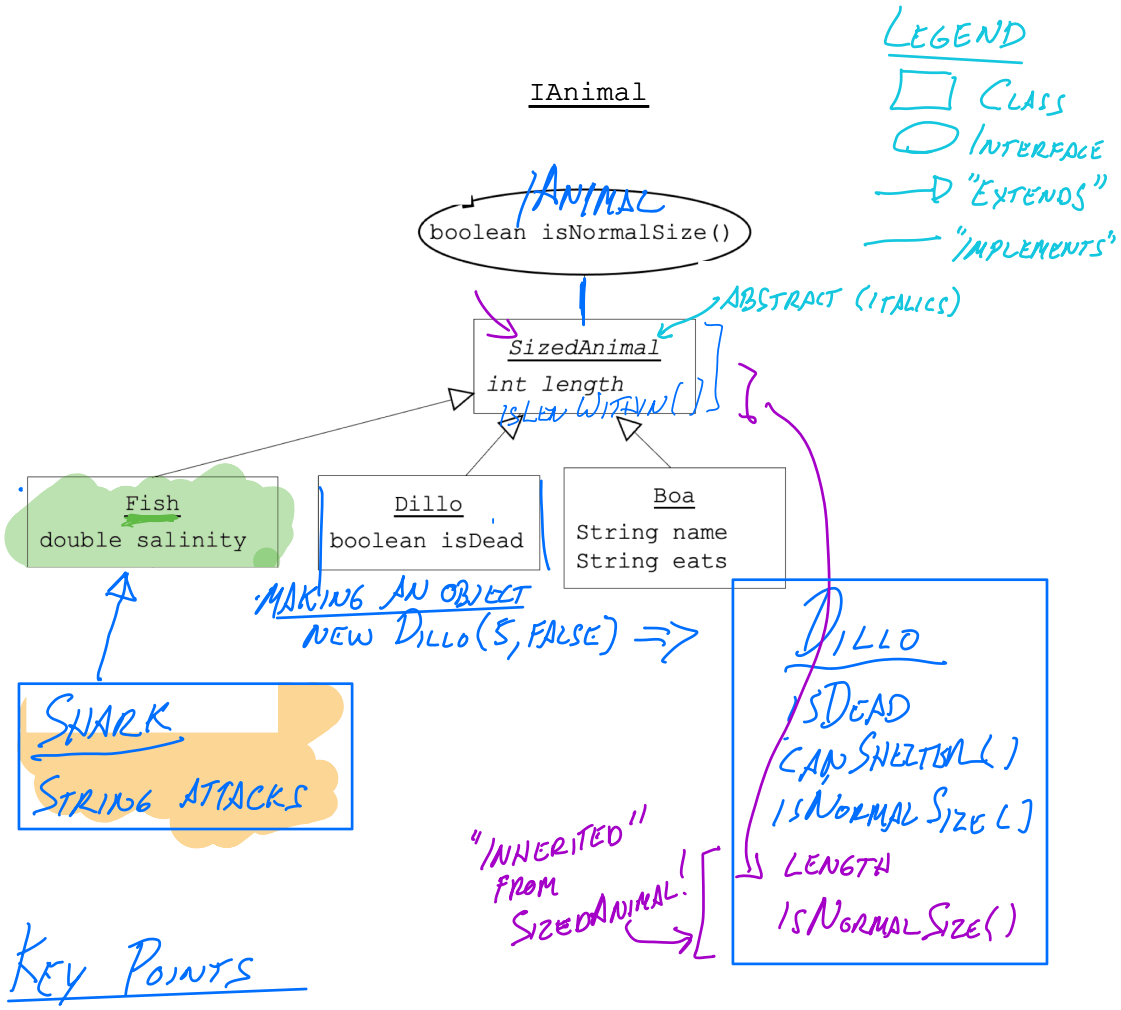
