Table of Contents

Problem #1: Even or Odd

Using TDD, write a program to check if a number is even or odd

Function definition:

- `function isEven(n) {}`

Function requirements:

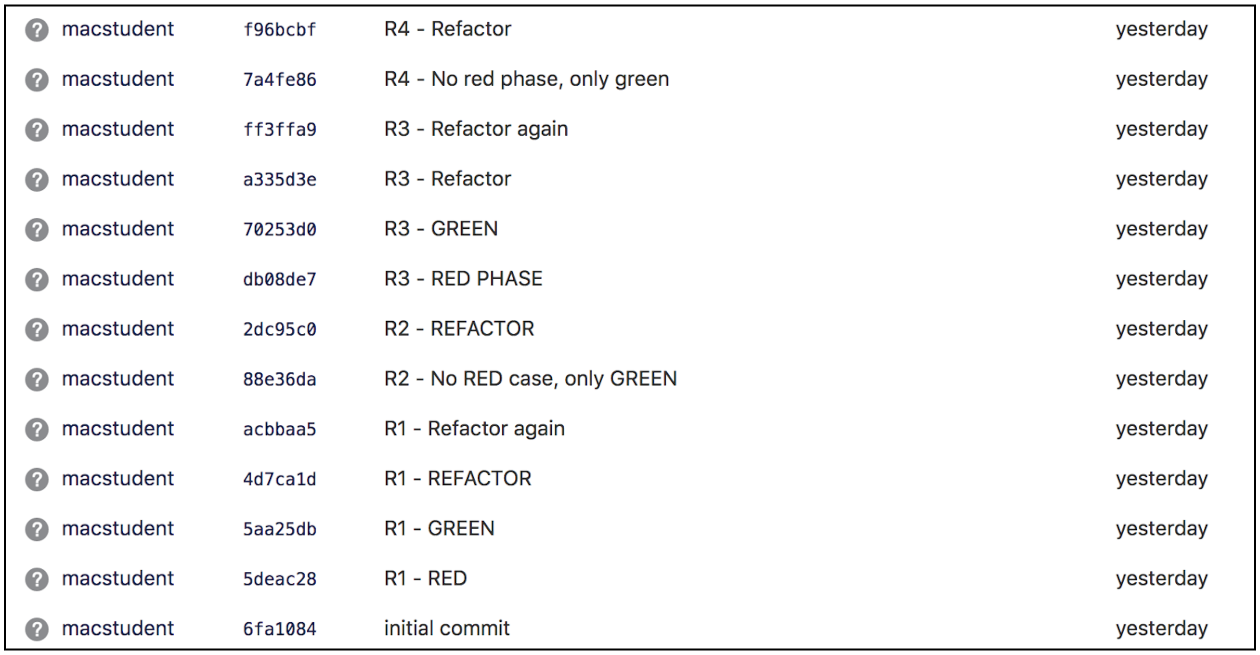
- R1: Accepts 1 integer value, n
  - N > 1
- R2: If N < 1, then return false
- R3: If n is even, return true
- R4: If n is odd, return false

Solution

Code:
- https://bitbucket.org/jenelleteaches/tddexamples/src/master/src/EvenNumbers.java

Java:
- https://bitbucket.org/jenelleteaches/tddexamples/src/master/test/EvenNumberTest.java

Check the commits to see the RED/GREEN/REFACTOR



macstudent	f96bcbf	R4 - Refactor	yesterday
macstudent	7a4fe86	R4 - No red phase, only green	yesterday
macstudent	ff3ffa9	R3 - Refactor again	yesterday
macstudent	a335d3e	R3 - Refactor	yesterday
macstudent	70253d0	R3 - GREEN	yesterday
macstudent	db08de7	R3 - RED PHASE	yesterday
macstudent	2dc95c0	R2 - REFACTOR	yesterday
macstudent	88e36da	R2 - No RED case, only GREEN	yesterday
macstudent	acbbaa5	R1 - Refactor again	yesterday
macstudent	4d7ca1d	R1 - REFACTOR	yesterday
macstudent	5aa25db	R1 - GREEN	yesterday
macstudent	5deac28	R1 - RED	yesterday
macstudent	6fa1084	initial commit	yesterday

Problem #2: Software Sales

Problem Description:

Microsoft sells a software package for $99. A discount is given to all customers who purchase a certain number of packages.

