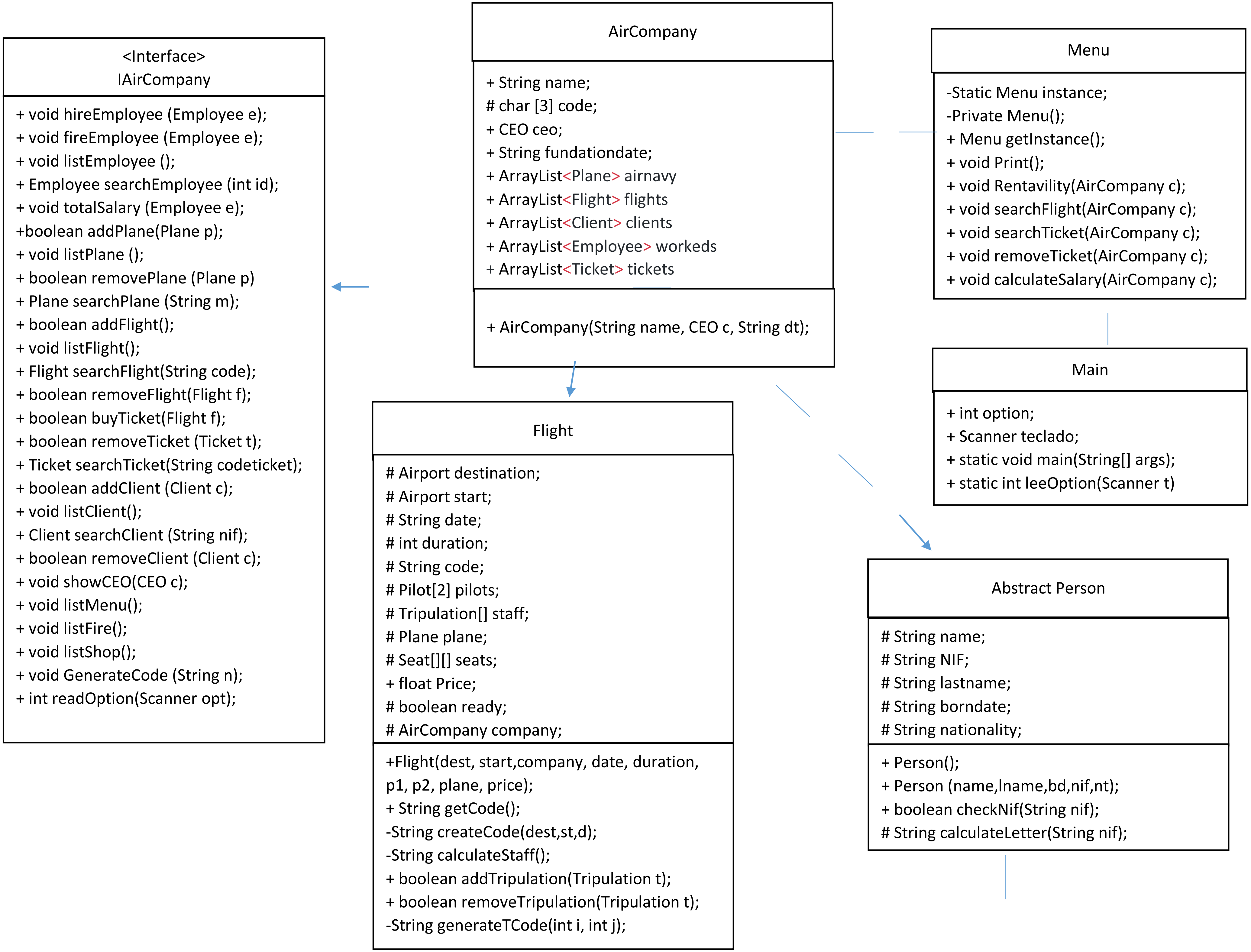


<Interface>
IAirCompany

+ void hireEmployee (Employee e);
+ void fireEmployee (Employee e);
+ void listEmployee ();
+ Employee searchEmployee (int id);
+ void totalSalary (Employee e);
+boolean addPlane(Plane p);
+ void listPlane ();
+ boolean removePlane (Plane p)
+ Plane searchPlane (String m);
+ boolean addFlight();
+ void listFlight();
+ Flight searchFlight(String code);
+ boolean removeFlight(Flight f);
+ boolean buyTicket(Flight f);
+ boolean removeTicket (Ticket t);
+ Ticket searchTicket(String codeticket);
+ boolean addClient (Client c);
+ void listClient();
+ Client searchClient (String nif);
+ boolean removeClient (Client c);
+ void showCEO(CEO c);
+ void listMenu();
+ void listFire();
+ void listShop();
+ void GenerateCode (String n);
+ int readOption(Scanner opt);

