# README

**ZOOtopia**

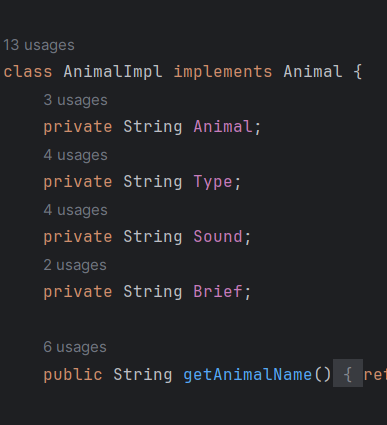
As instructed keeping Object Oriented Programing Principle( such as interfaces, inheritance, abstract classes, and polymorphism).in mind I built an User friendly Program

The required functionality is Implemented & code runs perfectly.

>> Admin ID: abc ; Admin Password: abc1

>> There is two login options one for Admin & other for Visitors ,

For admin I made two classes Main.java & Admin.java, in Admin.java there is one more class named Attraction. In Admin.java I have also created Animal interference and used the inheritance concept of OOPs where I tried to show how the interference methods can be easily implemented in other classes (by class AnimalImpl implements Animal).in my code.



>> In animalImpl class have declared the different field/instances (for animals) which all are

Private and then initialized getter ,setter,Constructor(with fields) later on all the methods

which are required .

>> Inheritance is used to for making sub classes of animal.

>> In Admin class, I have initialized the Array List (like of attraction in which I passed an

attraction object) and different HashMaps for different purposes.

Different methods of admin.

Created a Menu ,..,.. and used method to implement their work

>> Similarly Created the Visitor class where all the fields, arraylist,constructor ,methods are

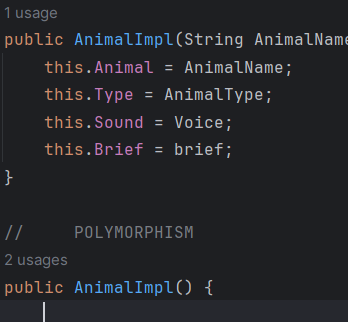
initialized/defined….,

>> Collection Framework is used for sorting purpose (Used collections like isEmpty,

(isEmpty in class AbstractCollection))

>> Used Object class fundamental methods like equals(Object obj)..,toString()

Not much but wherever required.



>> Applied Polymorphism concept wherever needed like in AnimalImpl.. As per requirement.

First we started with the Main.java class which contains login and verification things,after verification we moved to corresponding classes like Admin ,Visitor.

In admin as pe menu provided System.*out*.println("Admin Menu:\n" +

"1. Manage Attractions\n" +

"2. Manage Animals\n" +

"3. Schedule Events\n" +

"4. Set Discounts\n" +

"5. Set Special Deal\n" +

"6. View Visitor Stats\n" +

"7. View Feedback\n" +

"8. Exit\n");

>> In adding attractions assumed Initial ticket count & ticket Price to 0 & -1 respectively .(to be used for membership verification later on)

>> Set the Ticket Price while Scheduling the Attraction(if Ticket Price >0 means it’s scheduled)

>>Asked for Attraction Id while modification to modify attraction details.

>>asked for Attraction Id while adding attraction. (admin)

Discount

>> Asked for the Discount category from the admin while adding a discount.

>> Asked for a Discount Code to modify Discounts.

VisitorStats

>>calculted no. of visitors by finding the length of visitor database list.

>>calculated revenue by subtracting remaining balance from Initial balance(set during

registration)

`

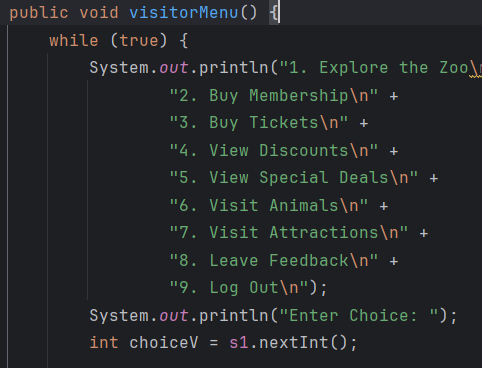
Visitor

Implemented method as per menu provided in Q,used Scanner obj to take Inputs.

