Mini-Assignment 3: Kennels

Handed out: Monday 1st March 2021
Hand-in: Friday 26th March 2021, 1pm via Blackboard upload
Feedback: Friday 23rd April 2021
Percentage of overall module marks: 16%

Problem

Lynda runs the district's Dog Home, looking after stray Dogs and those that can't be cared for any more by their original owners. Lynda re-houses her dogs to supporting families and carers in the surrounding areas.

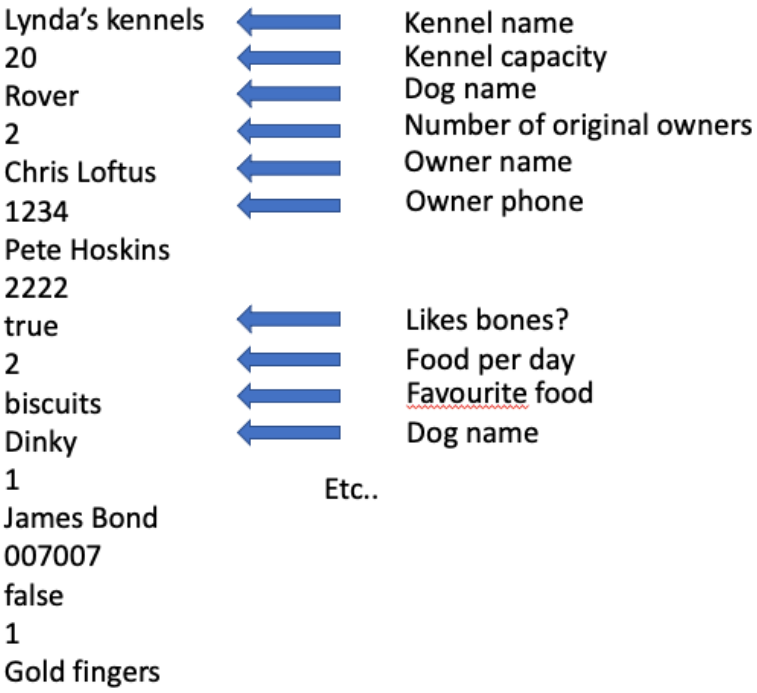
Lynda has an old barn which she has converted into a number of kennels for the dogs that she houses. She keeps a record of all her dogs, together with the original owner (where possible this is the person who donated the dog). Lynda really worries about her dogs and for each one she carefully maintains details on the kind of food they eat, how often they are fed and whether they like bones.

Your task is to write a Java program that will help Lynda organize her kennels.

You are already provided with a version that does not support inheritance. However, Lynda has been asked by her friend Cath to take cats as well as dogs and she has decided to do that. Cats are allowed to share a run (can use the same exercise area as other cats). However, she needs to know which cats are able to share. Lynda decided to use this opportunity to reorganise the code so that it will work more flexibly.

Cats and Dogs are both Animals. Some cats can share enclosed run areas (exercise areas), whereas some can't: they fight. Some dogs need to be taken for a walk whereas some are happy with just the enclosed run provided. Dogs should never share runs. Cats are not interested in bones.

A data file called data.txt is provided (shown next page). This is for the version that only handles dogs. You will need to modify this. When developing a solution make sure you make a backup of your data file, just in case it gets corrupted. I've provided a data-bck.txt backup file.



Steps

1. Copy the zipped kennels IntelliJ project from Blackboard.
2. Run and understand how the existing program works.
3. Some methods are currently not implemented (ENTER CODE HERE / CHANGE CODE / UPDATE CODE comments). You should only update the Kennel class for this step.
    a. Implement those methods, filling in any missing code.
    b. Also, if required, rework the existing code to make it more robust and to do proper error checking.
    c. Make sure all non-private code has suitable Javadoc comments.
    d. Also, change the toString methods to use StringBuilder instead of concatenating Strings. Do some research on how to do this (i.e. look at the Java library documentation); StringBuilder is more efficient **in some circumstances** than using String concatenation, e.g. where loops are required in the toString method.
4. Rework the provided code to support both cats and dogs:
    a. You must use inheritance as part of this solution. **See the provided example-inheritance.zip IntelliJ project. This will help with solving the current problem.**
    b. Equality is determined by animal name only.
    c. Provide the ability to sort animals by name. Change the *printAll* method so that it always lists all the animals in order of their name. See Workshop 13 slides.
    d. Update the existing KennelDemo class and data files to deal with Cats and Dogs. This should deal with the fact that some cats may share runs and some dogs may have daily walks. Make sure that you use polymorphic variables and parameters wherever possible. Avoid duplicated code.

