CSE-271: Object-Oriented Programming

Homework #3: Pet Shop

Phase #1: Wed Feb 23 2022 before 11:59 PM

Phase #2: Wed Mar 2 2022 before 11:59 PM

Email-based help Cutoff: 5:00 PM on Tue before deadline

Delayed (by no more than 24-hours) submissions earn only 80% credit

Maximum Points: 31

Key objectives of this project are:

- Gain experience with developing and working with a class hierarchy
- Understand translating UML class diagrams to Java source code
- Recap Java programming using an IDE like Eclipse
- Review and adhere to CSE department’s Style guide
- Use Javadoc to document methods and their return values

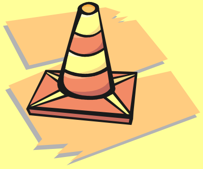
Submission Instructions

This homework assignment must be turned-in electronically via Canvas CODE plug-in. Ensure your program compiles successfully, without any warnings or style errors. Ensure you have documented the methods. Ensure you have tested operations of your classes. Once you have tested your implementation, upload your Java source files onto Canvas via the CODE plug-in.

1. Phase #1: The 9 Java source files developed for this phase of the project.
2. Phase #2: The 9 classes and `PetShop.java`

General Note: Upload each file associated with homework (or lab exercises) individually to Canvas. Do not upload archive file formats such as zip/tar/gz/7zip/rar etc.

Grading Rubric:



The source code submitted for this homework **must pass necessary base case test(s) in order to qualify for earning any score at all**. Programs that do not meet basic requirements or just skeleton code will be assigned zero score! Programs that do not compile, **have even 1 method longer than 25 lines**, or just some skeleton code will be assigned zero score.

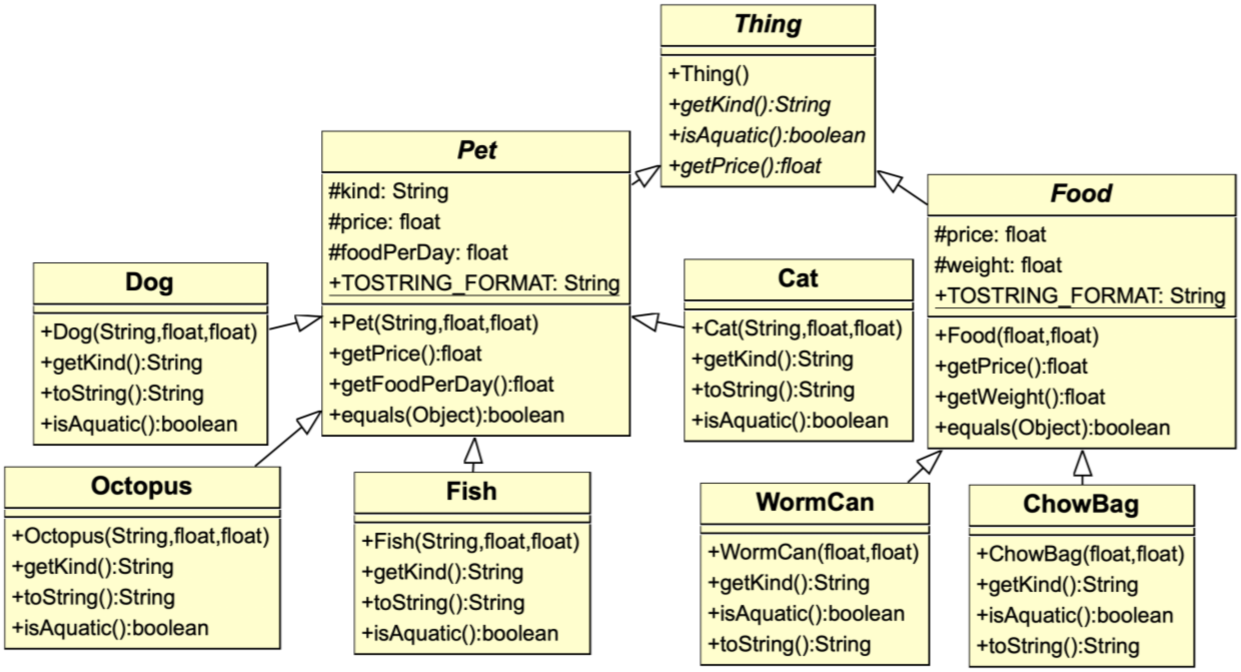
- **NOTE:** Violating CSE programming style guidelines is an error! Your program should not have any style errors.
- **Delayed submission: Only 80% points:** Submission delayed by no more than 24-hours will be accepted for a partial credit of maximum 80% of the points.
- **Conciseness, Formatting & Documentation: 3 points** – Reserved for concise solution, good Javadoc, and formatting.
- **Points**: Phase #1: 2 points per class – 2×9=**18 points** total. Phase #2: **10 points**.

Project Overview:

An object-oriented solution is being developed for a pet shop that has different types of pets and food for them.