>> for visit animal first checked for membership and then allowed accordingly.

>> In Buy membership First asked for type of membership then asked for Discount code and then checked for availabilty of minimum balance required for membership purchase (using if-else ) if condition are fulfilled then deducted the amount from visitor’s balance (after calculating discount ) and set membership Status(0,1 basic ,2 premium)

>>similarly for buying Tickets first checked for member ship status(either 1 or 2) then asked for attraction name ,checked for special deals if applicable (made hashmap for storing Special deals data in format (Min. Ticket required: %off) and then asked for Discount code and calculated remaining balance accordingly.



>> visit attraction

In order to visit attraction first checked for membership and then for tickets of that attraction which we stored in attraction Ticket map.

>>Handled

Note:- In some places due to ambiguity and unclear instructions(assumption based) some codes are in commented part ,these code were written to cover the other possibilities and error .

These code can also be uncommented to handle these cases (checked) to make code Error free.

## How to Run:-

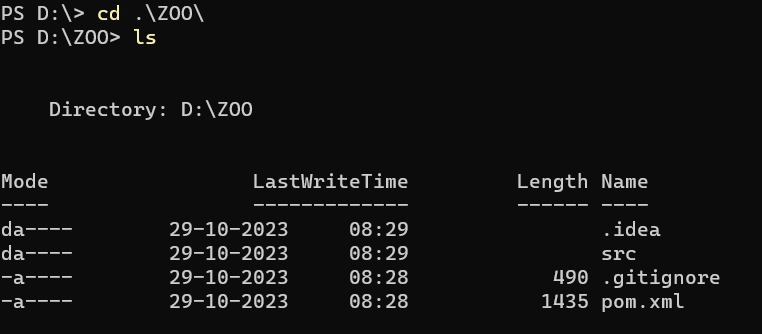
The assignment is done as a Maven Project in Intellij.

“Main.java” is the main Java class created under the package name “org.example” in the src/main/java section

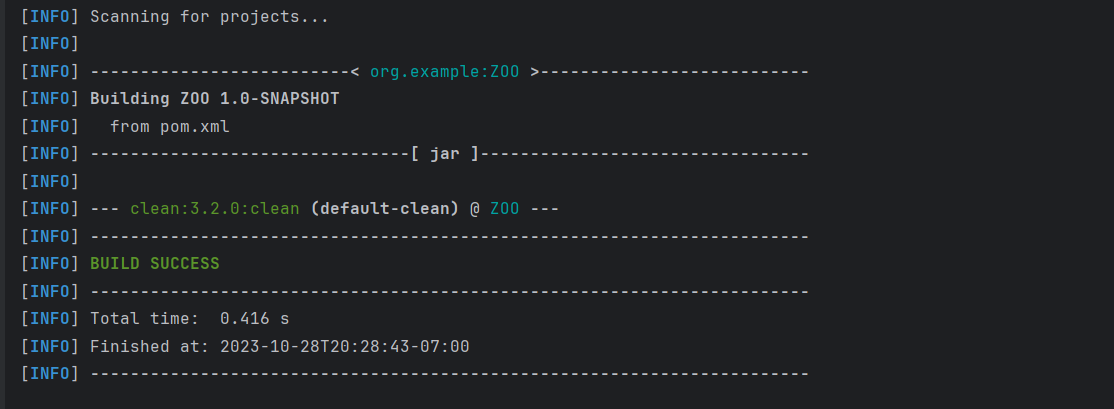
–To execute this program in the terminal:

