

```
interface IAnimal {
    boolean isNormalSize();
    boolean isDangerToPeople();
}

abstract class AbsAnimal implements IAnimal {
    int length;
    AbsAnimal (int length) {...}

    boolean isLenWithin(int low, int high)
        {...}

    public boolean isDangerToPeople() {
        return false;
    }
}
```

```
class Fish extends AbsAnimal {
    double salinity;

    Fish(int length, double salinity)
        {...}

    public boolean isNormalSize() {
        return is LenWithin(1, 15);
    }
}

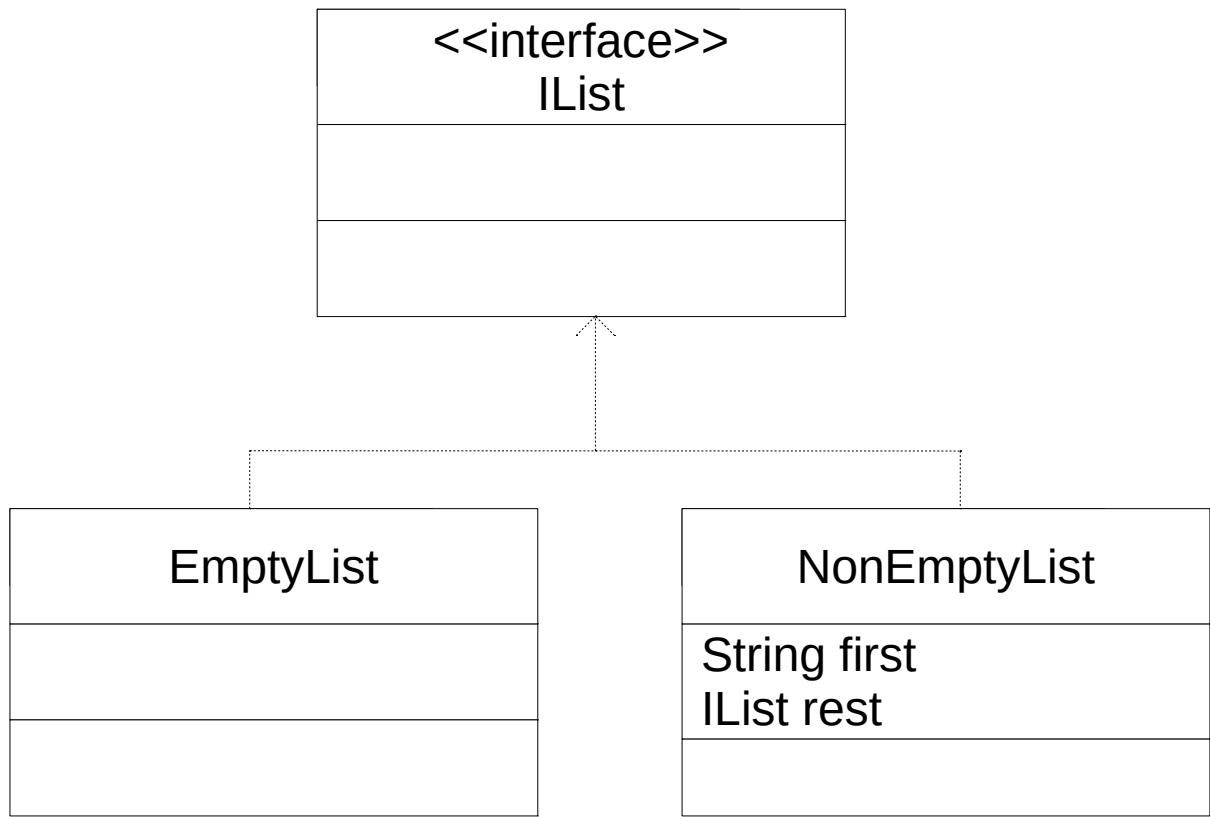
class Shark extends Fish {
    int attacks;

    Shark (int length, int attacks) {
        super(length, 3.75);
        this.attacks = attacks;
    }

    public boolean isNormalSize() {
        return this.length > 6;
    }

    public boolean isDangerToPeople() {
        return true;
    }
}
```

```
; a ListOfString is one of  
;   empty  
;   (cons String ListOfString)
```



KNOWN CLASSES

OBJECTS

EmptyList

NonEmptyList

"pears"

NonEmptyList

"apples"