Write a program that calculates the prices of software!! Depending on quantity:

Quantity Purchased	Discount
10-19	20%
20-49	30%
50-99	40%
100 or more	50%

If user enters invalid number (quantity < 0) ---> return -1

Function definition

```
calculatePrice(quantity)
```

→ returns the total price of the software package

Solution:

Requirements:

R1: Price of software is $99 / package
R2: Discount for 10-19 packages is 20%

- R2a: Final price for 10-19 is calculated correctly

R3: Discount for 20-49 packages is 30%

- R3a: Final price for 20-49 is calculated correctly

R4: Discount for 50-99 packages is 40%

- R4a: Final price for 50-99 s calculated correctly

R5: Discount for 100+ packages is 50%

- R5a: Final price for 100+ is calculated correctly

R6:   If quantity < 0, then return -1

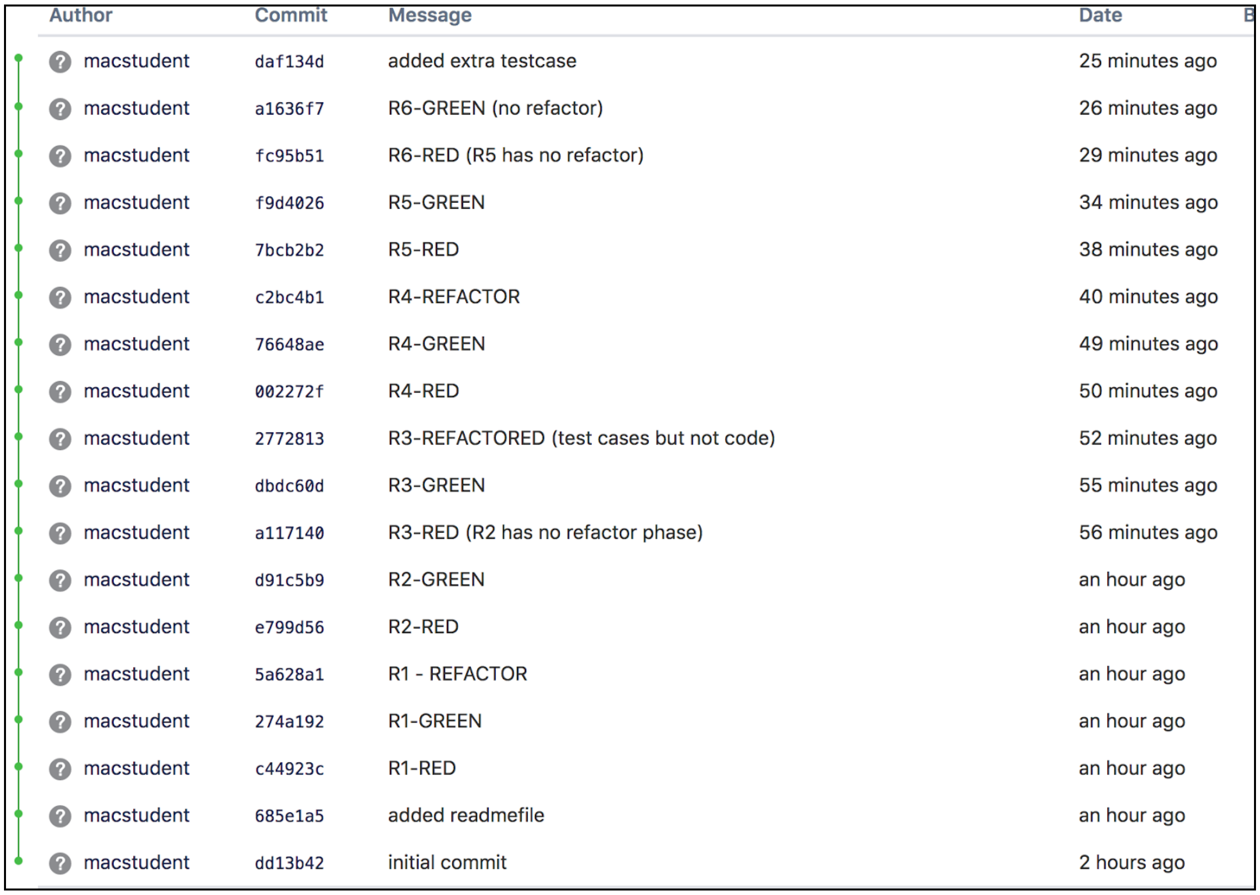
Test Data

Requirement	Test Input	Expected Output
R1	1	99
R2/ R2a	12	Subtotal 12*99 = 1188 20% discount = 237.6 Final price: **950.40**
R3	30	Subtotal: 30*99 = 2970 Discount: 30%: (2970 * 30/100) = 891 Final price = **2079**
R4	60	Subtotal: 60*99 = 5940 Discount: 40% = 2376 Final price = **3564**
R5	120	Subotal: 120*99 = 11880 Discount: 50% = 5940 Final Price: **5940**
R6	-100	-1

