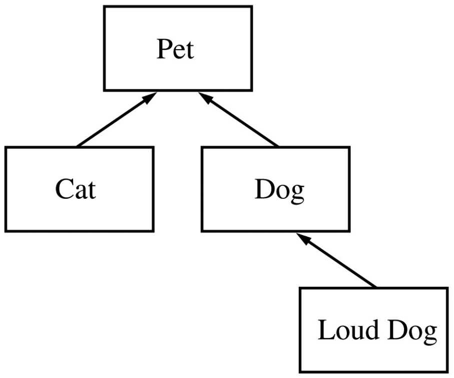
3. Consider the hierarchy of classes shown in the following diagram.



Note that a `Cat` "is-a" `Pet`, a `Dog` "is-a" `Pet`, and a `LoudDog` "is-a" `Dog`.

The class `Pet` is specified as an abstract class as shown in the following declaration. Each `Pet` has a name that is specified when it is constructed.

```
public abstract class Pet
{
  private String name;

  public Pet(String petName)
  {  name = petName;  }

  public String getName()
  {  return name;  }

  public abstract String speak();
}
```

The subclass `Dog` has the partial class declaration shown below.

```
public class Dog extends Pet
{
  public Dog(String petName)
  {   /* implementation not shown */   }

  public String speak()
  {   /* implementation not shown */   }
}
```

(a) Given the class hierarchy shown above, write a complete class declaration for the class `Cat`, including implementations of its constructor and method(s). The `Cat` method `speak` returns `"meow"` when it is invoked.

(b) Assume that class `Dog` has been declared as shown at the beginning of the question. If the `String` *dog-sound* is returned by the `Dog` method `speak`, then the `LoudDog` method speak returns a `String` containing *dog-sound* repeated two times.

Given the class hierarchy shown previously, write a complete class declaration for the class `LoudDog`, including implementations of its constructor and method(s).

(c) Consider the following partial declaration of class `Kennel`.

```
public class Kennel
{
  private List<Pet> petList;


  /** For every Pet in the kennel, prints the name followed by
   *  the result of a call to its speak method, one line per Pet.
   */
  public void allSpeak()
  {  /* to be implemented in part (c) */  }

  // There may be instance variables, constructors, and methods that are
  // not shown.
}
```

Write the `Kennel` method `allSpeak`. For each `Pet` in the kennel, `allSpeak` prints a line with the name of the `Pet` followed by the result of a call to its `speak` method.

In writing `allSpeak`, you may use any of the methods defined for any of the classes specified for this problem. Assume that these methods work as specified, regardless of what you wrote in parts (a) and (b).

Complete method `allSpeak` below.

```
/** For each Pet in the kennel, prints the name followed by
 *  the result of a call to its speak method, one line per Pet.
 */
public void allSpeak()
```