



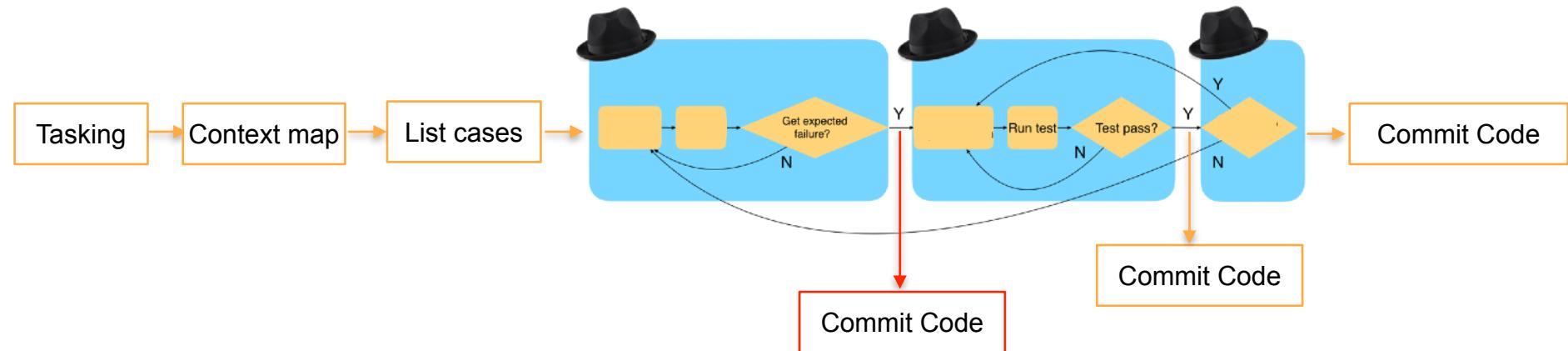
AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks ©

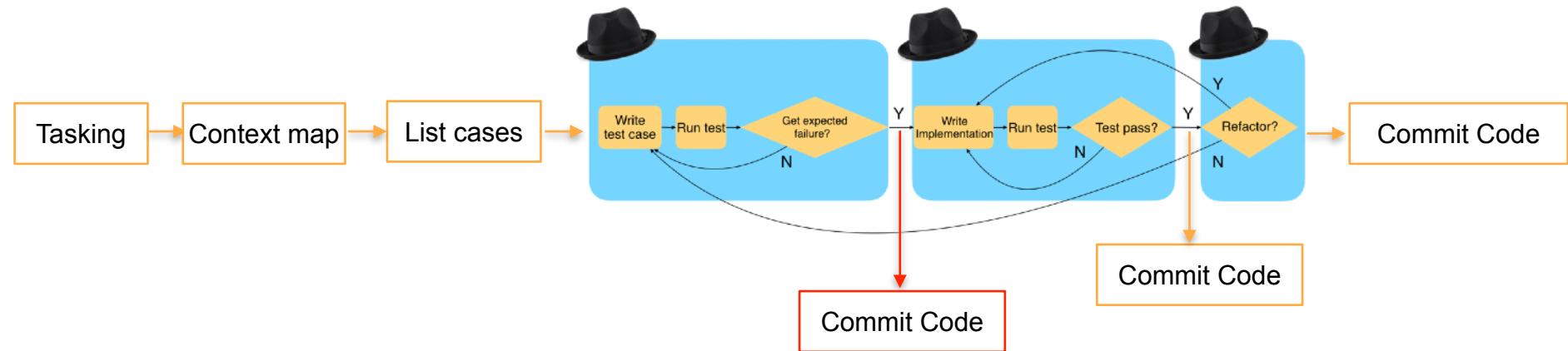
TDD - Practice

/thoughtworks

Whole Process



Whole Process



Why TDD

- Quick feedback
- Test as documentation
- Safety net

Mars Rover

Imagine you are in charge of software development on the Mars exploration team. Now you have to write a control program for the exploration vehicle landing on Mars, and control the actions of the rover according to the control commands sent by the earth.