Phase #1: Develop hierarch of classes from UML

Analyzing the user's requirements, a hierarchy of classes have been designed. In this part of the project, you are required to develop the Java source code (9 different classes) corresponding to the class hierarchy shown below:



Notes on specific of the methods

1. The return value of `getKind` method should be as follows:
   - For `Pet` objects, the return value must be the `kind` value but with class name as prefix. For e.g., "`Cat: Maincoon`", "`Dog: Poodle`" etc.
   - For `Food` objects, the return of this method must be just the name of the class as in "`WormCan`" or "`ChowBag`".

2. Note the following `toString` output format:
   - For `Pet`s the String representation is generated via:
     ```
     String.format("%s\t%s\t%.2f\t%.2f", "Cat", kind, price, foodPerDay);
     ```
   - For `Food`s the string representation is generated via:
     ```
     String.format("%s\t%.2f\t%.2f", "WormCan", price, weight);
     ```

3. Notes for the `equals` methods in `Pet` and `Food` classes:
   - Two `Pet` objects are equal if they are of the same Java-`class` and `kind` (ignore `price` and `foodPerDay` in comparisons)
   - Two `Food` objects are equal only if they are of the same Java-`class` and have the same `price` and `weight`.

Testing for Phase #1:

You are expected to develop your own approach to test your classes –

- Test each class as you develop them
- For each class test to ensure you are able to create the class
- Print the values returned by different methods to ensure they are consistent

Note: The tester used by CODE (on Canvas) performs similar tests. However, you are not given that tester to emulate a typical job scenario in which you will be expected to carefully test your own source code.

Phase #2: Complete methods in the `PetShop` source

In this phase of the project, you will be using the classes you developed in the previous phase to develop a simple pet store application. You are already given starter code with method stubs in in `PetShop.java` for you to complete. Use the Javadoc for the methods and sample outputs shown below to implement the various methods in `PetShop.java`.

Testing for Phase #2:

For this phase of testing a simple `PetShopUI.java` program is already supplied to you. Review this program and use it for testing methods in `PetShop`. Sample outputs are shown below (user inputs are in **green color**)

Test #1: Loading things from a text file

```
Welcome to the Pet Shop (0 pets & 0 food things).
   What would you like to do [0 for menu]: 0
    1. To add things from a text file
    2. Print summary of things.
    3. Print all things
    4. Print food status
    0. Show this menu
   -1. Quit

Welcome to the Pet Shop (0 pets & 0 food things).
   What would you like to do [0 for menu]: 1
Enter file name to add inventory: few_things.txt
Welcome to the Pet Shop (7 pets & 3 food things).
   What would you like to do [0 for menu]: -1
```

Test #2: Load and print things

```
Welcome to the Pet Shop (0 pets & 0 food things).
   What would you like to do [0 for menu]: 1
Enter file name to add inventory: few_things.txt
Welcome to the Pet Shop (7 pets & 3 food things).
   What would you like to do [0 for menu]: 3
List of all items:
Fish    Goldfish        4.75    0.02
Fish    Guppy   1.25    0.01
Octopus Argonaut        99.99   0.03
ChowBag 17.50   2.50
WormCan 5.75    1.00
WormCan 5.75    1.00
Dog     Poodle  725.00  0.30
Dog     Bulldog 450.50  0.20
Cat     Persian 1400.00 0.10
Cat     Meincoon        325.00  0.15
Welcome to the Pet Shop (7 pets & 3 food things).
   What would you like to do [0 for menu]: -1
```

Test #3: Load and print summary of things

```
Welcome to the Pet Shop (0 pets & 0 food things).
   What would you like to do [0 for menu]: 1
Enter file name to add inventory: few_things.txt
Welcome to the Pet Shop (7 pets & 3 food things).
   What would you like to do [0 for menu]: 2
Summary of items in Pet Shop
Aquatic pets & food summary
    Number of pets      : 3
    Total price pets    : $105.99
    Number of food items: 2
    Total price of food : $11.50
Non-aquatic pets & food summary
    Number of pets      : 4
    Total price pets    : $2900.50
    Number of food items: 1
    Total price of food : $17.50
Welcome to the Pet Shop (7 pets & 3 food things).
   What would you like to do [0 for menu]: -1
```

Test #4: Load and print food status

```
Welcome to the Pet Shop (0 pets & 0 food things).
   What would you like to do [0 for menu]: 1
Enter file name to add inventory: few_things.txt
Welcome to the Pet Shop (7 pets & 3 food things).
   What would you like to do [0 for menu]: 4
Pet Shop food status:
    Daily aquatic food needed      : 0.06 lb
    Daily non-aquatic food needed  : 0.75 lb
    Aquatic food stock in store    : 2.00 lb
    Non-aquatic food stock in store: 2.50 lb
Welcome to the Pet Shop (7 pets & 3 food things).
   What would you like to do [0 for menu]: -1
```

Submission:

This homework assignment must be turned-in electronically via Canvas CODE plug-in. Ensure your program compiles successfully, without any warnings or style errors. Ensure you have documented the methods. Ensure you have tested operations of your program as indicated. Upload the following to Canvas via the CODE plug-in:

1. Phase #1: The 9 Java source files developed for this phase of the project.
2. Phase #2: The 9 classes and `PetShop.java`