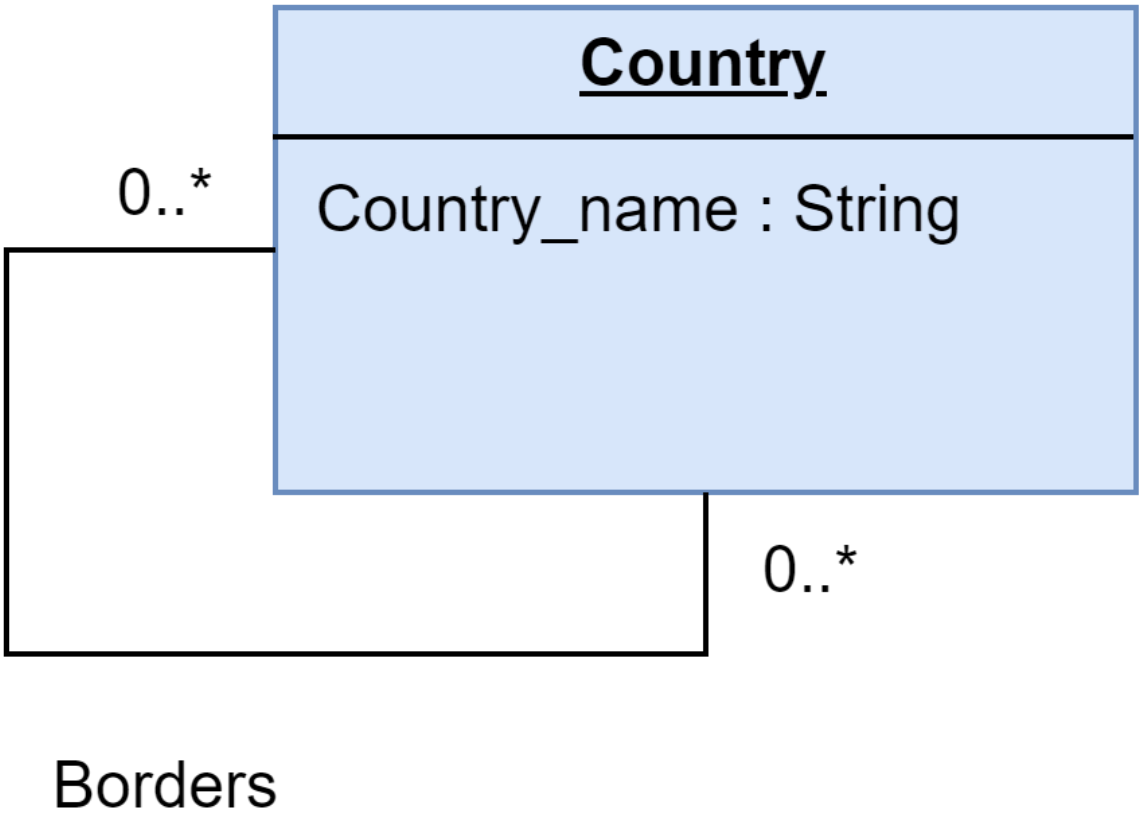
Software Engineering Lab 4

202201106 Vraj Dobariya

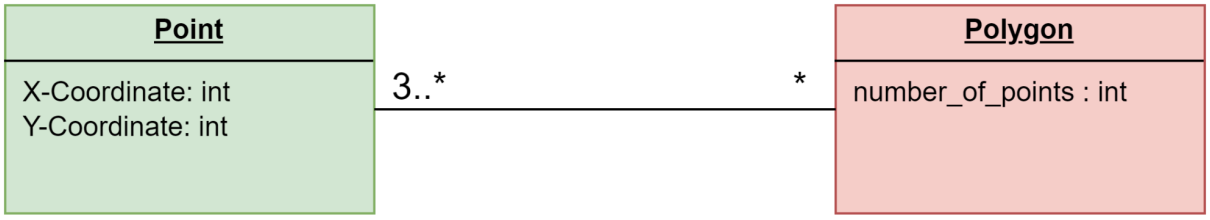
Q.1

Prepare a class diagram for the following object diagram that shows a portion of Europe.



