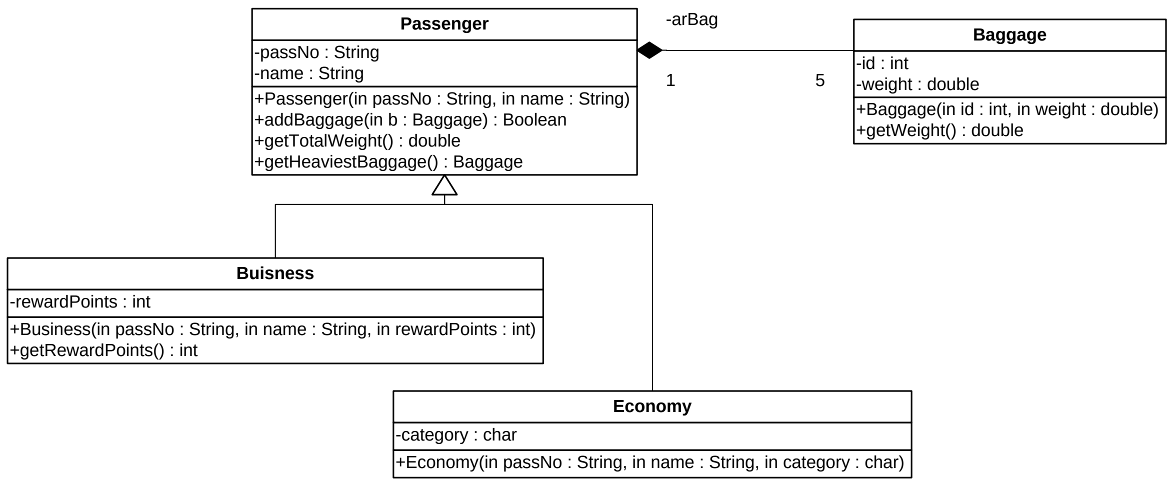
King Saud University
College of Computer and Information Sciences
Department of Computer Science
CSC113 – Computer Programming II – Final Exam – Spring 2016

Exercise1:


Passenger
-passNo : String
-name : String
+Passenger(in passNo : String, in name : String)
+addBaggage(in b : Baggage) : Boolean
+getTotalWeight() : double
+getHeaviestBaggage() : Baggage
-arBag
1
5
Baggage
-id : int
-weight : double
+Baggage(in id : int, in weight : double)
+getWeight() : double
Buisness
-rewardPoints : int
+Business(in passNo : String, in name : String, in rewardPoints : int)
+getRewardPoints() : int
Economy
-category : char
+Economy(in passNo : String, in name : String, in category : char)

Baggage class***:***

- Attributes:
  - ***id***: the id of the baggage item.
  - ***weight:*** the weight of the baggage item.
- Methods:
  - ***Baggage(id: int, weight: double*)**:** constructor.
  - ***getWeight():*** this method returns the weight of the baggage item.

Passenger class***:***

- Attributes:
  - ***passNo***: the passport number of the passenger.
  - ***name:*** the name of the passenger.
- Methods:
  - ***Passenger(passNo: String, name: String*)**:** constructor.
  - ***addBaggage(b: Baggage)*:** this method adds the Baggage ***b*** to the passenger. It returns true if ***b*** is added successfully, and false otherwise.
  - ***getTotalWeight()*:** this method calculates and returns the total weight of all baggage of the passenger.

- ***getHeaviestBaggage():*** this method returns the baggage object that has the maximum weight among all baggage of the passenger.

Business class***:***

- Attributes:
  - ***rewardPoints***: the number of reward points of the business passenger.
- Methods:
  - ***Business (passNo: String, name: String, rewardPoints: int)***: constructor
  - ***getRewardpoints()*****:** this method returns the reward points of the business passenger.

Economy class

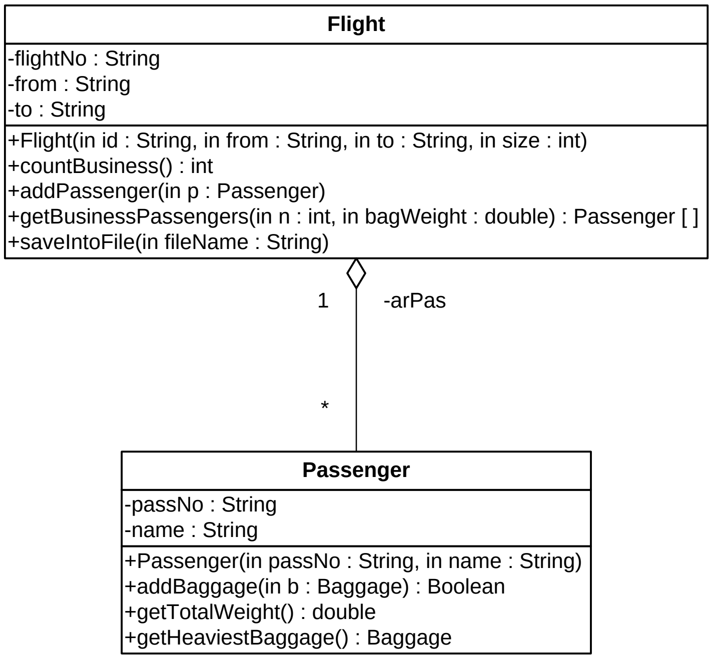
- Attributes:
  - ***category:*** the category of the economy passenger.
- Methods:
  - ***Economy (passNo: String, name: String, category: char)***: constructor

QUESTION: Translate into Java code:

- the class ***Baggage***
- and the class ***Passenger***.