Hand-in

Upload one zip file (not rar or anything else: 2 percentage points will be deducted if the wrong format is used) to Blackboard that contains:

1. Your IntelliJ project folder.
    a. If you are not using IntelliJ then package up the source code files and class files.
2. A pdf (2 percentage points will be deducted if the wrong format is used) document that contains:
    a. A front cover page that includes your name, email, document date, and the title of this mini-assignment.
    b. Page one must contain a UML use case diagram showing system functionality. This must be drawn with a software tool of your choice, such as draw.io used in workshops.
    We will cover use case diagrams in Seminar/Lecture 8.
    c. Page two must contain a UML class diagram for the final program. This must be drawn with software tool of your choice. It must not be auto-generated from existing code.
    d. 500 words (+/- 10%) that describe how you went about solving the assignment: what was difficult, what remains to be done, how you tested your solution, what mark do you think you should be awarded and why. This word count excludes the cover page and words inside diagrams.
    e. Pages with screenshots showing the program running: all features shown with a caption under each screenshot saying what it shows.

Note: this is an "individual" assignment and must be completed as a one-person effort by the student submitting the work.

This assignment is **not** marked anonymously.

PLEASE NOTE: By submitting your work for this worksheet using Blackboard, you are making the following declaration:
This submission is my own work, except where clearly indicated.
I understand that there are severe penalties for plagiarism and other unfair practice, which can lead to loss of marks or even the withholding of a degree. I have read the sections on unfair practice in the Students' Examinations Handbook and the relevant sections of the current Student Handbook of the Department of Computer Science. I understand and agree to abide by the University's regulations governing these issues.
See the guidelines in the Student Handbook regarding Unacceptable Academic Practice [1]

Marking scheme

Assessment will be based on the assessment criteria described in Appendix AA of the Student Handbook [2].
However, the following table gives you some indication of the weights associated with individual parts of the assignment. This will help you judge how much time to spend on each part. Note that I will award full marks for each category below

if you fully complete the requested functionality and to a good quality. However, the flair marks represent work that goes beyond the requested features.

Documentation	Does the documentation contain a good quality use-case diagram and class diagram? Does it also convey a convincing and detailed story of how the code was implemented and tested (including screenshots), problems encountered, what was learned and an indication of the mark that should be awarded and why? Were uploading instructions followed correctly?	20%
Implementation: refactoring (step 3)	Correct implementation of step 3. Quality of the code: excellent identifier names, following Java naming conventions, error checking, appropriate indentation, use of small methods and private methods. Classes that are highly cohesive. Appropriate use of access modifiers (private, public).	30%
Implementation: using inheritance (step 4)	Correct implementation of step 4 and all its parts. As above, and also appropriate use of inheritance.	40%
Flair	I am looking for applications that show flair, creativity or innovation. This addresses the Appendix AA 80% to100% assessment criteria that state: "and will more than completely fulfill the functional requirements." This could include things such as more functionality; use of JSON to store the data; etc. Ask me if you are not sure.	10%

References

[1] Computer Science Department (2020) "Student Handbook" (Online) https://impacs-inter.dcs.aber.ac.uk/en/cs-undergraduate/official-information/student-handbooks (Accessed 23rd February 2021)

[2] Appendix AA. Assessment Criteria for Development (Online) https://impacs-inter.dcs.aber.ac.uk/images/editor-content/Documentation/Handbooks/Appendices/AppendixAA.pdf (Accessed 23rd February 2021)