The Mars is initialized with given x-coordinate, y-coordinate and direction, for example, (0, 0, "N"). The direction contains N, S, E, W.

The commands received by the rover fall into three categories:

1. Move commands:

The rover can move forward (M), one space at a time.

2. Turn left commands:

The rover can turn left 90 degrees (L).

3. Turn right commands:

The rover can turn right 90 degrees (R).

Mars Rover

Imagine you are in charge of software development on the Mars exploration team. Now you have to write a control program for the exploration vehicle landing on Mars, and control the actions of the rover according to the control commands sent by the earth.

The Mars is initialized with given x-coordinate, y-coordinate and direction, for example, (0, 0, "N"). The direction contains N, S, E, W.

The commands received by the rover fall into three categories:

1. Move commands:

The rover can move forward (M), one space at a time.

2. Turn left commands:

The rover can turn left 90 degrees (L).

3. Turn right commands:

The rover can turn right 90 degrees (R).

- What does “commands” look like

It should be: String like “M”, “L”, “R”

- What is the format of Report

It should be like “0 0 N”

- What about the commands not supported

Ignore this, suppose commands are all valid

- What if commands are empty

Ignore this, it doesn’t happen

Mars Rover

Imagine you are in charge of software development on the Mars exploration team. Now you have to write a control program for the exploration vehicle landing on Mars, and control the actions of the rover according to the control commands sent by the earth.

The Mars is initialized with given x-coordinate, y-coordinate and direction, for example, (0, 0, "N"). The direction contains N, S, E, W.

The commands received by the rover fall into three categories:

1. Move commands:

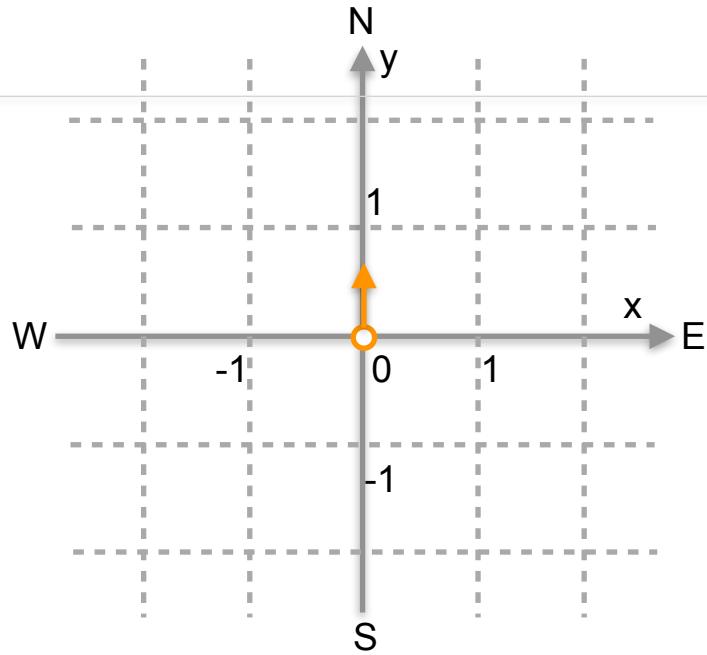
The rover can move forward (M), one space at a time.

2. Turn left commands:

The rover can turn left 90 degrees (L).