Exercise 2:

Let’s consider the same class ***Passenger*** described in exercise 1.


Flight
-flightNo : String
-from : String
-to : String
+Flight(in id : String, in from : String, in to : String, in size : int)
+countBusiness() : int
+addPassenger(in p : Passenger)
+getBusinessPassengers(in n : int, in bagWeight : double) : Passenger [ ]
+saveIntoFile(in fileName : String)
1
-arPas
*
Passenger
-passNo : String
-name : String
+Passenger(in passNo : String, in name : String)
+addBaggage(in b : Baggage) : Boolean
+getTotalWeight() : double
+getHeaviestBaggage() : Baggage

Flight class***:***

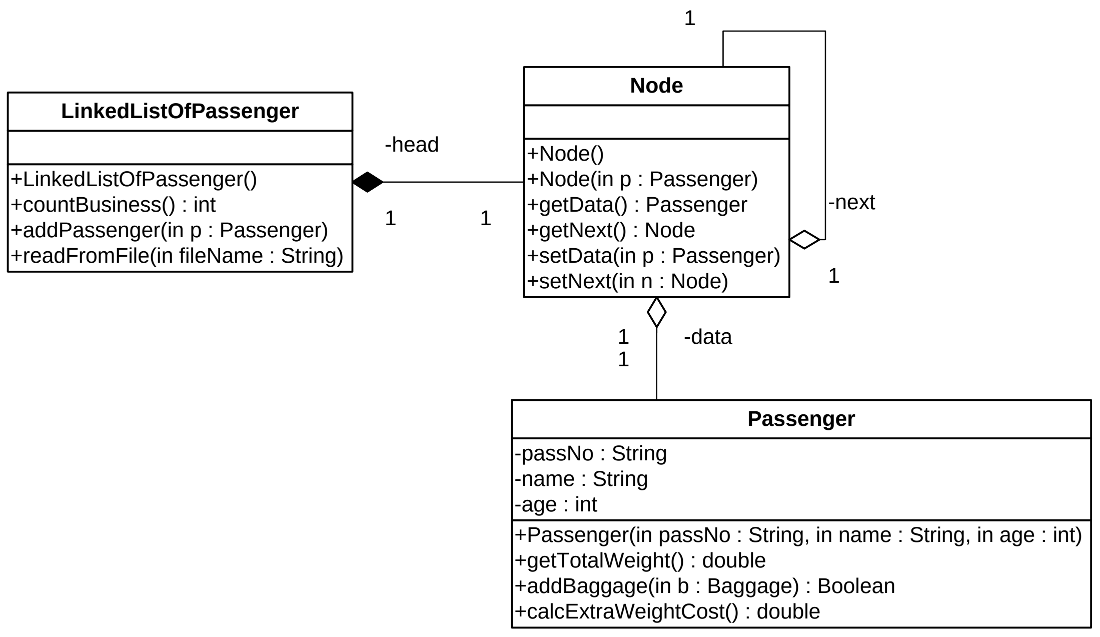
- Attributes:
    - ***flightNo:*** the flight number.
    - ***from***: the name of the departure airport.
    - ***to: the*** name of the arrival airport.
- Methods:
    - ***Flight (id: String, from: String, to: String, size: int)***: constructor. The parameter ***size*** defines the maximum number of passengers in the flight.
    - ***countBusiness ()***: this method returns the number of business passengers in the flight.
    - ***addPassenger (p: Passenger*)**:** this method adds the passenger ***p*** to the flight if possible. There are exactly 10 seats for business passengers on each flight. If adding a passenger is not possible, this method raises an exception with the following message “***No available seats***”.

- ***getBusinessPassengers(n: int, bagWeight: double*)**:** this method returns an array containing all Business passengers having reward points less than ***n***, and total baggage weight exceeding ***bagWeight***.
- ***saveIntoFile(filename: String):*** this method stores all passenger objects of the flight in a file named ***filename***.

QUESTION: Translate into Java code the class ***Flight.***

Exercise 3:

Let's consider the same class ***Passenger*** described in exercise 1.


1
Node
LinkedListOfPassenger
-head
+LinkedListOfPassenger()
+countBusiness() : int
+addPassenger(in p : Passenger)
+readFromFile(in fileName : String)
1
1
+Node()
+Node(in p : Passenger)
+getData() : Passenger
+getNext() : Node
+setData(in p : Passenger)
+setNext(in n : Node)
-next
1
1
1
-data
Passenger
-passNo : String
-name : String
-age : int
+Passenger(in passNo : String, in name : String, in age : int)
+getTotalWeight() : double
+addBaggage(in b : Baggage) : Boolean
+calcExtraWeightCost() : double

LinkedlistOfPassenger class***:***

- Methods:
  - ***LinkedlistOfPassenger()***: constructor.
  - ***countBusiness(***): this method returns the number of business passengers in the list.
  - ***addpassenger(p: Passenger):*** this method inserts the passenger ***p*** at the back of the list.
  - ***readFromFile(filename: String):*** this method reads all passenger objects stored in the file named ***filename*** and adds them to the list.

QUESTION: Translate into Java code the class ***LinkedListOfPassenger.***