Solution:

https://bitbucket.org/jenelleteaches/tddexamples-softwaresales/src/master/

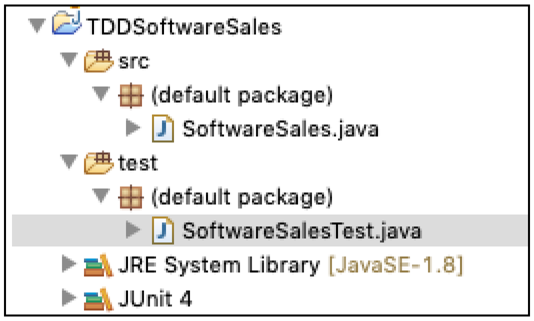
RED/GREEN/REFACTORS



Author	Commit	Message	Date
macstudent	daf134d	added extra testcase	25 minutes ago
macstudent	a1636f7	R6-GREEN (no refactor)	26 minutes ago
macstudent	fc95b51	R6-RED (R5 has no refactor)	29 minutes ago
macstudent	f9d4026	R5-GREEN	34 minutes ago
macstudent	7bcb2b2	R5-RED	38 minutes ago
macstudent	c2bc4b1	R4-REFACTOR	40 minutes ago
macstudent	76648ae	R4-GREEN	49 minutes ago
macstudent	002272f	R4-RED	50 minutes ago
macstudent	2772813	R3-REFACTORED (test cases but not code)	52 minutes ago
macstudent	dbdc60d	R3-GREEN	55 minutes ago
macstudent	a117140	R3-RED (R2 has no refactor phase)	56 minutes ago
macstudent	d91c5b9	R2-GREEN	an hour ago
macstudent	e799d56	R2-RED	an hour ago
macstudent	5a628a1	R1 - REFACTOR	an hour ago
macstudent	274a192	R1-GREEN	an hour ago
macstudent	c44923c	R1-RED	an hour ago
macstudent	685e1a5	added readmefile	an hour ago
macstudent	dd13b42	initial commit	2 hours ago

TDD
TEST → CODE → REFACTOR
RED ---> GREEN --REFACTOR

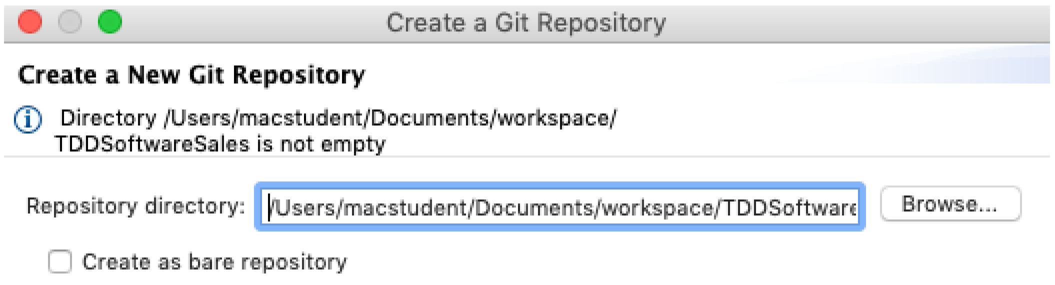
1. Set up your java porject in eclipse



```
▼ TDDSoftwareSales
  ▼ src
    ▼ (default package)
      ▶ SoftwareSales.java
  ▼ test
    ▼ (default package)
      ▶ SoftwareSalesTest.java
  ▶ JRE System Library [JavaSE-1.8]
  ▶ JUnit 4
```

