```
classdef parameterDictionary

    %parameterDictionary Define all variable input parameters here
 including
    %their datatypes, allowed, and current values.
    %   This function serves as both a text and programmatic reference
 to all
    %   parameters to the model.  Every parameter given should have a
 match
    %   here.  The defaults defined here should be rarely changed and
    %   considered code modifications.  Any other change should be
 applied from
    %   the outside by a script or GUI.
    %
    %   Since this is just a wrapper around a Map, subclassing
 containers.Map
    %   seems like the right thing to do, but it's a known problem
 that this
    %   doesn't work well.


  %{"RCP":"rcp85","OA":1,"E":true,"everyx":1,"useTestThreads":5,"doProgressBar":tr
[],
    %
 "superStart":2035,"superMode":0,"superAdvantage":0,"newMortYears":false,"doCoralC

    properties
        params
    end

    methods (Access = 'private')
        function obj = addOne(obj, p)
            obj.params(char(p.name)) = p;
        end
    end


ans =

  parameterDictionary with properties:

    params: [11×1 containers.Map]


    methods
        function p = parameterDictionary()
            p.params = containers.Map;

            addOne(p, modelCharParameter('RCP', 'string', 'rcp8.5',
 {'rcp2.6', 'rcp4.5', 'rcp6.0', 'rcp 8.5'}));
            addOne(p, modelLogicalParameter('OA', 'logical', false));
            addOne(p, modelLogicalParameter('E', 'logical', false));
            % TODO everyx can be  a string or integer in V11!
 Redefine it?
```

```
            addOne(p, modelIntParameter('everyx', 'integer', 1, 1,
1925));
            pc = parcluster('local');
            maxW = pc.NumWorkers;
            addOne(p, modelIntParameter('useThreads', 'integer', 2, 1,
maxW));
            addOne(p, modelIntParameter('keyReefs', 'integer', 5, 1,
1925));

            addOne(p, modelIntParameter('superStart', 'integer', 2035,
1861, 2100));
            addOne(p, modelIntParameter('superMode', 'integer', 0, 0,
6));
            addOne(p, modelDoubleParameter('superAdvantage', 'double',
0.0, 0.0, 10.0));
            addOne(p, modelLogicalParameter('newMortYears', 'logical',
false));
            addOne(p,
modelLogicalParameter('doCoralCoverFigure', 'logical', true));

        end

        % Set an existing value.  New values may not be added "on the
fly".
        function obj = set(obj, name, value)
            % MATLAB will raise an error if there's not existing value
called
            % name.
            p = obj.params(name);
            p.set(value);
        end
    end

end
```

Published with MATLAB® R2017b