**Admin Functionalities**

1. **Enter as Admin**: Admins must provide their username and password to access all admin privileges. A constant is used for the username and password with fixed values for authentication.
2. **Manage Attractions/Events**: Admins can view, add, modify, or remove attractions in the zoo. Each attraction has a unique ID. Attractions can be any cultural exhibit, and they are free of cost for premium members. Basic members need to buy tickets for a fixed price for each attraction.
3. **Schedule Events**: Admins can set the opening and closing times for attractions and specify the ticket price for each attraction. The system also maintains the count of ticketed visitors for each attraction.
4. **Manage Animals**: Admins can add animals to specific attractions, update animal details, and remove animals. Animals can be of three types: Mammals, Amphibians, and Reptiles. There should be at least two kinds of animals for each type.
5. **Set Discounts**: Admins can set discount codes on ticket prices for different visitor categories and attractions. Discounts include:
   * 10% discount for visitors under 18.
   * 20% discount for senior citizens over 60.
6. **Set Special Deals**: Admins can create special deals, including:
   * A 15% discount if a person buys more than 2 attractions.
   * A 30% discount if a person buys more than 3 attractions.
7. **View Visitor Stats**: Admins can access statistics on the number of visitors, revenue, and popular attractions.
8. **View Feedback**: Admins can view feedback left by visitors.

**Visitor Functionalities**

1. **Register**: Visitors can register by providing their name, age, phone number, balance, email, and password.
2. **Log In**: Registered visitors can log in using their email and password to access their account.
3. **Explore the Zoo**: Visitors can browse through different attractions and learn about the animals.
4. **Buy Membership**: Visitors can purchase tickets based on their chosen experience level (Basic or Premium).
5. **Buy Tickets**: Basic users can buy tickets for attractions, and they can also take advantage of special deals. Tickets are one-time usage and become invalid after a visit to the attraction.
6. **View Discounts**: During the ticket purchase, visitors can see all available discounts and select the appropriate discount code.
7. **Apply Discounts**: Visitors can apply available discount codes to their ticket purchase if they meet the eligibility criteria.
8. **Visit Animal**: Visitors can choose to visit an animal and decide whether to feed the animal or read about it. Feeding an animal results in a noise (e.g., a lion roars), while reading provides information about the animal's history.
9. **Visit Attractions/Visit Event**: Visitors can access attractions based on their experience level and purchased tickets. A simple welcome message is displayed to count as a visit to the attraction.
10. **Provide Feedback**: Visitors can leave feedback about their zoo experience and suggest improvements.

**How to Run ZooBuddies**

Create a class hierarchy for Admins, Visitors, Attractions, Animals, Discounts, and Deals, utilizing inheritance and interfaces to capture the functionalities of each.

1. Implement the user-friendly CLI for both admins and visitors.
2. Store data related to attractions, events, visitors, discounts, and deals in data structures such as lists and arraylist.
3. Apply the principles of polymorphism, encapsulation, and interface to ensure modularity and maintainability of the code.
4. Implement user authentication for admins using fixed username and password constants.
5. Ensure proper input validation and error handling to maintain data integrity.

All the commands should be run on the terminal in the HOME\_FOLDER unless otherwise specified.

0) Download the src code folder from Classroom zip file contain another folder named A2\_Zoo\_Management which contains the maven file or src files.

1) mvn clean

2) mvn compile

3) mvn package

4) For <name of the functionality>: java jar <assignment1.jar> <class-name> <args>

5) For <name of the functionality>: java jar <assignment1.jar> <class-name> <args>

6) For <name of the functionality>: java jar <assignment1.jar> <class-name> <args>