Inheritance - Assignment 1

We need to instantiate inside a Java main method several planes belonging to the fleet of a country. The fleet contains passenger planes as well as fighter planes.

There are two types of passenger planes: *Boeing* or *Concorde*. Fighter planes can be *MiG* or *TomCat*.

Each type of plane is modelled through a class and the actual planes will be instances of those classes. Each plane can perform a specific range of operations and procedures, as specified below. It is required to "build" more planes of various types and execute all the operations provided by each airplane.

Each plane must contain an unique private member planeID (e.g. 1, 2, 3, 4). The ID of a plane is implicitly set when the plane is instantiated and its value is the ID of the previous plane plus one. The ID of the first plane is 1.

Each plane must contain an int denoting the power of its engine. This value is set in the constructor of the plane that receives this value via a parameter.

Each plane provides the following services:

public String getPlaneID() - returns the ID of the plane
public int getTotalEnginePower() - returns the power of the engine
public void takeOff() - prints "takeOff" on the screen
public void land() - prints "land" on the screen
public void fly() - prints "fly" on the screen

Only passenger planes contain a member maxPassengers that is set in the constructor of the class and a method public int getMaxPassengers() to return the value of this member.

Concorde passenger planes are supersonic and, consequently, the next two services are provided by this plane:

public void goSuperSonic() - prints "Supersonic mode activated" on the screen
public void goSubSonic() - prints "Supersonic mode deactivated" on the screen

Fighter planes have the capacity to launch rockets on different targets, so for every fighter we must be able to call the method:

public void launchMissile() which prints "Launching rocket" on the screen.

MIG planes have variable geometry for normal and high-speed flight. The class that models this plane have the next methods:

public void highSpeedGeometry() - prints "High speed selected geometry" on the screen

public void and normalGeometry() - prints "Normal selected geometry" on the screen

TomCat planes can be refuelled when flighting, so for such planes it makes sense to call a method public void refuel() which prints on the screen "TomCat - Refuelling".

Each method that prints a message on the screen must display also the ID of the plane.
Create inside a main method more variables that have the same type and refer to all existing planes.