5004 Final Synthesis Report

Lulu Xu

1. Introduction

- **Project Overview**

This Java project is a Pet Health Record Application following the Model-View-Controller (MVC) architectural pattern. The application allows users to manage health records for pets, including health checkups and follow-up events. The project consists of multiple classes, each playing different roles within the MVC architecture.

- **Classes in the MVC Architecture**

Model Layer

1. **Pet (Abstract Class)**
    - Description: Represents a pet, including basic attributes such as name, age, breed, and health records.
    - Subclasses: **Dog** and **Cat**, providing specific implementations for different types of pets.
    - Association: Associated with **HealthEvent** for storing health events.
2. **Dog and Cat**
    - Description: Concrete implementation classes of **Pet**, representing dogs and cats.
    - Functionality: Implement the **HealthCheckable** interface, providing pet-specific health check methods.
    - The **Pet** class **hierarchy** reflects the relationships between different types of pets. For example, a **Dog** or **Cat** is a specific type of pet, and they inherit attributes and behavior from the **Pet** class. This hierarchy represents the relationships between pet types in your application.
3. **HealthEvent (Abstract Class)**
    - Description: Represents a health event for a pet.
    - Subclass: **HealthRecord**, providing a concrete implementation for specific health records.
4. **HealthRecord**
    - Description: A concrete implementation of **HealthEvent**, representing a specific health record.
5. **HealthCheckable (Interface)**
    - Description: Defines **performCheckup** and **scheduleCheckup** methods for performing and scheduling health checks.

View Layer

1. **PetView**
    - Description: Responsible for displaying the user interface and providing elements for user interaction.
    - Functionality: Displays pet information, responds to user actions.

Controller Layer

1. **PetController**
    - Description: Handles user input, controls interactions between Model and View.
    - Functionality: Responds to UI events, updates Model and View.

2. Concept Map

Concept 1: Recursion in Practice	I used this concept in the Pet class when implementing the findEventByDescription method. This recursive method searches for a specific health event by description within a list of health events. You can find this in the Pet class, specifically in the findEventByDescription method on lines where recursion is used to iterate through the list.
Concept 2: Abstract Classes and Interfaces	I applied this concept extensively in the project. For example, the Pet class is an abstract class, serving as a base for concrete pet types like Dog and Cat. Additionally, the HealthCheckable interface defines the methods performCheckup and scheduleCheckup that are implemented by concrete pet classes.
Concept 3: Abstracted Linked Lists	The abstract health events using the HealthEvent class, and specific health records using the HealthRecord class. These classes provide abstraction for managing health-related data.
Concept 4: Higher order functions map, filter, and fold	I demonstrated this concept in the Pet class with the use of higher-order functions. For example, the mapEvents and filterEvents methods use higher-order functions to transform and filter health events. The foldEvents method also employs higher-order functions for accumulating data during event processing.
Concept 5: Hierarchical Data Representation	This concept is exemplified by the class hierarchy in the project. The Pet class serves as the base, while Dog and Cat are hierarchical subclasses. The health event hierarchy is also present, with HealthEvent as the base class and HealthRecord as a specific type
Concept 6: MVC Design	My project adheres to the Model-View-Controller (MVC) design pattern. The PetController class serves as the controller, managing user input and interactions. The PetView class represents the view, responsible for

	displaying the user interface. The Pet and HealthEvent classes form the model, encapsulating data and logic.
Concept 7: SOLID Principles	The SOLID principles are followed in your project. The Single Responsibility Principle (SRP) is upheld in the PetController class, which has a single responsibility. The Open/Closed Principle (OCP) is supported by the code's extensibility. The Liskov Substitution Principle (LSP) is maintained through subclasses like Dog and Cat. The Interface Segregation Principle (ISP) is followed with focused interfaces like HealthCheckable. Finally, the Dependency Inversion Principle (DIP) is embraced through the use of abstractions.
Concept 8: Observer Pattern	This pattern could be seen in the way PetView and PetController interact. The PetView could act as the subject, sending out updates (like user input), while PetController acts as an observer, reacting to these updates. This is particularly evident in how user actions in the view trigger methods in the controller.

3. Academic Integrity Statement

I understand that my learning is dependent on individual effort and struggle, and I acknowledge that this assignment is a 100% original work and that I received no other assistance other than what is listed here.

Acknowledgements and assistance received:

I did not use generative AI in any form to create this content and the final content was not adapted from generative AI created content.

I did not view content from any one else's submission including submissions from previous semesters nor am I submitting someone else's previous work in part or in whole.

I am the only creator for this content. All sections are my work and no one else's with the exception being any starter content provided by the instructor. If asked to explain any part of this content, I will be able to.

By putting your name and date here you acknowledge that all of the above is true and you acknowledge that lying on this form is a violation of academic integrity and will result in no credit on this assignment and possible further repercussions as determined by the Khory Academic Integrity Committee.

Name: Lulu Xu	Date: 12/11/2023

4. Code files

https://github.com/lululoveyuanyuan/Final_Project_5004.git

5. Video Walkthrough

https://youtu.be/XjvudK_Uz8k?si=5aftWx50KUDpTlNs