AirCompany

+ String name;
# char [3] code;
+ CEO ceo;
+ String fundationdate;
+ ArrayList<Plane> airnavy
+ ArrayList<Flight> flights
+ ArrayList<Client> clients
+ ArrayList<Employee> workeds
+ ArrayList<Ticket> tickets

+ AirCompany(String name, CEO c, String dt);

Menu

-Static Menu instance;
-Private Menu();
+ Menu getInstance();
+ void Print();
+ void Rentavility(AirCompany c);
+ void searchFlight(AirCompany c);
+ void searchTicket(AirCompany c);
+ void removeTicket(AirCompany c);
+ void calculateSalary(AirCompany c);

Main

+ int option;
+ Scanner teclado;
+ static void main(String[] args);
+ static int leeOption(Scanner t)

Flight

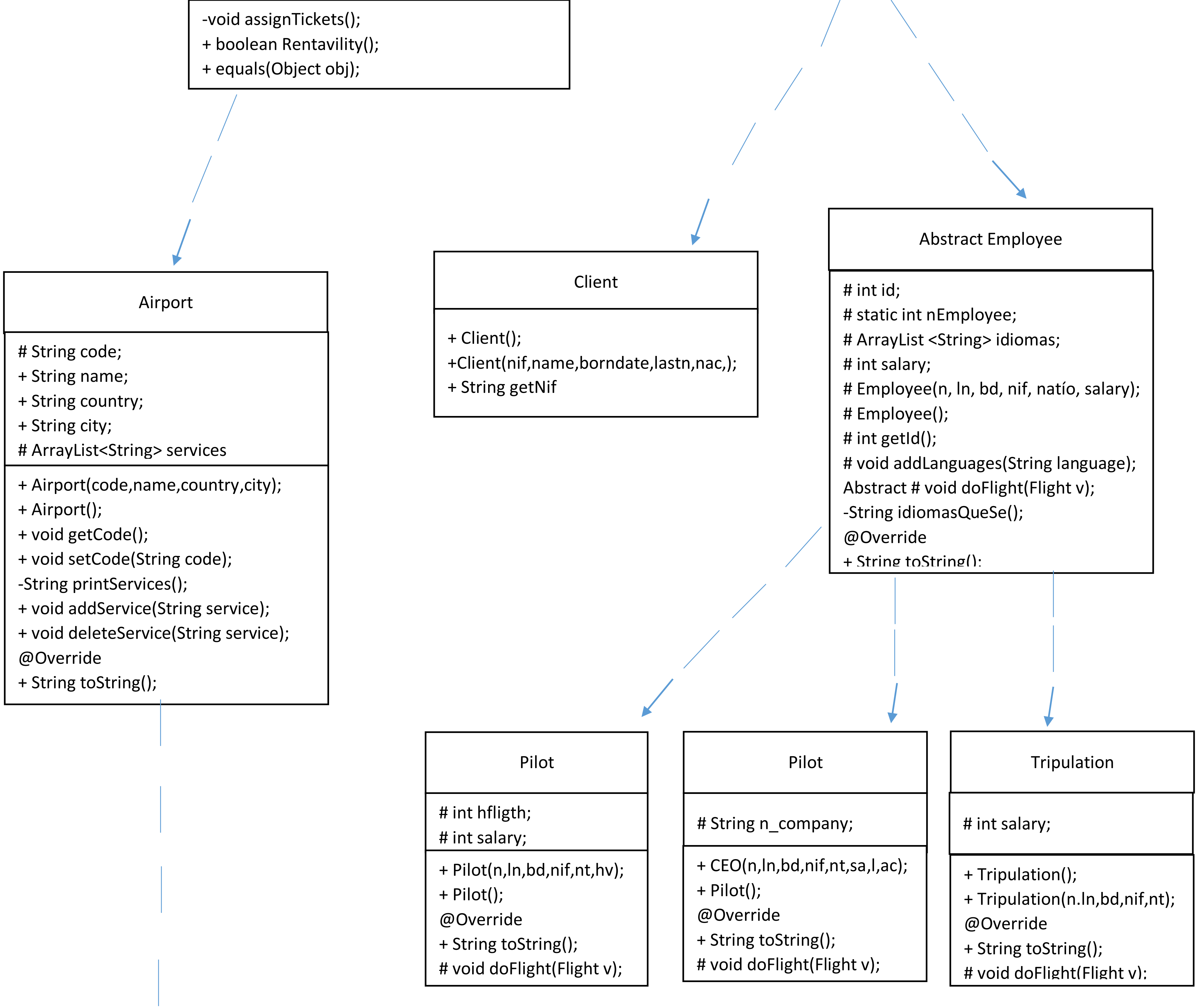
# Airport destination;
# Airport start;
# String date;
# int duration;
# String code;
# Pilot[2] pilots;
# Tripulation[] staff;
# Plane plane;
# Seat[][] seats;
+ float Price;
# boolean ready;
# AirCompany company;