2. Conenct your github account

Setup a new local repostority
Make sure you select your JAVA PROJECT folder



Create a Git Repository

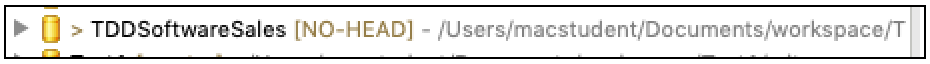
Create a New Git Repository

Directory /Users/macstudent/Documents/workspace/
TDDSoftwareSales is not empty

Repository directory: /Users/macstudent/Documents/workspace/TDDSoftware

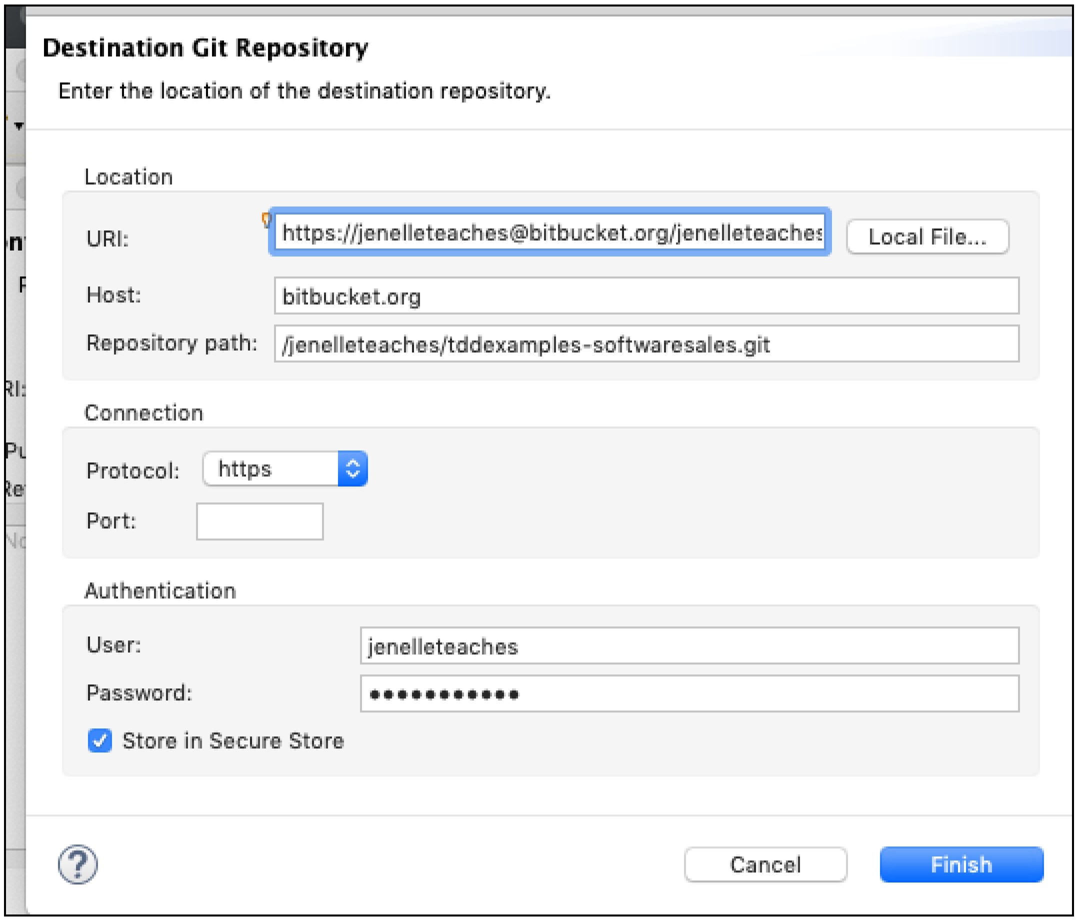
☐ Create as bare repository

Exclipse will update:



```
▶ > TDDSoftwareSales [NO-HEAD] - /Users/macstudent/Documents/workspace/T
```

Add your remote


Destination Git Repository

Enter the location of the destination repository.