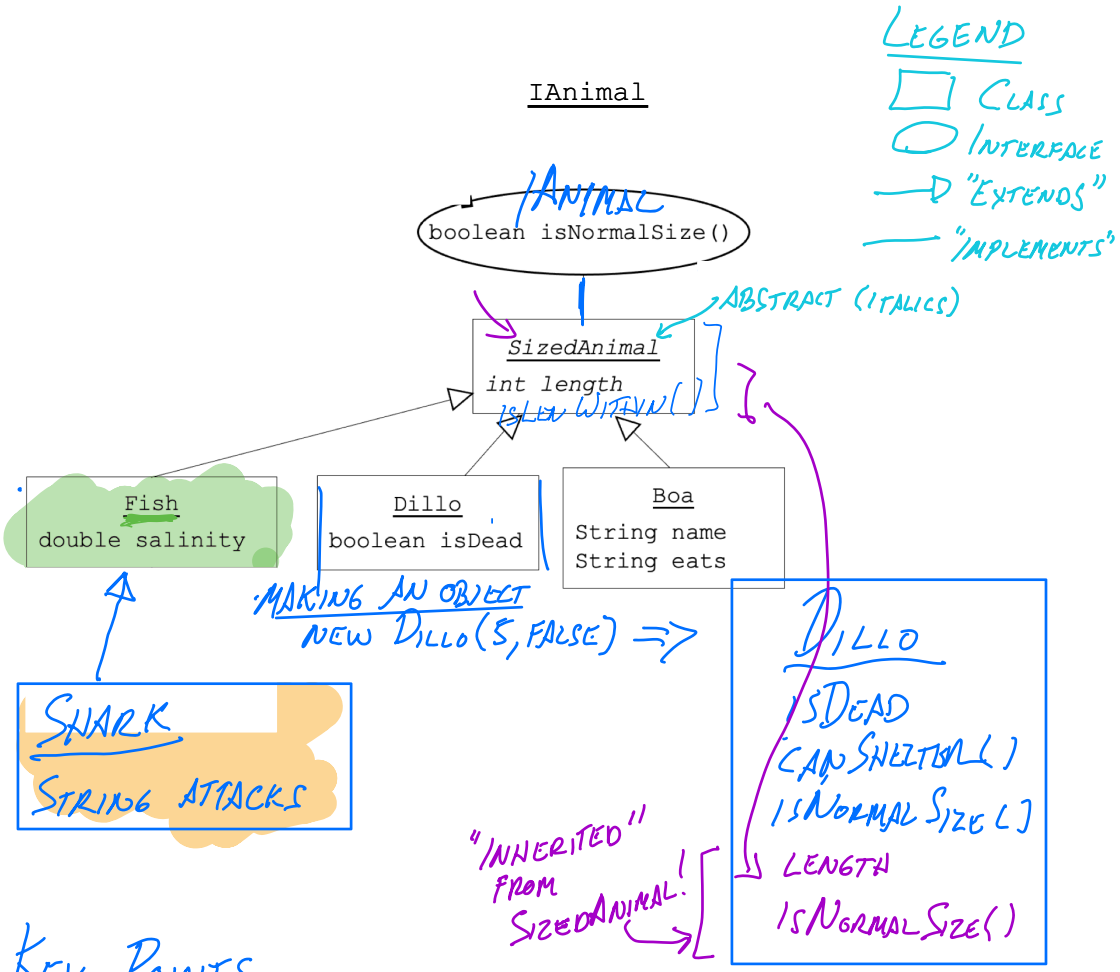