+Flight(dest, start,company, date, duration, p1, p2, plane, price);
+ String getCode();
-String createCode(dest,st,d);
-String calculateStaff();
+ boolean addTripulation(Tripulation t);
+ boolean removeTripulation(Tripulation t);
-String generateTCode(int i, int j);

Abstract Person

# String name;
# String NIF;
# String lastname;
# String borndate;
# String nationality;

+ Person();
+ Person (name,lname,bd,nif,nt);
+ boolean checkNif(String nif);
# String calculateLetter(String nif);



-void assignTickets();
+ boolean Rentavility();
+ equals(Object obj);

Airport

# String code;
+ String name;
+ String country;
+ String city;
# ArrayList<String> services

+ Airport(code,name,country,city);
+ Airport();
+ void getCode();
+ void setCode(String code);
-String printServices();
+ void addService(String service);
+ void deleteService(String service);
@Override
+ String toString();

Client

+ Client();
+Client(nif,name,borndate,lastn,nac,);
+ String getNif

Abstract Employee

# int id;
# static int nEmployee;
# ArrayList <String> idiomas;
# int salary;
# Employee(n, ln, bd, nif, natío, salary);
# Employee();
# int getId();
# void addLanguages(String language);
Abstract # void doFlight(Flight v);
-String idiomasQueSe();
@Override
+ String toString();

Pilot

# int hfligth;
# int salary;

+ Pilot(n,ln,bd,nif,nt,hv);
+ Pilot();
@Override
+ String toString();
# void doFlight(Flight v);