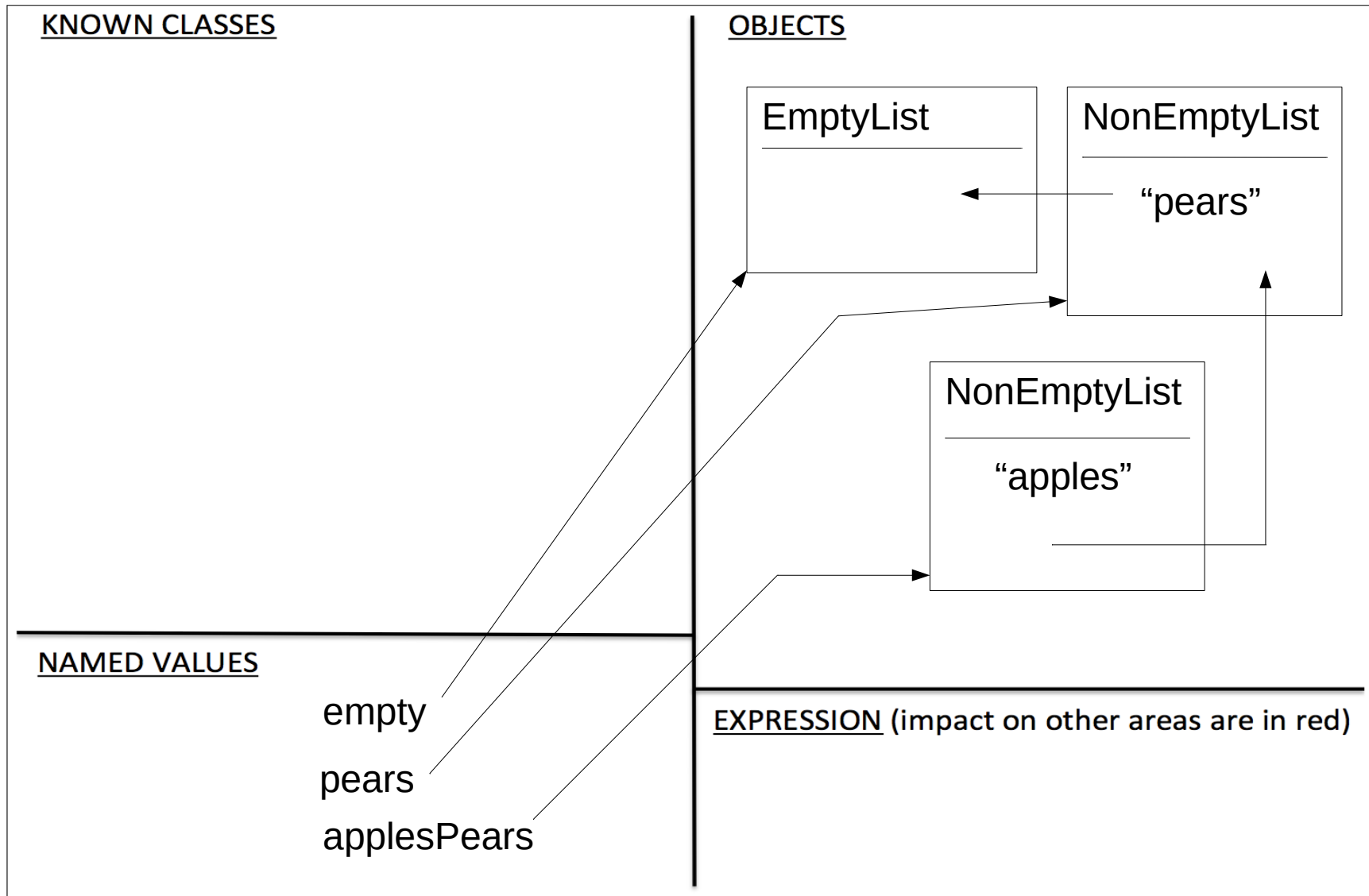
NAMED VALUES

empty

pears

applesPears

EXPRESSION (impact on other areas are in red)



KNOWN CLASSES

OBJECTS

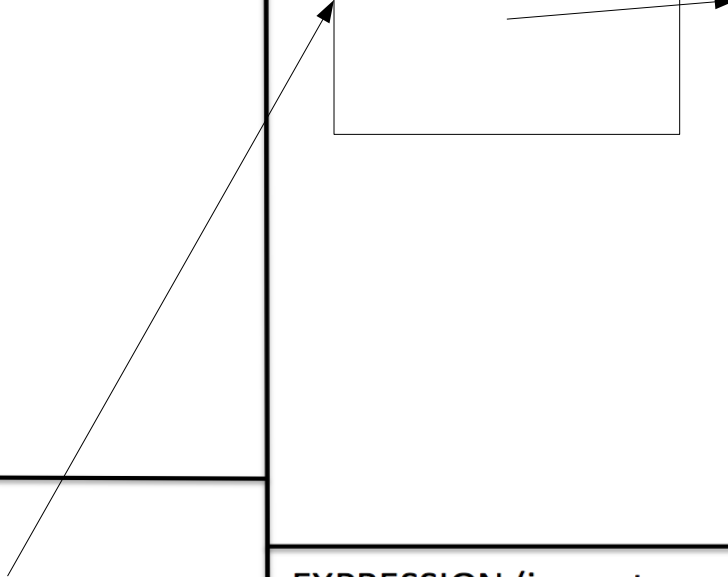
WordList

LinkedList

NAMED VALUES

fruitBasket

EXPRESSION (impact on other areas are in red)



```
for (type var: specific-list) {  
    statement(s);  
}
```