CS 101 - Algorithms & Programming I

Fall 2024 - Lab 9

Due: Week of December 9, 2024

*Remember the **honor code** for your programming assignments.*
*For all labs, your solutions must conform to the CS101 style **guidelines**!*
All data and results should be stored in variables (or constants where appropriate) with meaningful names.

This lab aims to learn how to define custom objects and understand object references. Object-oriented programming is fundamentally about creating a collection of interacting objects that communicate through references. It's important to analyze your problems and design your solution on paper before starting the implementation. In this lab, you will organize your data and the methods that operate on them, allowing you to effectively manage your virtual pets within the system.

0. Setup Workspace

Start Visual Studio Code (VSC) and open your previously created workspace named `labs_ws`. Under the `labs` folder, create a new folder named `lab9`.
In this lab, you will have four Java classes/files under the `labs/lab9` folder that make up the **Virtual Pet Adoption System** as described below for the original assignments. Create classes/files with different names (e.g. `PetRev`) for revisions as needed. Submit your completed classes **without compressing them**. Do ***not* upload other or previous lab solutions** in your submission.

Outputs of sample runs are shown in **brown** whereas the user inputs are shown in **blue**.

Important Note:
All data members should have private access modifiers and be accessed or modified, if applicable, using accessor (getter) and mutator (setter) methods, as required. Implement constructors and standard methods such as `toString()` and `equals()` where appropriate.

1. Virtual Pet Adoption System

You are tasked with developing a program that allows users (multi-user) to adopt and manage virtual pets. This system enables users to create and interact with their pets, tracking attributes such as health, happiness, and mood. The focus is on implementing object-oriented principles such as encapsulation, constructors, and method overriding.
You will be provided with a main class (see `VPAManager.java`). Your task is to implement the necessary other classes and their methods as described below to complete the functionality sketched in this main class. You may also create additional helper methods and classes as needed.

Class `Pet`

Attributes:

– The name of the pet

– The type of pet (Dog, Cat, Rabbit)

– Health status (0 to 100)

– Happiness level (0 to 100)

– Current mood (Happy, Sad, Playful)

– The owner user object (a reference to a `User` object), if currently adopted

– The pet's age in years (increments with user interactions).

– A flag indicating whether the pet is “old” (10 years for Cats, 8 years for Dogs, 6 years for Rabbits).

Constructors:

– A constructor that initializes a pet’s name, type, and owner user with specified values. The health is set to 50, and the happiness to 50 by default. The mood, on the other hand, is set based on happiness (as Playful). The user of a pet should only be changed when the pet is released by the user’s associated method or adopted by another user (see below).

Methods:

– A method to feed the pet. Increases the pet's health by 10 (max 100), and happiness by 5 (max 100). Print a message if health has already been maxed. Adjust mood based on happiness.

– A method to play with the pet. When called, it boosts the pet's happiness and changes its mood. Happiness increases by 15 (max 100). Print a message if happiness has already been maxed.

– A method to groom the pet. Increases health by 15 (max 100). Print a message if health has already been maxed.

– A method to update the pet's mood based on its happiness level, setting it to "Happy" if happiness is greater than 70, "Playful" if it's between 51 and 70, and "Sad" if 50 or below.

– A method that returns the pet’s status as a formatted string with the pet's name, type, health, happiness, and mood.

– A method that increments the pet’s age by 0.25 years after every user interaction.

– Getters and setters for each attribute, where applicable.

– Override `toString()` to return pet information as a string.

– Override `equals()` to compare pets based on their name and type.

Class `User`

Attributes:

– The name of the user.

– The password of the user

– A list of pets adopted by the user.

– The user’s budget starts at 100 points, and adopting pets reduces the budget.

Constructors:

– A constructor that initializes the user with a specified username, password, and an empty list of pets.

Methods:

– A method that verifies if the given password matches the user’s stored password.

– A method to adopt a pet. Creates/instantiates and adds a new `Pet` object to the user's collection with the provided name and type, and the user's budget is reduced accordingly. printing a confirmation message.

Pets cost points depending on their type (Cats: 30 pts, Dogs: 40 pts, Rabbits: 20 pts).

Prevent adoption if the user does not have enough points.