Pilot

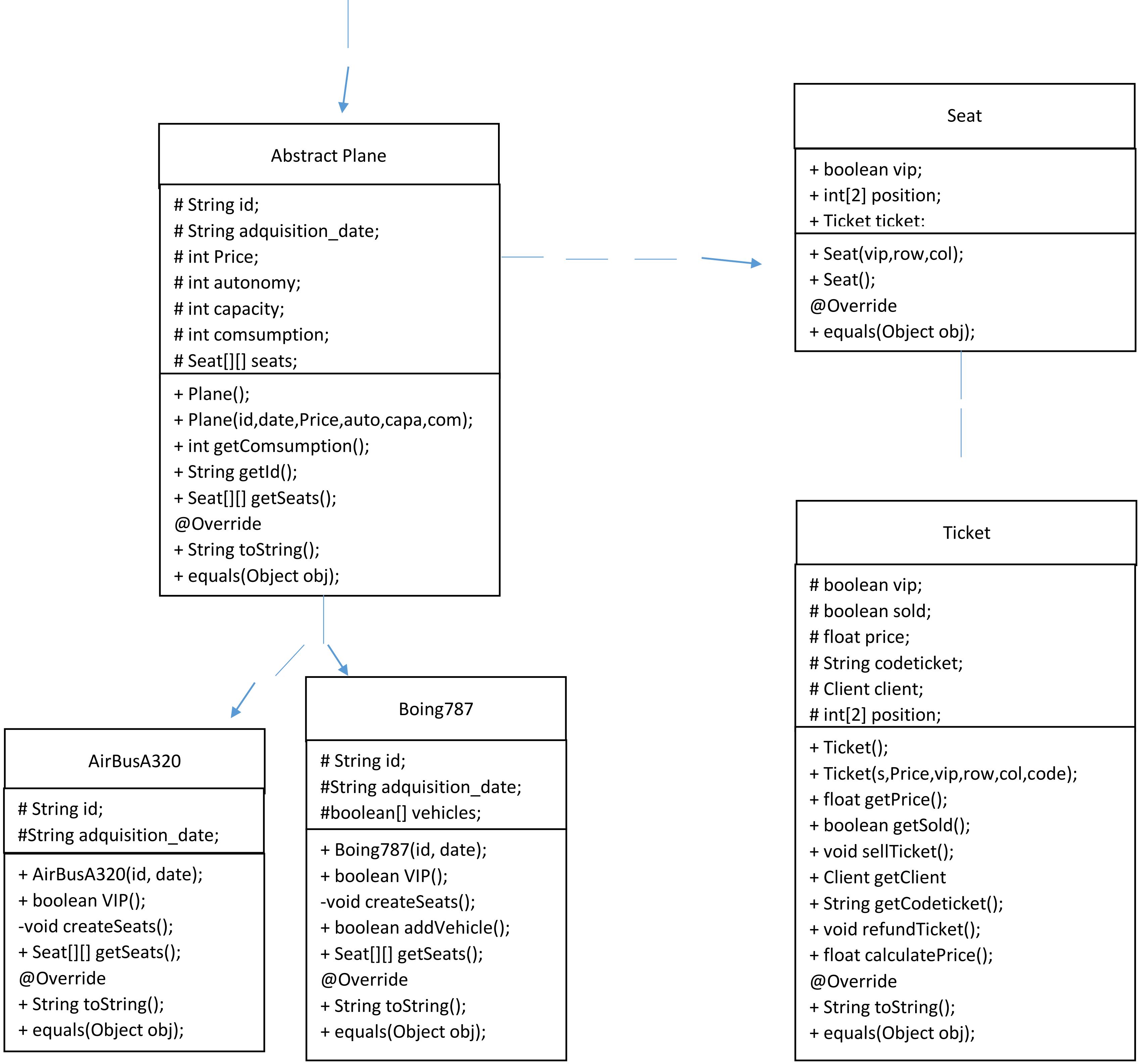
# String n_company;

+ CEO(n,ln,bd,nif,nt,sa,l,ac);
+ Pilot();
@Override
+ String toString();
# void doFlight(Flight v);

Tripulation

# int salary;

+ Tripulation();
+ Tripulation(n.ln,bd,nif,nt);
@Override
+ String toString();
# void doFlight(Flight v);



Abstract Plane

# String id;
# String adquisition_date;
# int Price;
# int autonomy;
# int capacity;
# int comsumption;
# Seat[][] seats;

+ Plane();
+ Plane(id,date,Price,auto,capa,com);
+ int getComsumption();
+ String getId();
+ Seat[][] getSeats();
@Override
+ String toString();
+ equals(Object obj);

Seat

+ boolean vip;
+ int[2] position;
+ Ticket ticket;

+ Seat(vip,row,col);
+ Seat();
@Override
+ equals(Object obj);

AirBusA320

# String id;
#String adquisition_date;

+ AirBusA320(id, date);
+ boolean VIP();
-void createSeats();
+ Seat[][] getSeats();
@Override
+ String toString();
+ equals(Object obj);

Boing787

# String id;
#String adquisition_date;
#boolean[] vehicles;

+ Boing787(id, date);
+ boolean VIP();
-void createSeats();
+ boolean addVehicle();
+ Seat[][] getSeats();
@Override
+ String toString();
+ equals(Object obj);

Ticket

# boolean vip;
# boolean sold;
# float price;
# String codeticket;
# Client client;
# int[2] position;

+ Ticket();
+ Ticket(s,Price,vip,row,col,code);
+ float getPrice();
+ boolean getSold();
+ void sellTicket();
+ Client getClient
+ String getCodeticket();
+ void refundTicket();
+ float calculatePrice();
@Override
+ String toString();
+ equals(Object obj);