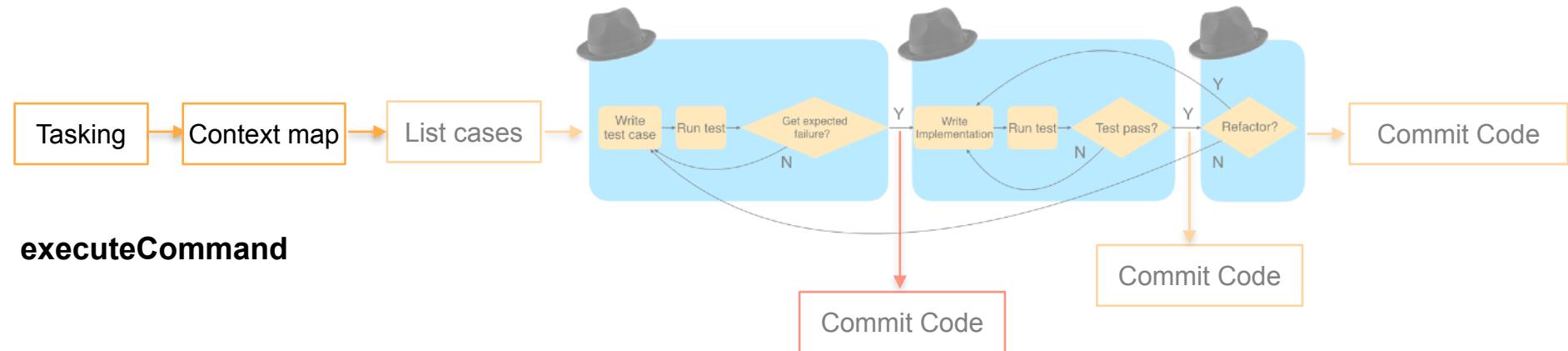
3. Turn right commands:

The rover can turn right 90 degrees (R).



1. M => ?
2. L => ?
3. M => ?
4. R => ?

Tasking & Context map



Mars Rover

1. executeCommand

- input:
 - locationX: int
 - locationY: int
 - direction: string
 - command: string
- output:
 - locationX:int
 - locationY:int
 - direction:string

1.1 move

- input:
 - locationX: int
 - locationY: int
 - direction: string
 - command: string
- output:
 - locationX:int
 - locationY:int
 - direction:string