– A method to adopt an existing pet (previously released by another user) specified by its name. Adds the given pet to the user's collection, removes it from the collection of unowned pets maintained by the `VPNManager`, and prints a confirmation message.

– Releases the pet by removing the provided pet (a reference to a `Pet` object) from the user's collection and adds it to the unowned pet collection maintained by the `VPNManager` if it exists; print an error if not found. It returns the removed pet if any, `null` otherwise.

– Displays all adopted pets of this user using the status display method of the `Pet` class for each pet.

– Returns a pet object given its index in the list.

– Returns a pet object given its name (assume all pets are uniquely named for each type).

– Searches and returns pets whose happiness is greater than or equal to the specified minimum happiness.

– Searches and returns pets whose health is greater than or equal to the specified minimum health.

– Displays the average health and happiness of the user's pets.

– Getters and setters for each attribute, where applicable.

– Override `toString()` to return the user's information and their pets (use `toString()` method of the `Pet` class in turn).

– Override `equals()` to compare users based on their usernames.

Class `VPAManager`

You will be provided with a main class skeleton that outlines basic functionality by instantiating a user object with the provided name and applying pet-related operations for this user such as creating a user, adopting pets, interacting with pets (feeding, playing, grooming), listing or searching pets based on attributes like happiness or health, and showing a summary of the user's pets.

By the end of this lab, you will have created a fully functional virtual pet management system, enhancing your understanding of object-oriented programming concepts while providing an engaging user experience.

Note that the parts highlighted in sample runs are for drawing attention to important parts of the output.

Sample run:

```
Welcome to the Virtual Pet Adoption System!
1. Register
2. Login
3. Exit
Choose an option: 1
Enter a username: Selda
Enter a password: 12345
Registration successful! You can now log in.

Welcome to the Virtual Pet Adoption System!
1. Register
2. Login
3. Exit
Choose an option: 2
Enter your username: Selda
Enter your password: 123456
Invalid username or password. Please try
again.

Welcome to the Virtual Pet Adoption System!
1. Register
2. Login
3. Exit
Choose an option: 2
Enter your username: Selda
Enter your password: 12345
Login successful! Welcome, helya!

Menu:
1. Adopt a new pet
2. Adopt an unowned pet
3. Release a pet
4. Feed a pet
5. Play with a pet
6. Groom a pet
7. Display your pets
8. Search pets by happiness
9. Search pets by health
10. Display average health and happiness
11. Logout
12. Switch User
Choose an option: 1
Enter pet name: Gofi
Enter pet type (Dog/Cat/Rabbit): Dog
You have adopted a new Dog named Gofi.

Menu:
1. Adopt a new pet
…
7. Display your pets
…
Choose an option: 7
Your pets:
Name: Gofi, Type: Dog, Health: 50, Happiness:
50, Mood: Sad

Menu:
1. Adopt a new pet
…
Choose an option: 1
Enter pet name: Ego
Enter pet type (Dog/Cat/Rabbit): Cat
You have adopted a new Cat named Ego.

Menu:
1. Adopt a new pet
2. Adopt an unowned pet
3. Release a pet
…
Choose an option: 3
Enter the name of the pet to release: Gofi
You have released Gofi the Dog.

Menu:
1. Adopt a new pet
…
7. Display your pets
…
Choose an option: 7
Your pets:
Name: Ego, Type: Cat, Health: 50, Happiness:
50, Mood: Sad

Menu:
1. Adopt a new pet
…
4. Feed a pet
…
Choose an option: 4
Enter the name of the pet to feed: Ego

Menu:
…
12. Switch User
Choose an option: 12
You have been logged out.
Enter your username: Selda
Enter your password: 12345
Login successful! Welcome, helya!

Menu:
…
10. Display average health and happiness
…
Choose an option: 10
Average Health: 60.00, Average Happiness:
55.00

Menu:
…
5. Play with a pet
…
Choose an option: 5
Enter the name of the pet to play with: Ego

Menu:
…
7. Display your pets
…
Choose an option: 7
Your pets:
Name: Ego, Type: Cat, Health: 60, Happiness:
70, Mood: Playful

Menu:
…
11. Logout
…
Choose an option: 11
You have been logged out.

Welcome to the Virtual Pet Adoption System!
1. Register
2. Login
3. Exit
Choose an option: 3
```