README

Attractions_ds Class

The Attractions_ds class manages information about attractions in the zoo. It includes attributes like the attraction's name, description, ticket price, unique identifier, schedule status, and discounts for minors and seniors.

Attributes:

attraction_name: Name of the attraction.

description: Description of the attraction.

ticket_price: Price of a ticket for the attraction.

unique_id: Unique identifier for each attraction.

ticketed: Number of visitors with tickets for the attraction.

schedule: Schedule status of the attraction (e.g., "CLOSED" or other statuses).

minor_discount: Discount percentage for minor visitors.

senior_discount: Discount percentage for senior visitors.

Methods:

setName(String name): Set the name of the attraction.

getName(): Get the name of the attraction.

setDescription(String description): Set the description of the attraction.

getDescription(): Get the description.

setTicket(Double price): Set the ticket price.

getTicket(): Get the ticket price.

setId(int id): Set the unique identifier.

getId(): Get the unique identifier.

setSchedule(String schedule): Set the schedule status.

getSchedule(): Get the schedule status.

setMinor(int discount): Set the discount for minors.

getMinor(): Get the minor discount percentage.

setTicketed(int value): Increase the ticketed count.

getTicketed(): Get the ticketed count.

setSenior(int discount): Set the discount for seniors.

getSenior(): Get the senior discount percentage.

Visitor_ds Class

The Visitor_ds class represents visitors to the zoo, including their personal and financial information.

Attributes:

name: Name of the visitor.

age: Age of the visitor.

phone_no: Phone number of the visitor.

balance: Account balance of the visitor.

email: Email address of the visitor.

password: Password of the visitor.

membership: Membership level of the visitor.

spend: Spending at the zoo.

feedback: Visitor's feedback.

ticketed: Number of tickets purchased by the visitor.

Methods:

setTicketed(int value): Increase the ticketed count.

getTicketed(): Get the ticketed count.

setFeedback(String feedback): Set visitor feedback.

getFeedback(): Get visitor feedback.

setMembership(String level): Set the membership level.

getMembership(): Get the membership level.

setName(String name): Set the name of the visitor.

setSpend(Double cash): Increase the spending.

setAge(int age): Set the age of the visitor.

setPhone(String phone): Set the phone number.

setBalance(Double balance): Set the account balance.

setEmail(String email): Set the email address.

setPassword(String password): Set the password.

getName(): Get the name of the visitor.

getEmail(): Get the email address.

getPassword(): Get the password.

getBalance(): Get the account balance.

getAge(): Get the age of the visitor.

getSpend(): Get the spending.

Visitor_list Class

The Visitor_list class manages a list of visitors and provides methods for adding visitors, checking visitor existence based on email and password, and retrieving the visitor list.

Methods:

setVisitor(Visitor_ds visitor): Add a visitor to the list.

exists(Visitor_ds visitor): Check if a visitor with the same email and password exists.

getVisitorlist(): Get the list of visitors.

Animal_ds Class

The Animal_ds class holds information about zoo animals.

Attributes:

animal_name: Name of the animal.

animal_detail: Details about the animal.

animal_type: Type or category of the animal.

noise: Sound made by the animal.

Methods:

setAnimalName(String name): Set the name of the animal.

setAnimalDetail(String detail): Set details about the animal.

setAnimalType(String type): Set the animal's type or category.

setAnimalNoise(String noise): Set the noise made by the animal.

getAnimalName(): Get the name of the animal.

getAnimalDetail(): Get details about the animal.

getAnimalType(): Get the type of the animal.

getAnimalNoise(): Get the noise made by the animal.

Animal_list Class

The Animal_list class manages a list of zoo animals. It provides methods to add animals, retrieve the animal list, remove animals by name and type, and modify animal details.

Methods:

set_animal(Animal_ds animal): Add an animal to the list.

getAnimalList(): Get the list of animals.

get_animal(): Display the details of animals in the list.

remove(String name, String type): Remove an animal from the list based on its name and type.

modify(String new_name, String new_type, String detail, String noise): Modify the details of an animal in the list based on its name and type.

Structure Class

The Structure class is an abstract class with common attributes for all classes, including name, password, and email. It defines an abstract method login_menu() to be overridden in subclasses.

Attributes:

name: Name of the structure.

password: Password.

email: Email address.

Methods:

setName(String name): Set the name.

setEmail(String email): Set the email.

getEmail(): Get the email.

getName(): Get the name.

setPassword(String password): Set the password.

getPassword(): Get the password.

Abstract Method:

login_menu(): An abstract method to be overridden in subclasses to provide specific login functionality.

manage_attractions Interface

The manage_attractions interface defines methods for managing zoo attractions, including adding, viewing, modifying, removing, and scheduling attractions.

Methods:

add_attraction(): Add a new attraction to the zoo.

view_attraction(): Display a list of all attractions.

modify_attraction(): Modify attraction details.

remove_attraction(): Remove an attraction from the zoo.

schedule_event(): Schedule an event or attraction.

manage_animals Interface

The manage_animals interface defines methods for managing zoo animals, including adding, modifying, viewing, and removing animals.

