Boris Bikes

v.2.0

**Key:** Ruby RSpec

**Setting Up Project**

1. Create domain model
2. Set up folder structure
3. Git init

**Building the Bike**

1. Bike should not be broken
2. Make Bike class
3. Def broken?
4. It should be able to break
5. Def break!
6. Def initialize, and make broken? and break! Work
7. Bike should be able to get fixed
8. Def fix!
9. Git commit

**Refactoring**

1. Call fix! In initialize
2. Add let() helper to create new bike
3. Git commit

**The Docking Station (D/S)**

1. Set up Docking Station files
2. D/S should accept a bike
3. Make DockingStation Class
4. Def bike\_count
5. Set bike\_count to 0
6. Def initialize and dock, and make bike\_count work
7. Make dock method work
8. D/S should release a bike
9. Refactor with let() statements
10. Def release
11. D/S should know when it’s full
12. Pass the capacity as a parameter to the initialiser
13. Fix the problem the test has uncovered
14. Define the default capacity at top of DockingStation class
15. Def full?
16. Make full? Work
17. D/S should not accept a bike if it’s full (use lambda)
18. Update dock() method
19. Refactor – extract repetition to a helper method (inside D/S block)
20. D/S should provide the list of available bikes
21. Def available\_bikes
22. Git commit

**Van & Garage**

1. Extract common functionality from DockingStation to BikeContainer
2. Create lib/bike\_container.rb
3. Extract all methods from DockingStation to BikeContainer module
4. Set default capacity to 10
5. Create accessor methods for @capacity and @bikes
6. Use ||= operators to initialise values
7. Refactor DockingStation
8. Create artificial class to include BikeContainer
9. BikeContainer should accept a bike
10. DockingStation should now only have one test to check initializer
11. Git commit and push to Github
12. Create Van class that reuses BikeContainer
13. Create Garage class that reuses BikeContainer – must fix bikes
14. Git commit
15. Push to Github