Inheritance

**DEADLINE:** 07/10/2018

## FOLDER STRUCTURE

FL\_9\_4\_homework\_inheritance/\*

homework/\*

.eslintrc.js\*

index.html\*

js/\*

app.js\*

## TASK

1. Create your own **assign** method (see **Code** section for invocation example).
2. Write your custom movement simulator (see **Code** section for invocation example).  
     
   Create **Bot**, **Racebot** and **Speedbot**.

All classes have similar attributes such as **name**, **speed** and **x, y** coordinates, which they accept on instantiation from a config object.

**If you need other attributes - you can add them.**

These attributes should be accessed by get/set functions (for example, to access **speed**, you need to have **getSpeed** and **setSpeed** functions)

Both classes should have **at least** following methods:

* 1. getSpeed - returns **speed**.
  2. setSpeed - sets **speed**.
  3. getDefaultSpeed - returns **default speed** (passed to function on initialization).
  4. getCoordinates - returns object with 2 keys: **x** and **y**.
  5. setCoordinates - sets **x** and **y**.
  6. move – updates bot’s position. Can accept 1 of 4 possible directions (up, down, left, right – use basic coordination system) If direction is specified badly – don’t move and return error (can be simple console.log).
  7. showPosition –outputs bot’s position(simple console.log).

**If you need other methods - you can add them.**

**Racebot** class details:

If **Racebot’s** previous move was the same as current – his **speed** increases by 1. (if he moves the same direction few times, speed gets increased on each step).

If **Racebot’s** previous move was different – his **speed** is reduced to **default speed**.

**Speedbot** class details:

**Speedbot** should have method called **prepareEngine**, which increases **speed** by 2. After each move, **speed** gets decremented by 1 until it matches **default speed**. (for example, if **default speed** is equal to 3, on first step **speed** will be 5, on second step – 4, on third step – 3, on fourth step - 3).

Effect is stackable.

## RESTRICTIONS

* Do not use ES6 classes
* Do not use any external libraries

## CODE

//Task 1

var defaults = { a: 123, b: 777 };

var options = { a: 456 };

var configs = assign({}, defaults, options); // {a: 456, b: 777}

//Task 2

let Botty = new Bot({name: "Betty", speed: 2, x: 0, y: 1});

Botty.showPosition(); // I am Bot 'Betty'. I am located at 0:1.

Botty.move('up');

Botty.showPosition(); // I am Bot 'Betty'. I am located at 0:3.

Botty.move('left');

Botty.move('down');

Botty.move('up');

Botty.move('up');

Botty.showPosition(); // I am Bot 'Betty'. I am located at -2:5.

Botty.move('up');

Botty.showPosition(); // I am Bot 'Betty'. I am located at -2:7.

Botty.move('up');

Botty.showPosition(); // I am Bot 'Betty'. I am located at -2:9.

let Zoom = new Racebot({name: "Lightning", speed: 2, x: 0, y: 1});

Zoom.showPosition(); // I am Racebot 'Lightning'. I am located at 0:1.

Zoom.move('up');

Zoom.showPosition(); // I am Racebot 'Lightning'. I am located at 0:3.

Zoom.move('left');

Zoom.move('down');

Zoom.move('up');

Zoom.move('up');

Zoom.showPosition(); // I am Racebot 'Lightning'. I am located at -2:6.

Zoom.move('up');

Zoom.showPosition(); // I am Racebot 'Lightning'. I am located at -2:10.

Zoom.move('up');

Zoom.showPosition(); // I am Racebot 'Lightning'. I am located at -2:15.

let Broom = new Speedbot({name: "Thunder", speed: 2, x: 0, y: 1});

Broom.showPosition(); // I am Speedbot 'Thunder'. I am located at 0:1.

Broom.move('up');

Broom.showPosition(); // I am Speedbot 'Thunder'. I am located at 0:3.

Broom.prepareEngine();

Broom.move('left');

Broom.move('down');

Broom.move('up');

Broom.move('up');

Broom.showPosition(); // I am Speedbot 'Thunder'. I am located at -4:4.

Broom.move('up');

Broom.showPosition(); // I am Speedbot 'Thunder'. I am located at -4:6.

Broom.move('up');

Broom.showPosition(); // I am Speedbot 'Thunder'. I am located at -4:8.

## BEFORE SUBMIT

* Check if invocation examples are working with your code;
* Check file structure;
* Check code structure (remove unnecessary code/comments).

## SUBMIT

* The folder should be uploaded to github repository 'fl-9' into master branch.

## USEFUL LINKS

* <http://eloquentjavascript.net/06_object.html>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Inheritance_and_the_prototype_chain>