- Interfaces define behavior required for a class ("All IAnimals must have isNormalSize() method"

- Classes are built in hierarchies, fields/methods are "copied down" or inherited in subclass from its superclass

- "abstract" disallows "new" → CAN'T DO NEW SIZEDANIMAL(5)
  => Use this when making a class that's only for code sharing!
  (Doesn't make sense to have a SizedAnimal, it's not a specific animal)

- "implements" is also "copied down when inheriting from a class. Boa, Dillo, etc. all inherit the requirement for isNormalSize()



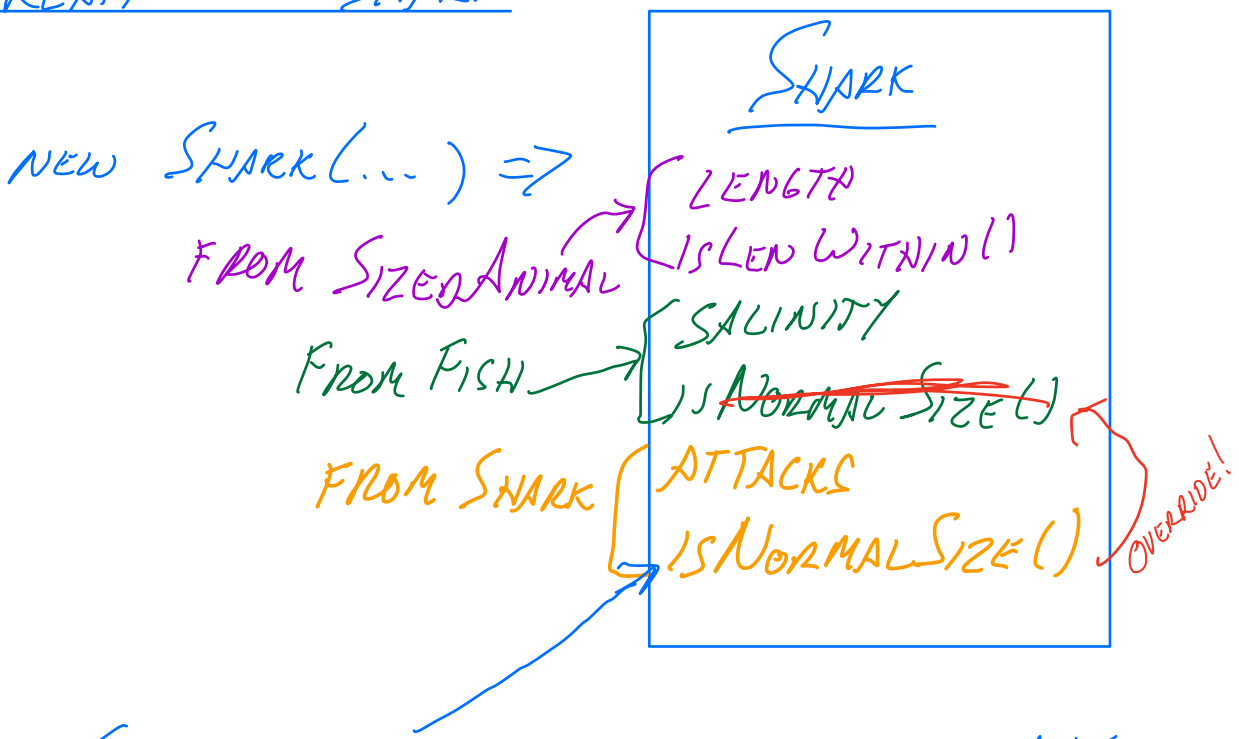
Key Points

- CLASSES ARE BUILT IN HIERARCHIES, FIELDS/METHODS GET "COPIED DOWN" ("INHERITED") FROM SUBCLASS FROM SUPER CLASS.
- "ABSTRACT" DISALLOWS "NEW" => USE WHEN MAKING A CLASS THAT'S ONLY FOR CODE SHARING!

(IT DOESN'T MAKE SENSE TO HAVE A SIZED ANIMAL IN OUR ZOO — NOT A SPECIFIC ANIMAL)