1. Unzip the 2022125\_Ayush.zip file

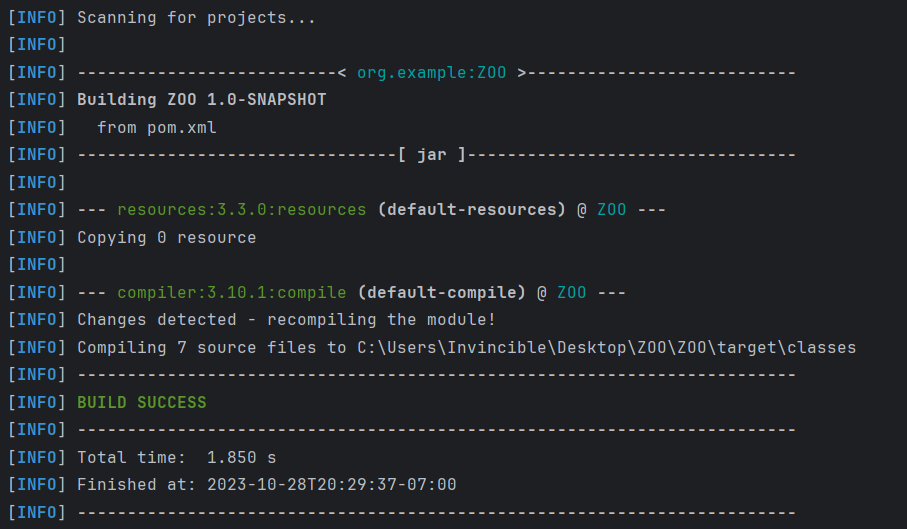
Now open the terminal and go to the ZOO.



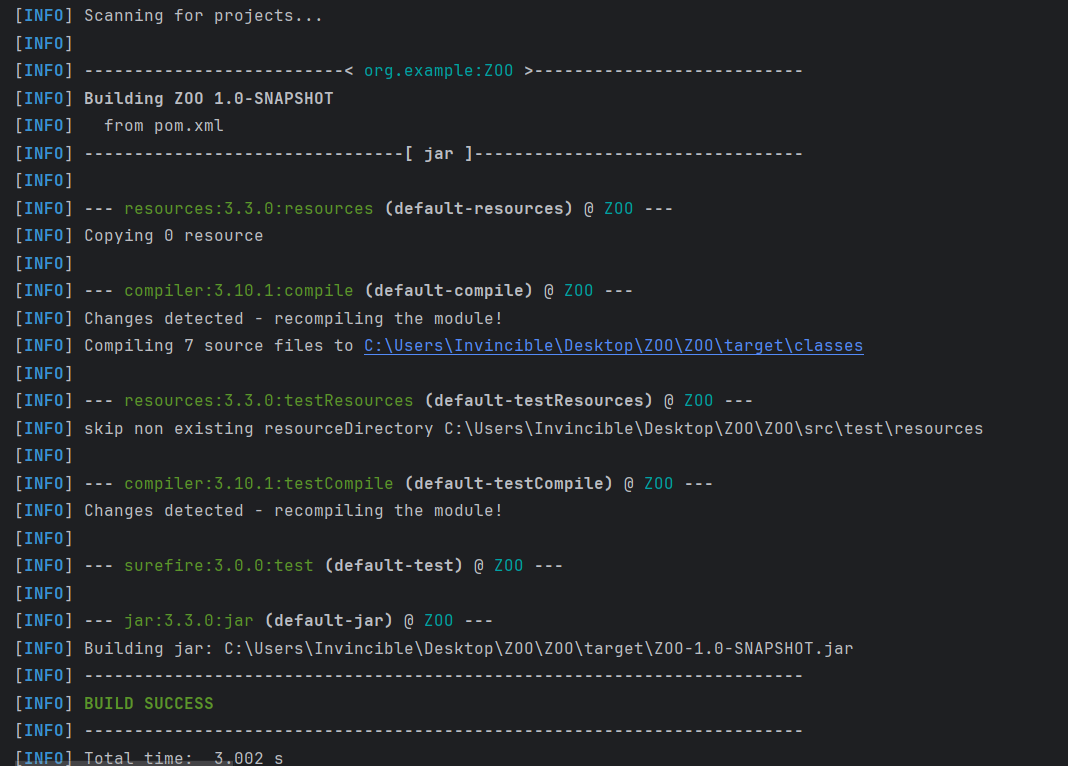
– Enter mvn clean



– Enter mvn compile



– Enter mvn package



now, .jar file has been created to run this file go tho the target folder using ‘cd target’ and type “java -jar <name of jar file that generated> in your terminal

