William Fishburn

Milestone Assignment 3-2

**Enhancement One:**

**Software Design and Engineering**

#### 

### **1) Briefly describe the artifact. What is it? When was it created?**

The artifact is a Rescue Animal system built for animal shelters, developed as a Java application for managing and reserving animals; specifically dogs and monkeys. The system allows users to intake new animals, reserve animals based on availability, and print lists of animals by animal type or as a collection of all animals. This project was created in January 2023 and was my first Java project.

### **2) Justify the inclusion of the artifact in your ePortfolio.**

**I. Why did you select this item?**

I selected this artifact because it represents a foundational project that demonstrates my initial foray into Java programming. It showcases my ability to design and implement a functional application using core Java concepts such as object-oriented programming, data management, and user input handling. This project serves as a concrete example of my early development skills and my progress over time.

**II. What specific components of the artifact showcase your skills and abilities in software development?**

* **Object-Oriented Design**: The use of classes (Dog and Monkey) to model real-world entities and manage their attributes.
* **User Input Handling**: Methods to intake and validate user inputs, ensuring robust data management.
* **Data Management**: Utilization of ArrayList to store and manage collections of animal objects.
* **Error Handling**: Implementation of try-catch blocks to handle input mismatches and other exceptions.

**III. How was the artifact improved?**

* **Documentation**: Added detailed Javadoc comments and inline comments to explain the purpose, inputs, and outputs of each method.
* **Code Refactoring**: Consolidated repetitive code into reusable methods to improve maintainability.
* **Input Validation**: Enhanced user input validation and error handling to ensure the system can handle incorrect inputs gracefully.
* **Optimization**: Optimized loops and branching for better performance and readability.

### 

**3) Did you meet the course objectives you planned to meet with this enhancement in Module One?**

The enhancements made in Module One aligned with my course objectives, particularly in improving code readability, error handling, and refactoring. These improvements helped in achieving:

* **Better Code Maintainability**: Through refactoring and documentation.
* **Enhanced User Experience**: By implementing robust input validation and error handling.
* **Optimized Performance**: By refining loops and branching structures.

**I. Do you have any updates to your outcome-coverage plans?**

There have been no major changes to the outcome-coverage plans. The enhancements met the initial objectives and demonstrated a clear improvement in my software development capabilities.

### **4) Reflect on the process of enhancing and modifying the artifact.**

**I. What did you learn as you were creating it and improving it?**

* **Importance of Documentation**: Clear documentation is crucial for maintaining and understanding code, especially when working on collaborative projects or revisiting code after some time.
* **Effective Error Handling**: Proper validation and error handling improve user experience and prevent application crashes due to unexpected inputs.
* **Code Refactoring**: Refactoring helps manage and simplify codebases, making them easier to maintain..

**II. What challenges did you face?**

* Ensuring that the code remains simple and readable while incorporating robust features like error handling and input validation. It took much attention to detail, implementing comprehensive input validation to cover all possible edge cases and user errors.

#### 

#### 

#### 

#### 

#### **Enhancements Checklist**

#### **Enhancement 1: Add Proper Documentation**

* **Description**: Each method now includes detailed documentation explaining its purpose, inputs, outputs, and any enhancements made.
* **Purpose**: Improves code readability and maintainability, making it easier for other developers (or future me) to understand the code.
* **Implementation**:
  + Added Javadoc comments for each method.
  + Provided inline comments to explain key parts of the code logic.

#### **Enhancement 2: Refactor Repetitive Code**

* **Description**: Repetitive code was refactored into reusable methods to enhance maintainability and readability.
* **Purpose**: Simplifies the codebase by reducing redundancy, making it easier to manage and less prone to errors.
* **Implementation**:
  + Created reusable methods for input prompts and validation checks.

#### **Enhancement 3: Improve User Input Validation and Error Handling**

* **Description**: Enhanced input validation and error handling to ensure the system can handle unexpected inputs gracefully.
* **Purpose**: Enhances user experience by providing clear feedback and preventing crashes due to invalid inputs.
* **Implementation**:
  + For the printAnimals statement, choices have been modified to reflect the desired selection: “d” for Dog, “m” for Monkey, and selection “a” to print a list of all animals.

#### **Enhancement 4: Optimize Loops and Branches**

* **Description**: Optimized loops and branches for better performance and readability.
* **Purpose**: Ensures the code runs efficiently and is easy to follow.
* **Implementation**:
  + Refactored main menu loop from if/else to switch/case statements.
  + Ensured that all switch statements contain a default case.