**The FidoFintessClub Member Management System** v.1.1



1. The club needs a program that can be used to register the members and their dog. They want to store the **members**’ **names**, **phone numbers**, and **email addresses**. Also, *each member* has a **dog**. For *each dog*, they want to be able to record the dog’s **name** and the year, the owner had paid a membership fee.

Also, they want to be able to determine for each member whether he had paid the fee for his dog, and whether he currently has a dog. For this, you must implement two methods: **isFeeDue(thisYear:int):boolean** and **hasDog():bool**ean. Also, implement the **describe():void** method that prints the member’s name, phone and email.  
You can use today’s slides as a starting point.

* 1. Draw a class diagram showing all the fields and methods you must implement. If you have already installed UMLet, use it, or just use pen and paper.
  2. Implement the two classes Dog and Member.
  3. Test it creating three members: Joe, Jane and Jean. Joe and Jane have dogs, but Jean has just lost his dog. Fill in all the fields. For each of them, call the isFeeDue(), the hasDog() and the describe() methods.
  4. Do you get an exception in some cases? Fix your code so it doesn’t crash. You may need to check for null pointers before calling a method on your instance variables (fields).

1. We want the dog’s name to be shown after the member’s name, phone, and email in his or her description.
   1. Extend your **describe():void** method to include the dog’s name in the member’s description. If the member has no dog, the method should print “no dog” instead of crashing (with a NullPointerException).
   2. Test it with Joe, Jane, and Jean.