floors

A trailer belongs to a particular trailer park (specified by the trailer park address). The rent for a trailer is always $500.

The property managemet software is required to have an Admin module that supports various functions. One of these functions is to compute total rent for all the properties registered in the system. Another function is to list all properties in the system that are in a specified city.