**Java Project :**

**Event Expenses Calculator using OOPS Concept**

Introduction: In this project we are building a calculator to get the total cost of the event. Here in this application we are collecting the information about the events 1)Exhibition Events and 2)Stage Events.

Calculating cost is key aspect. Additionally to this we have also used different data type such as int,string, double, have also used Private and Protected access modifier. For better object representation we overriding ToString method as well.

**Code used for the Implementation**

**package** projectFloder;

/\*\*

\* In this class, we let users create Events. We have exposed Event constructor to initialize the event object.

\* We have created variables name, type, cost per day and no of days to be captured.

\* This program is for calculating total amount for the event.

\* In this class used protected access modifier.

\* Protected attributes/variables ensure the visibility only by the event class and class inheriting the event class

\* To initialize variables Parameterized constructor have been used.

\*/

**public** **class** Event {

**protected** String name;

**protected** String type;

**protected** Double costPerDay;

**protected** **int** noOfDays;

**public** Event(String name, String type, Double costPerDay, **int** noOfDays) {

**super**();

**this**.name = name;

**this**.type = type;

**this**.costPerDay = costPerDay;

**this**.noOfDays = noOfDays;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getType() {

**return** type;

}

**public** **void** setType(String type) {

**this**.type = type;

}

**public** Double getCostPerDay() {

**return** costPerDay;

}

**public** **void** setCostPerDay(Double costPerDay) {

**this**.costPerDay = costPerDay;

}

**public** **int** getNoOfDays() {

**return** noOfDays;

}

**public** **void** setNoOfDays(**int** noOfDays) {

**this**.noOfDays = noOfDays;

}

@Override

**public** String toString() {

**return** "Event Details\n" + "Name:" + name + "\n" + type;

}

}

**package** projectFloder;

/\*\*

\* In this class, exhibition which extend the event class with 2 more variables i.e Gst and NoofStalls.

\* Gst is the static variable with value 5%.

\* This class also inherits the Event class.

\* Crate a constructor class to initialize variable and also getter setter for all the variables.

\* In that super keyword initialize the variable of event class(parent class).

\* Call the method is to calculate the total amount with 5% Gst.

\* ToString:- This method helps to represent object as String

\*/

**public** **class** Exhibition **extends** Event {

**private** **static** **int** *gst* = 5;

**private** **int** noOfStalls;

**public** Exhibition(String name, String type, Double costPerDay, **int** noOfDays, **int** noOfStalls) {

**super**(name, type, costPerDay, noOfDays);

**this**.noOfStalls = noOfStalls;

}

**public** Double totalcost() {

**double** cost = costPerDay\*noOfDays\*noOfStalls ;

**double** gstcost = (*gst*\*cost)/100;

**return** (cost+gstcost);

}

**public** **int** getNoOfStalls() {

**return** noOfStalls;

}

**public** **void** setNoOfStalls(**int** noOfStalls) {

**this**.noOfStalls = noOfStalls;

}

@Override

**public** String toString() {

**return** **super**.toString() + "\n" + "noOfStalls:" + noOfStalls + "\n" + "total Amount:"

+ String.*format*("%.2f", totalcost());

}

}

**package** projectFloder;

/\*\*

\* This class helps to calculate the total cost of the event based on the cost per day, no of days and gst.

\* Gst is static variable with value 15%.

\* Same as Exhibition class Crate a constructor class to initialize variable and also getter setter for all the variables.

\* This method is to calculate the total amount with 15% Gst.

\* Use override ToString method in all classes to display the event details in the format specified is simple input and output

\*/

**public** **class** StageEvent **extends** Event {

**private** **static** **final** **int** ***CostPerDay*** = 0;

**private** **static** **int** *gst* = 15;

**int** noOfSeats;

**public** StageEvent(String name, String type, Double costPerDay, **int** noOfDays, **int** noOfSeats) {

**super**(name, type, costPerDay, noOfDays);

**this**.noOfSeats = noOfSeats;

}

**public** **static** **int** getGst() {

**return** *gst*;

}

**public** **static** **void** setGst(**int** gst) {

StageEvent.*gst* = gst;

}

**public** **int** getNoOfSeats() {

**return** noOfSeats;

}

**public** **void** setNoOfSeats(**int** noOfSeats) {

**this**.noOfSeats = noOfSeats;

}

**public** **double** totalCost() {

**double** total = costPerDay \* noOfDays;

**double** gstAmount = (total \* *gst*) / 100;

**return** total + gstAmount;

}

@Override

**public** String toString() {

**return** **super**.toString() + "\n" + "NoOfSeats:" + noOfSeats + "\n" + "total Amount:"

+ String.*format*("%.2f", totalCost());

}

}

**package** projectFloder;

**import** java.util.Scanner;

/\*\*

\* The main class with main method in control the flow of execution.

\* This class accepts the input from the user and creates Event, Stage Event and Exhibition

\* Accepts event name,costPerday,noOfDays,from user using scanner class.

\* If type value is 1 Represent Exhibition:- Initialize event object using exhibition class constructor.

\* If type value is 2 Represent Stage Event:- Initialize event object using Stage Event class constructor.

\* Else if value is enter user is other then 1, 2 then show invalid input error massage.

\*/

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter Event name:");

String name = sc.nextLine();

System.***out***.println("Enter cost per day:");

**double** costPerday = sc.nextDouble();

System.***out***.println("Enter no Of days:");

**int** noOfDays = sc.nextInt();

System.***out***.println("Enter the type of event \n 1. Exhibition \n 2. StageEvent");

**int** type = sc.nextInt();

Event event;

**if** (type == 1) {

System.***out***.println("Enter the no of stalls:");

**int** noOfStalls = sc.nextInt();

event = **new** Exhibition(name, "Exhibition", costPerday, noOfDays, noOfStalls);

System.***out***.println(event.toString());

} **else** **if** (type == 2) {

System.***out***.println("Enter the no of seats");

**int** NoOfSeats = sc.nextInt();

event = **new** StageEvent(name, "StageEvent", costPerday, noOfDays, NoOfSeats);

System.***out***.println(event.toString());

} **else** {

System.***out***.println("Invalid input");

}

}

}

This end the project execution flow. Here in a summery we have use Multiple Inheritance, Parameterized Constructor, Access modifier Concepts.