

## 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!
  - "IMPLEMENTS" IS ALSO "COPIED DOWN" WHEN INHERITING FROM A CLASS. (EG. BOA, DILLO, FISH MUST ALL IMPLEMENT isNormalSize() FROM ANIMAL.)
- IT DOESN'T MAKE SENSE TO HAVE A SIZED ANIMAL IN OUR ZOO - NOT A SPECIFIC ANIMAL

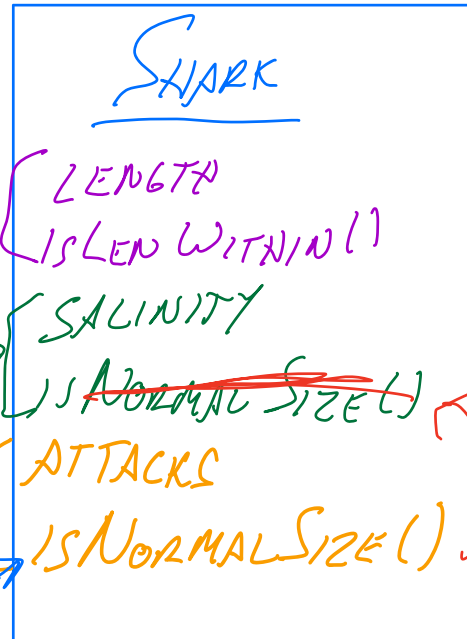
## CREATING A SHARK

NEW SHARK(...) =>

FROM SIZEDANIMAL

FROM FISH

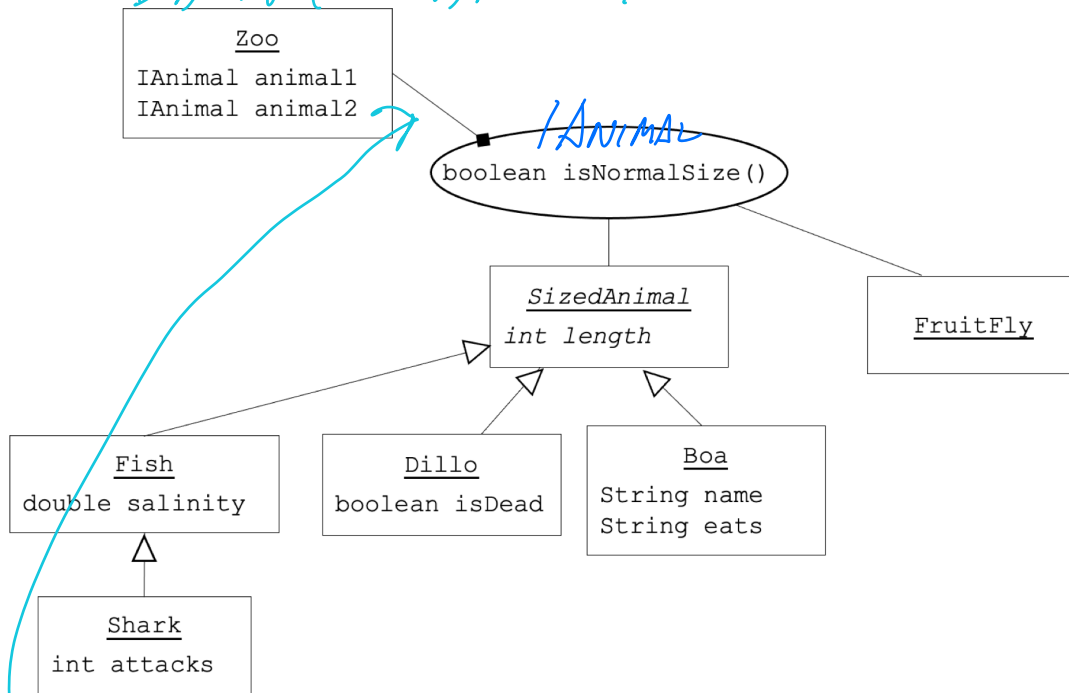
FROM SHARK



- 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

IN THIS EXAMPLE...

WHAT IS

BOA1. EATS??

```
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 ;  
    }  
}
```

```
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)

HEAP (OBJECTS)

1 BOA1

Boa  
"SLINKY"  
30  
~~"APPLES"~~

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

2 BOA2

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

Boa  
"SLIM"  
15  
"BUGS"

(6) UPDATE THE  
FIELD NAME OF  
THIS OBJECT!

3 BOA3

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

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.