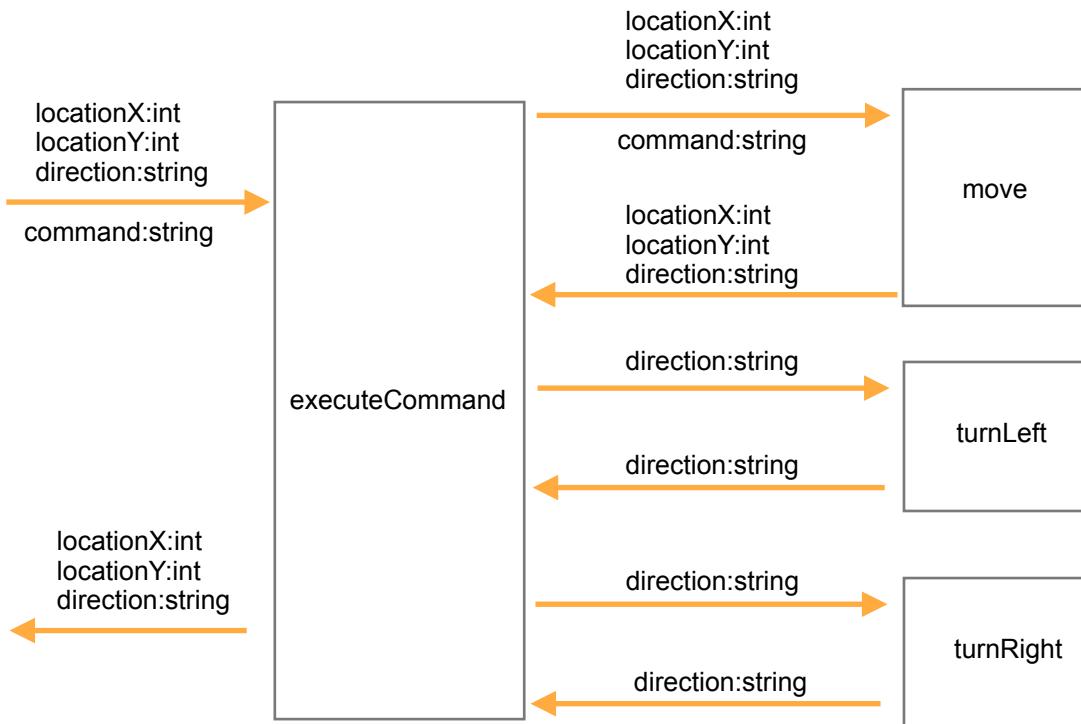
1.2 turnLeft

- input:
 - direction: string
 - command: string
- output:
 - direction:string

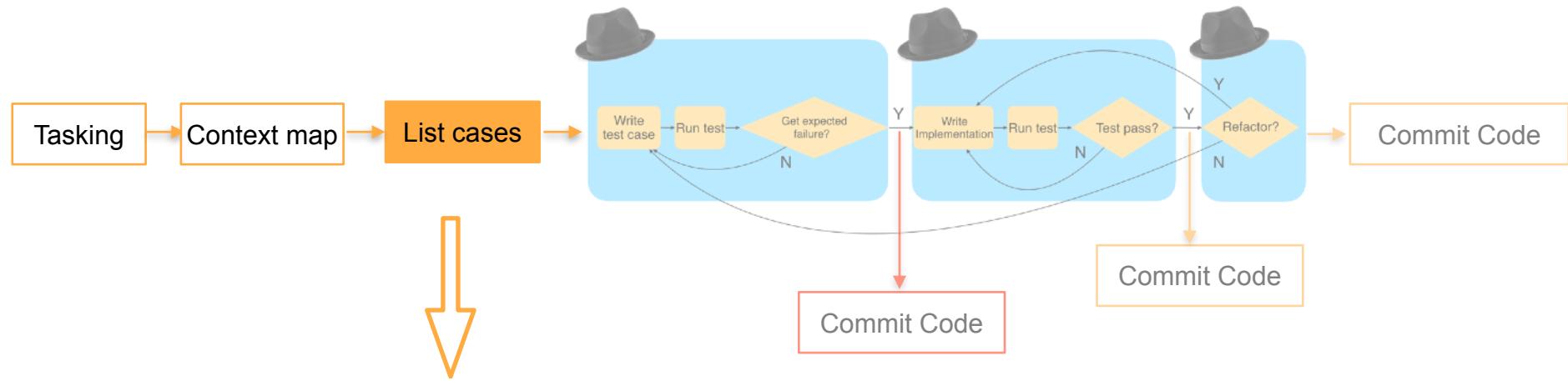
1.3 turnRight

- input:
 - direction: string
 - command: string
- output:
 - direction:string

Mars Rover

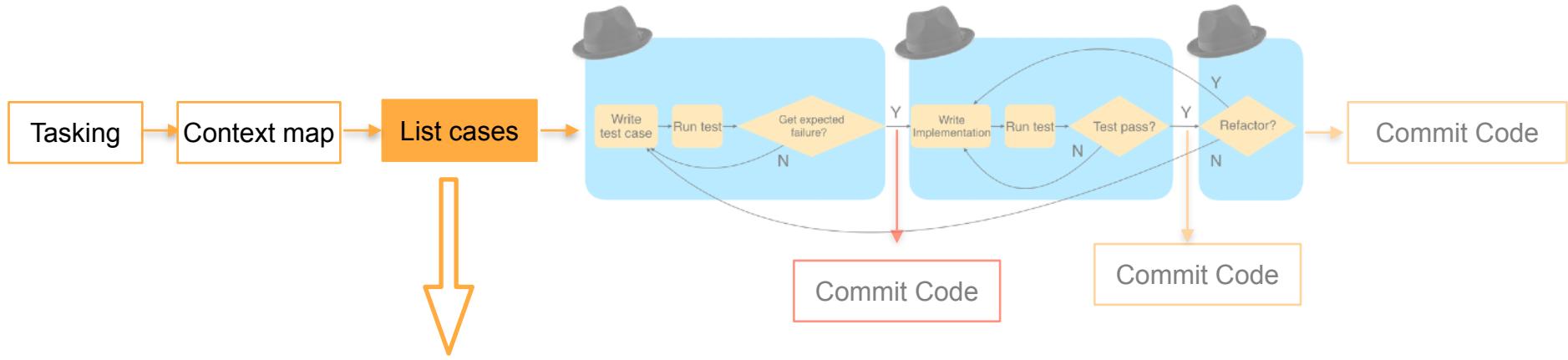


List Cases



Given xxxx When executeCommand Then xxxxx

List Cases



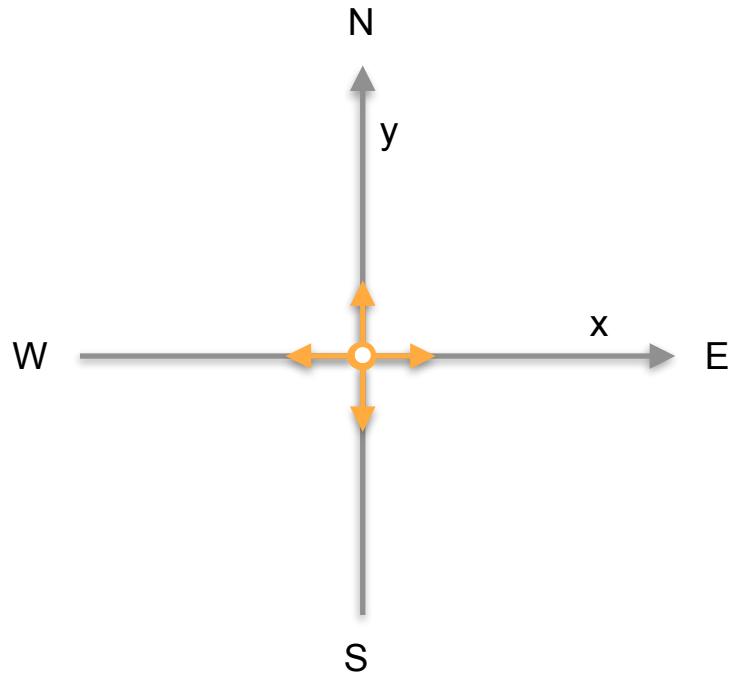
Given xxxx	When executeCommand	Then xxxxx