Location

URI: https://jenelleteaches@bitbucket.org/jenelleteaches

Host: bitbucket.org

Repository path: /jenelleteaches/tddexamples-softwaresales.git

Connection

Protocol: https

Port:

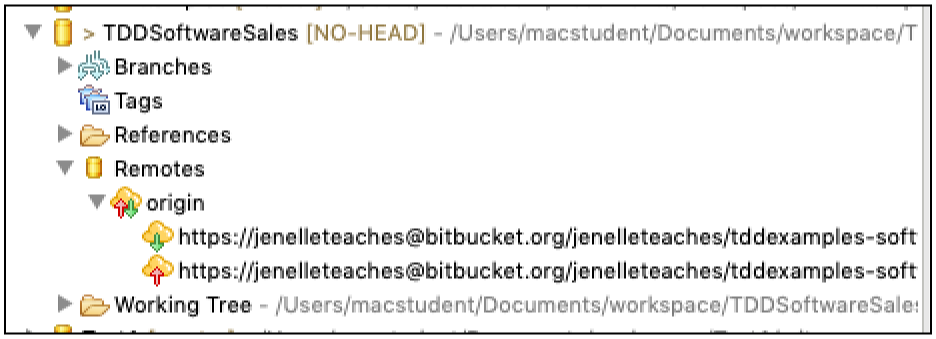
Authentication

User: jenelleteaches

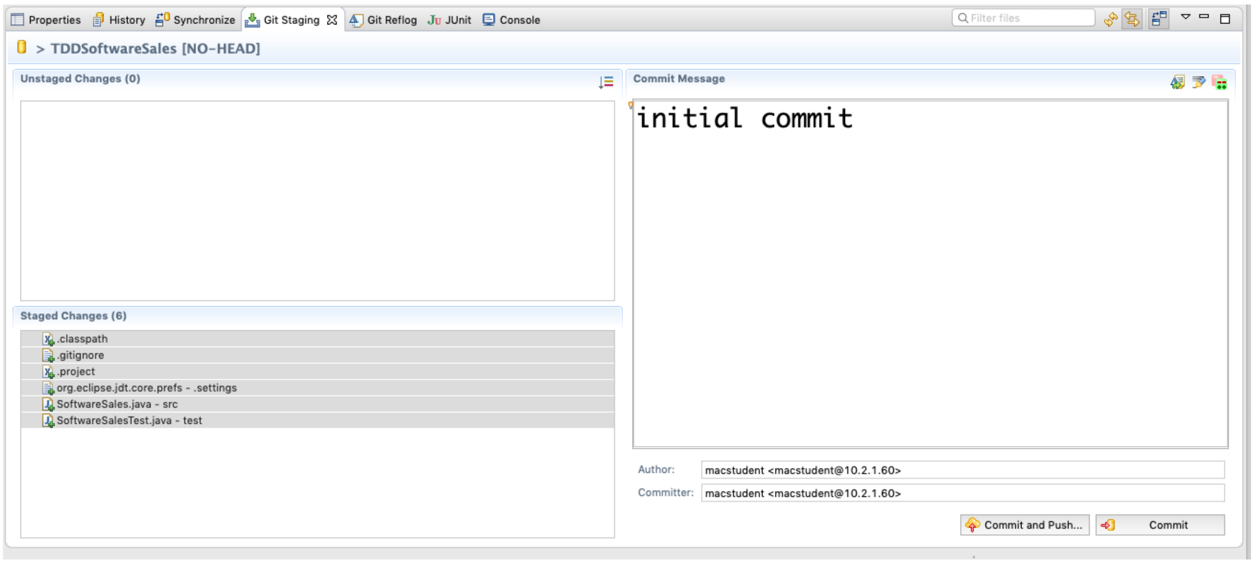
Password: ●●●●●●●●●●●●

☑ Store in Secure Store

Remotes will update


```
▼ > TDDSoftwareSales [NO-HEAD] - /Users/macstudent/Documents/workspace/T
  ▶ Branches
    Tags
  ▶ References
  ▼ Remotes
    ▼ origin
        https://jenelleteaches@bitbucket.org/jenelleteaches/tddexamples-soft
        https://jenelleteaches@bitbucket.org/jenelleteaches/tddexamples-soft
  ▶ Working Tree - /Users/macstudent/Documents/workspace/TDDSoftwareSale
```

Do an initial commit of your project



TDDSoftwareSales [NO-HEAD]

Unstaged Changes (0)

Staged Changes (6)

- .classpath
- .gitignore
- .project
- org.eclipse.jdt.core.prefs - .settings
- SoftwareSales.java - src
- SoftwareSalesTest.java - test

