

# + Birds

- is Extinct : boolean      - is Near Water : boolean

- number Of Wings : int

- number Of Characteristic : int

- number Of Food : int

+ get Is Extinct ( ) : boolean

+ set Is Extinct ( )

+ get Number Of Wings ( ) : int

+ set Number Of Wings ( )

+ get Number Of Characteristic ( ) : int

+ get Num Of Food ( ) : int

+ get Is Near Water ( ) : boolean



(For isNearWater == 1)  
e.g. Shorebirds or Waterfowl

## + Waterfowl

- characteristic = Character [ ]

- Food : Food [ ]

- nameOf Water : Water [ ]

+ get Characteristic ( ) : Character [ ]

+ print Characteristic ( )

+ get Food ( ) : Food [ ]

+ print Food ( )

+ get NameOf Water ( ) : Water [ ]

(For isNearWater == 0)

## + Flightless

- characteristic : Character [ ]

- Food : Food [ ]

+ get Characteristic ( ) : Character [ ]

+ print Characteristic ( )

+ get Food ( ) : Food [ ]

+ print Food ( )



(For all types)

## + EMUS

- type Name : String

+ get Name ( ) : String

## + Parrot

- characteristic : Character [ ]

- Food : Food [ ]

- num Of Words : int

- Words : String [ ]

+ get Characteristic ( ) : Character [ ]

+ get Food ( ) : Food [ ]

+ get Num Of Words ( ) : int

+ get Words ( ) : String [ ]

+ ~~num~~ mimic Sound ( )

Northeastern University  
San Francisco

Northeastern University  
Silicon Valley

NORTHEASTERN.EDU/BAYAREA/

+ enum = Food
---------------

+ Seeds
---------

+ Fruits
----------

...
-----

+ enum = Water
----------------

+ fresh
---------

+ salt
--------

Northeastern University  
**San Francisco**

Northeastern University  
**Silicon Valley**

# Design of test plan

## Criterion and Strategy:

To convince someone else my test code work correctly,  
I should test all methods and consider as many special cases  
as possible (For instance: 0, +, -)

Before the test, set up the constructive.

For the most commonly used methods. 'get' and 'set'

1st step: test whether the results meet expectations.

e.g. `getNumOfWings()`

it should return a integer value 2 for my testing data,  
compare with the practical returned result. (int 2)

2nd step. test special cases.

String: `Null`, `" "`, `"ab"`

Int: `0`, `-1`, `+1`, `1.1`

Comparison: `a > 0`, `a < 0`, `a ≤ 0`, `a = 0`, ...

type in enum, try all values

More details need to be analyzed  
for more complicated cases (exceptions)  
(`print`, `mimicSound`)

**Northeastern University  
San Francisco**

**Northeastern University  
Silicon Valley**

**NORTHEASTERN.EDU/BAYAREA/**