Q-2

Prepare a class diagram for object diagram given in Figure -2. Explain your multiplicity decisions. What is the smallest number of points required to construct a polygon? Does it make a difference whether or not point may be shared between polygons? Your answer should address the fact that points are ordered.

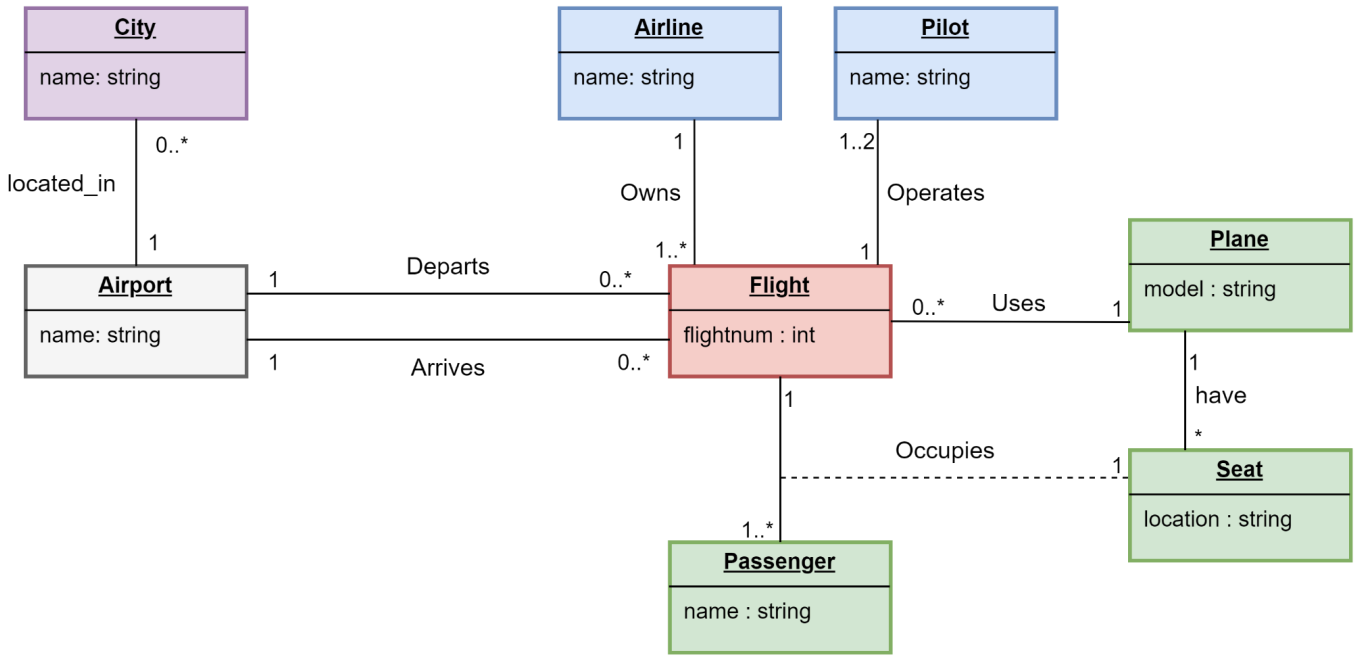


=> Smallest polygon is triangle which is made up of 3 sides and can be described within 3 points so that it is polygon so minimum of points needed.

=> A Common point can be used by multiple polygons but if they use unique points then each polygon can be described uniquely by a set of points there is nothing else to worry about here.