Methods:

add_animal(): Add a new animal to the zoo.

modify_animal(): Modify animal details.

view_animal(): Display a list of all animals.
remove_animal(): Remove an animal from the zoo.
no_balance Interface
The no_balance interface defines a single method for handling cases when a user has insufficient balance.
Admin Class
The Admin class is a concrete class that extends the Structure class and implements the manage_attractions, manage_animals, and no_balance interfaces. It administers the zoo and manages attractions, animals, discounts, special deals, and visitors.
Attributes:
attraction_list: List of attractions.
input: Instance of the Scanner class for user input.
unique_id: Identifier for attractions.
animal_list: Instance of the Animal_list class for managing animals.
discounts: Discounts for different categories.
specialdeal: Special deals.
visitor_list: Instance of the Visitor_list class for managing visitors.
Methods:
getAnimal_list(): Get the Animal_list.
setVisitorlist(Visitor_list list): Set the visitor list.
getDiscount(): Get the discount map.
getAttractionlist(): Get the list of attractions.
getSpecialDeal(): Get the special deal map.
add_attraction(): Add a new attraction.
not_enough_balance(): Handle cases when a visitor has insufficient balance.
view_attraction(): Display a list of attractions.
modify_attraction(): Modify attraction details.
remove_attraction(): Remove an attraction.
add_animal(): Add a new animal.
modify_animal(): Modify animal details.
view_animal(): Display a list of animals.
remove_animal(): Remove an animal.
schedule_event(): Schedule an event or attraction.
attraction_menu(): Display a menu for managing attractions.
animal_menu(): Display a menu for managing animals.
set_discount_menu(): Allow the admin to set, modify, view, and remove discounts.
Visitor Class
The Visitor class represents a zoo visitor and inherits from the Structure class. It handles visitor interactions in the zoo.
Functionalities:
Exploring the zoo.
Buying memberships.
Buying tickets.
Viewing discounts and special deals.

Visiting animals and attractions.

Leaving feedback.

Viewing the visitor's profile.

Methods:

login_menu(): Provide a menu for visitors to interact with the zoo.

Main Class

The Main class contains the main method to start the zoo management system. It provides options for users to log in as an admin or visitor, view special deals, or exit the program. The class initializes an Admin instance and a Visitor_list.

Assumption:

1) Can type any username and password when login to admin
2) Use small case minor and senior for discount for person less than 18 and greater than 65 age
3) For modify animal you can only modify its noise and description
4) You can also view visitor profile
5) You can set minor senior only once if u add another minor or senior discount code then it will just replace
6) Use proper data variable

Screenshot 1 (top):

```java
/** ANIMAL METHODS STARTS HERE **/
1 usage
@Override
public void add_animal() {
    input.nextLine();
    System.out.print("Enter Animal Name: ");
    String name = input.nextLine();
    System.out.print("Enter Animal Type: ");
    String type = input.nextLine();
    System.out.print("Enter Animal Noise: ");
    String noise = input.nextLine();
    Animal_ds animal = new Animal_ds();
```

Run output:
```
ID: 1|| Attraction Name: jungle safari || Attraction Description: this is jungle|| Attraction Price: 20.0 || Attraction Schedule: CLOSED

Manage Attractions:
1. Add Attraction
2. View Attractions
3. Modify Attraction
4. Remove Attraction
5. Exit

Enter your Choice:
```

Screenshot 2 (bottom):

```java
/** ANIMAL METHODS STARTS HERE **/
1 usage
@Override
public void add_animal() {
    input.nextLine();
    System.out.print("Enter Animal Name: ");
    String name = input.nextLine();
    System.out.print("Enter Animal Type: ");
    String type = input.nextLine();
    System.out.print("Enter Animal Noise: ");
    String noise = input.nextLine();
    Animal_ds animal = new Animal_ds();
    animal.setAnimalName(name);
```

Terminal output:
```
[INFO] Building zootopia 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] ------------------------------[ jar ]-----------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ zootopia ---
[INFO] ------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------
[INFO] Total time:  0.470 s
[INFO] Finished at: 2023-10-27T23:48:00+05:30
[INFO] ------------------------------------------------------------------
PS C:\Users\91807\Desktop\intellij\zootopia>
```

```java
/** ANIMAL METHODS STARTS HERE **/
1 usage
@Override
public void add_animal() {
    input.nextLine();
    System.out.print("Enter Animal Name: ");
    String name = input.nextLine();
    System.out.print("Enter Animal Type: ");
    String type = input.nextLine();
    System.out.print("Enter Animal Noise: ");
    String noise = input.nextLine();
    Animal_ds animal = new Animal_ds();
    animal.setAnimalName(name);
```

Terminal output (top):

```
[INFO] Copying 0 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ zootopia ---
[INFO] Changes detected - recompiling the module! :source
[INFO] Compiling 1 source file with javac [debug target 20] to target\classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.434 s
[INFO] Finished at: 2023-10-27T23:48:15+05:30
[INFO] ------------------------------------------------------------------------
PS C:\Users\91807\Desktop\intellij\zootopia>
```

Terminal output (bottom):

```
[INFO] --- surefire:3.1.2:test (default-test) @ zootopia ---
[INFO] No tests to run.
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ zootopia ---
[INFO] Building jar: C:\Users\91807\Desktop\intellij\zootopia\target\zootopia-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.407 s
[INFO] Finished at: 2023-10-27T23:48:26+05:30
[INFO] ------------------------------------------------------------------------
PS C:\Users\91807\Desktop\intellij\zootopia>
```