Art Gallery Management System

A Comprehensive Report for Task #4

Aleksandre Maghlakelidze
Student at Ilia State University
alex.maglakelidze@gmail.com, aleksandre.maghlakelidze.1@iliauni.edu.ge

Description

Create an Art Gallery Management System (AGMS) in Java. An AGMS is a software application used by art galleries to manage exhibitions and legal entity information. Our example is a basic one, which has the following features:

1. Storage for exhibition records
2. Ability to add a new exhibition record
3. Ability to remove an exhibition record
4. Ability to print exhibition information to the console
5. Ability to save and load exhibition data to/from a file
6. Implementation of legal entity information

AGMS Structure

We will need the following classes and interface for the software:

1. LegalEntity – an interface for legal entity information.
2. Exhibition – the exhibition record itself.
3. ArtGallery – the art gallery management system.

Interface LegalEntity

The LegalEntity interface defines the contract for legal entity information:

```
package finalexam.task4;

public interface LegalEntity {
    String getAddress();
    String getVatNumber();
```

Class Exhibition

The Exhibition class represents individual exhibitions with name, start date, and end date:

```
package finalexam.task4;

public class Exhibition {
    // Core attributes of an exhibition
    private String name;
    private String startDate;
    private String endDate;

    // Constructor to initialize all fields
    public Exhibition(String name, String startDate, String endDate) {
        this.name = name;
        this.startDate = startDate;
        this.endDate = endDate;
    }

    // Getter methods for accessing private fields
    public String getName() {
        return name;
    }

    public String getStartDate() {
        return startDate;
    }

    public String getEndDate() {
        return endDate;
    }

    // Override equals method for proper object comparison
    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Exhibition that = (Exhibition) obj;
        return name.equals(that.name) &&
startDate.equals(that.startDate) && endDate.equals(that.endDate);
    }

    // Override hashCode for consistency with equals method
    @Override
```

```
    public int hashCode() {
        return name.hashCode() + startDate.hashCode() +
endDate.hashCode();
    }

    // Override toString for meaningful string representation of the
object
    @Override
    public String toString() {
        return "Exhibition{" +
                "name='" + name + '\'' +
                ", startDate='" + startDate + '\'' +
                ", endDate='" + endDate + '\'' +
                '}';
    }
}
```

This class encapsulates the data for each exhibition and provides methods for comparison and string representation.

Class ArtGallery

The ArtGallery class implements the LegalEntity interface and manages the list of exhibitions:

```
package finalexam.task4;

import java.io.*;
import java.util.ArrayList;
import java.util.List;

public class ArtGallery implements LegalEntity {
    // Core attributes of an art gallery
    private String name;
    private String address;
    private String vatNumber;
    private List<Exhibition> exhibitions;
    private static final String FILE_NAME = "exhibitions.txt";

    // Constructor to initialize the gallery with legal entity
information
    public ArtGallery(String name, String address, String vatNumber) {
        this.name = name;
        this.address = address;
```

```
        this.vatNumber = vatNumber;
        this.exhibitions = new ArrayList<>();
    }

    // Implementation of LegalEntity interface methods
    @Override
    public String getAddress() {
        return address;
    }

    @Override
    public String getVatNumber() {
        return vatNumber;
    }

    // Method to add a new exhibition to the gallery
    public void addExhibition(Exhibition exhibition) {
        if (exhibition != null && !exhibitions.contains(exhibition)) {
            exhibitions.add(exhibition);
        }
    }

    // Method to remove an exhibition from the gallery
    public boolean removeExhibition(Exhibition exhibition) {
        return exhibitions.remove(exhibition);
    }

    // Method to print all exhibitions in the gallery
    public void printExhibitions() {
        for (Exhibition e : exhibitions) {
            System.out.println(e);
        }
    }

    // Method to get the total number of exhibitions
    public int getExhibitionCount() {
        return exhibitions.size();
    }

    // Method to save all exhibitions to a file
    public void saveExhibitionsToFile() {
        try (BufferedWriter writer = new BufferedWriter(new
FileWriter(FILE_NAME))) {
            for (Exhibition e : exhibitions) {
```

```
                writer.write(e.getName() + "," + e.getStartDate() +
"," + e.getEndDate());
                writer.newLine();
            }
            System.out.println("Exhibitions saved successfully.");
        } catch (IOException e) {
            System.out.println("Error saving exhibitions: " +
e.getMessage());
        }
    }

    // Method to load exhibitions from a file
    public void loadExhibitionsFromFile() {
        exhibitions.clear();
        try (BufferedReader reader = new BufferedReader(new
FileReader(FILE_NAME))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts.length == 3) {
                    exhibitions.add(new Exhibition(parts[0], parts[1],
parts[2]));
                }
            }
            System.out.println("Exhibitions loaded successfully.");
        } catch (IOException e) {
            System.out.println("Error loading exhibitions: " +
e.getMessage());
        }
    }
}
```

This class is the core of the AGMS, providing methods to add, remove, and display exhibitions, as well as save and load the exhibition list to/from a file.

The Art Gallery Management System is designed to help art galleries efficiently manage their exhibitions and maintain legal entity information. It provides a flexible structure for storing exhibition data and implements file I/O operations for data persistence.