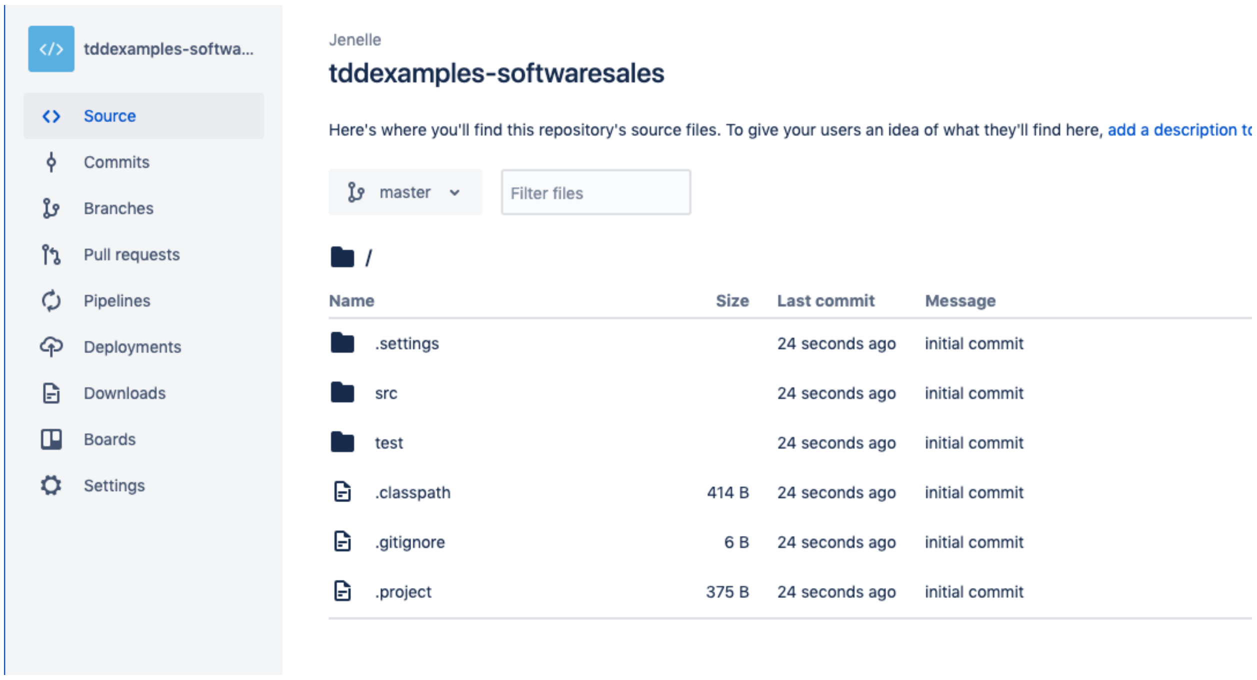
Commit Message

```
initial commit
```

Author: macstudent <macstudent@10.2.1.60>

Committer: macstudent <macstudent@10.2.1.60>

Refresh your bitbucket



Jenelle

tddexamples-softwaresales

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, add a description to

/

Name	Size	Last commit	Message
.settings		24 seconds ago	initial commit
src		24 seconds ago	initial commit
test		24 seconds ago	initial commit
.classpath	414 B	24 seconds ago	initial commit
.gitignore	6 B	24 seconds ago	initial commit
.project	375 B	24 seconds ago	initial commit

Problem #4 - FizzBuzzWhizz using TDD

Create a function that returns: "fizz", "buzz" or "fizzbuzz".

The function should:

- R0: Function must accept a number > 0. Number less than 0 return “error”
- R1: Return "fizz" if the number is divisible by 3
- R2:Return "buzz" if the number is divisible by 5
- R3: Return "fizzbuzz" if the number is divisible by 3 or 5 (15)
- R4: Return the number if no other requirement is fulfilled. The numbers must be returned as a string.
- R5: If number is prime, return “whizz”
- R6 If number meets (R5) AND (R1, R2, or R3) - append “whizz” to end of string
    - Example:
        - 1 returns “1”
        - 2 returns “whizz”
        - 3 returns “fizzwhizz”
        - 4 returns “4”
        - 5 returns “buzzwhizz”
        - 6 returns “6”
        - 7 returns “whizz”
        - 9 returns “fizz”
        - 10 returns “buzz”
        - 15 returns “fizzbuzz”

Solution

What does function look like?

buzzzzzzz(number)
-> number = a integer
-> function returns a String

What files do we need?

- FizzBuzz.java → Code
- FizzBuzzTest.java → JUnit

Bitbucket repo: https://bitbucket.org/jenelleteaches/tdd-s19-fizzbuzz/src/master/