Travel

int travelID,
String Name,
int passengerCapacity,
List<Destination> destinationLIst,
List<Passenger> passengerList,

Destination

int destinationID,
String name,
List<Activity> activityList

Activity

int activityID,
int destinationID,
String name,
String description,
Int cost,
Int activityCapacity,

Enum PassengerType {
    standard,
    gold,
    premium
}

Passenger
    int passengerID,
    PassengerType passengerType,
    String name,
    int balance,

Functional requirement :

    Passenger can sign up for 0 or more activity

    If activityCapacity == 0 , no further subscription

    PassengerType.standard :

        If balance == 0, no further subscription

        has a balance and each time a passenger signs up for an activity cost is deducted from their balance.

PassengerType.gold :

If balance == 0, no further subscription

has a balance and each time a passenger signs up for an activity 10% discount is applied on activity cost and discounted amount deducted from their balance.

PassengerType.premium :

Can sign up for any activity for FREE.

AppRequirement ::

The app should display

1. **Travel package details like**

   Travel package name
   Destinations
   and details of the activities available
   At each destination like name, cost, capacity and description

2. **Passenger list of travel package including**

   Travel package name
   passengerCapacity
   Number of passengers currently enrolled
   Name and passenger number of each passenger

3. **Passenger details like**

   Name
   Passenger number
   Balance if non-premium user

   List of activity signed up for including
   Destination at which activity is taking place
   And the price paid for that activity.

4. Details of each activity that have spaces available
   Including count of spaces

SUB _ TASKS :

Use Java prog lang

High level diagram :

A block diagram indicating interaction between different sub-blocks

Low level diagram :

A UML class diagram of all the classes that needs to Implemented.

The classes must be designed with suitable attributes and methods to represent the requirement asked out.

Implementation :

Java prog lang to be used for implementation

For every class unit test is to be written, JUnit framework is to be used for unit testing the classes

TODO :: Dependency Injection (If time is left)