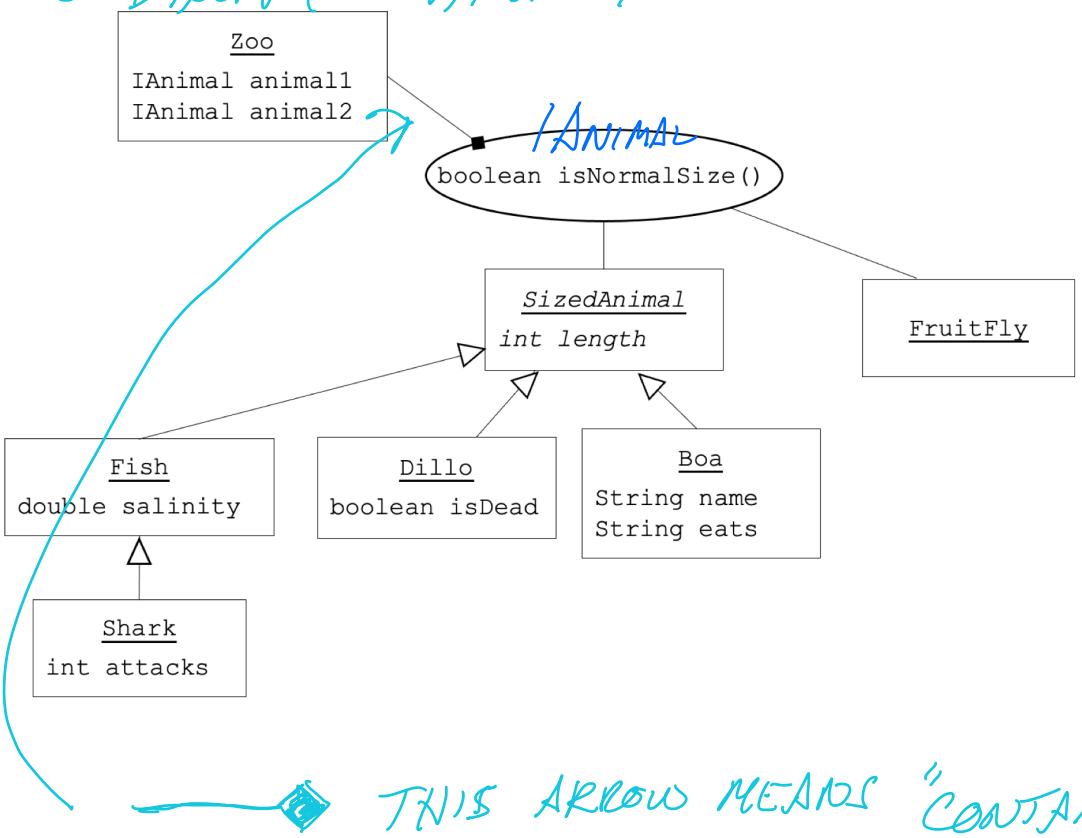
- "IMPLEMENTS" IS ALSO "COPIED DOWN" WHEN INHERITING FROM A CLASS. (EG. BOA, DILLO, FISH MUST ALL IMPLEMENT isNormalSize() FROM IANIMAL.

CREATING A SHARK



- SHARK IS NOT REQUIRED TO HAVE ITS OWN ISNORMALSIZE() BECAUSE FISH HAS ONE.
  BUT, WE ADD AN ISNORMALSIZE() TO HAVE A SHARK-SPECIFIC VERSION.
  IF WE DO THIS IT "OVERRIDES" THE VERSION IN FISH.

MORE DIAGRAM NOTATION YOU'LL SEE.



THIS ARROW MEANS "CONTAINS"
IE. Zoo CONTAINS FIELDS OF CLASS IANIMAL.

UPDATING OBJECTS/FIELDS

```
public class Boa {
    public string name;
    public int length;
    public string eats;

    public Boa (String name,
                int length,
                String eats) {
       this.name = name ;
       this.length = length ;
       this.eats = eats ;
     }
 }
```

IN THIS EXAMPLE ...

WHAT IS BOA1.EATS??

```
1 Boa boa1 = new Boa("slinky", 30, "apples");
2 Boa boa2 = new Boa("slim", 30, "bugs");

3 boa1.eats = "tofu";
4 boa2 = new Boa("slim", 15, "grass");

5 Boa boa3 = boa1;
6 boa3.eats = "donuts";
```

ENVIRONMENT (NAMES)

① BOA1

② BOA2

④ BOA2 NOW REFERS TO NEW OBJECT! (OLD ONE IS CLEANED UP)

BOA3

⑤ THE NAME "BOA3" NOW REFERS TO THE SAME OBJECT AS THE NAME "BOA1"

HEAP (OBJECTS)

BOA
"SLINKY"
30
"~~APPLES~~"

③ FIELD UPDATES TO "~~TOFU~~" "DONUTS"

⑥ UPDATE THE FIELD NAME OF THIS OBJECT!

BOA
"SLIM"
15
"BUGS"

BOA
"SLIM"
15
"GRASS"

END RESULT

PRINT(BOA1.EATS) => "DONUTS"
PRINT(BOA3.EATS) => "DONUTS"

THIS IS A VERY COMMON MISCONCEPTION! OKAY IF IT SEEMS WEIRD NOW.