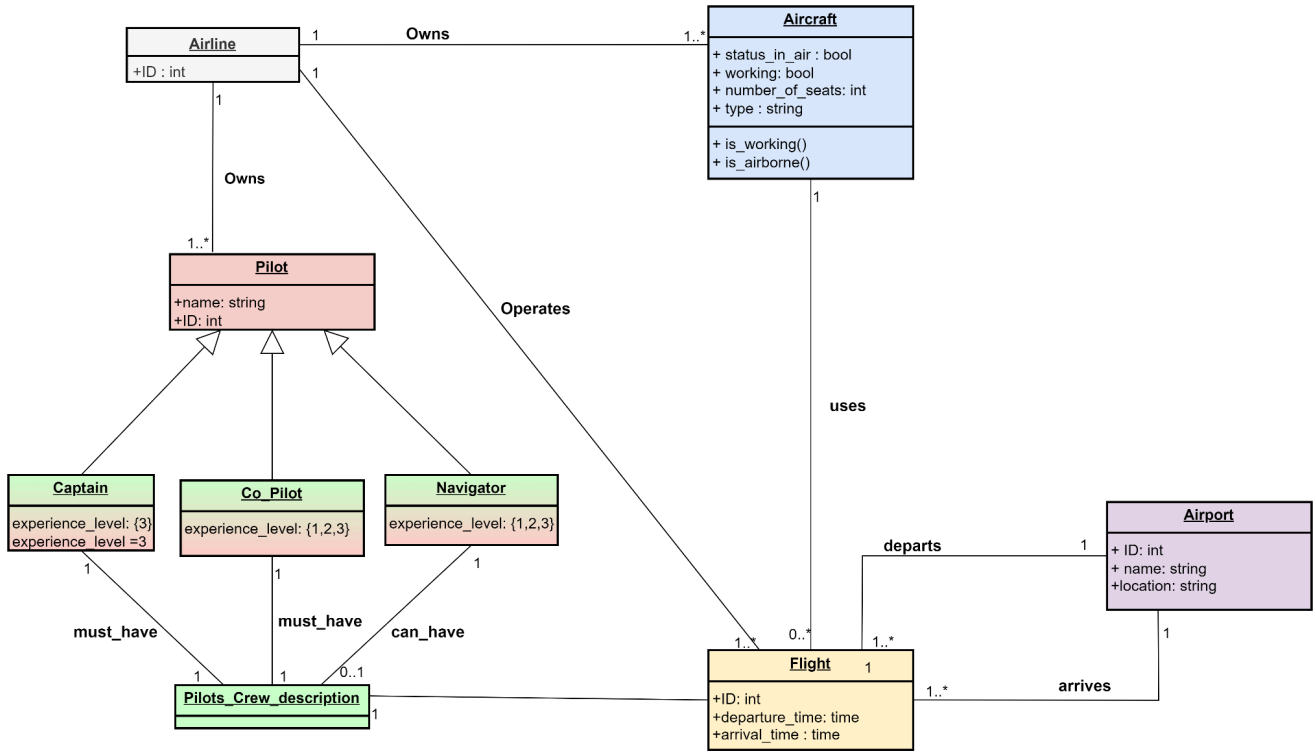
Q-3

Figure 3 is a partially completed class diagram of an air transportation system. Add multiplicities in the diagram. Also add association names to unlevelled associations.



Q-4.

We want to model a system for management of flights and pilots. An airline operates flights. Each the airline has an ID. Each flight has an ID a departure airport and an arrival airport: an airport as a uniqueidentifier. Each flight has a pilot and a co-pilot, and it uses an aircraft of a certain type; a flight has also a departure time and an arrival time. An airline owns a set of aircrafts of different types. An aircraft can be in a working state or it can be under repair. In a particular moment an aircraft can be landed or airborne. A company has a set of pilots: each pilot has an experience level: 1 is minimum, 3 is maximum. A type of airplane may need a particular number of pilots, with a different role (e.g.: captain, co-pilot, navigator): there must be at least one captain and one co-pilot, and a captain must have a level 3.


Airline
+ID : int
Owns
1
1..*
Aircraft
+ status_in_air : bool
+ working: bool
+ number_of_seats: int
+ type : string
+ is_working()
+ is_airborne()
Owns
Pilot
+name: string
+ID: int
Operates
uses
Captain
experience_level: {3}
experience_level =3
Co_Pilot
experience_level: {1,2,3}
Navigator
experience_level: {1,2,3}
must_have
must_have
can_have
Airport
+ ID: int
+ name: string
+location: string
departs
arrives
Pilots_Crew_description
Flight
+ID: int
+departure_time: time
+arrival_time : time
0..1
0..*