Given locationX: 0 locationY: 0 direction: N command: M	When executeCommand	Then locationX: 0 locationY: 1 direction: N

List cases for executeCommand

- Do it individually in breakout-room
- Share screen when you are starting
- 10 minutes

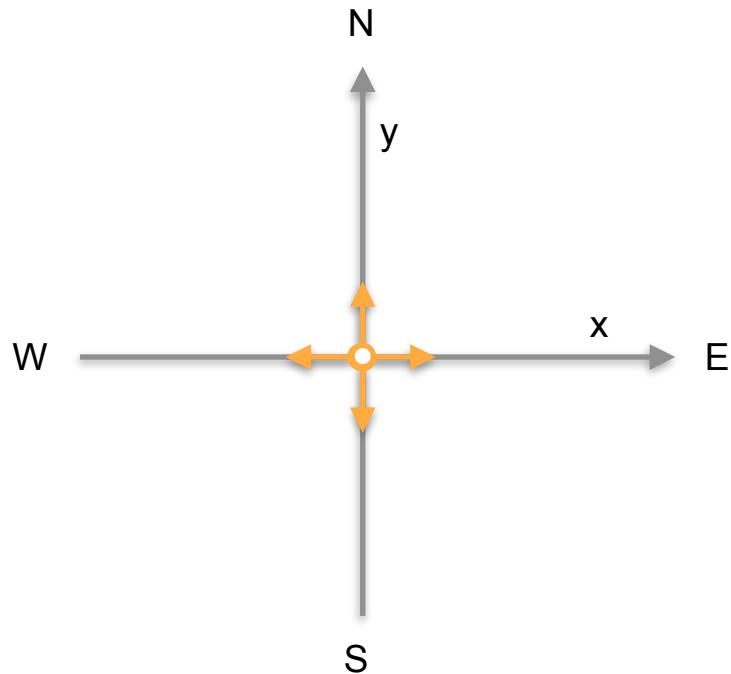
Given xxxx	When executeCommand	Then xxxxx
Given locationX: 0 locationY: 0 direction: N command: M	When executeCommand	Then locationX: 0 locationY: 1 direction: N

MarsRover - list cases of `executeCommand`



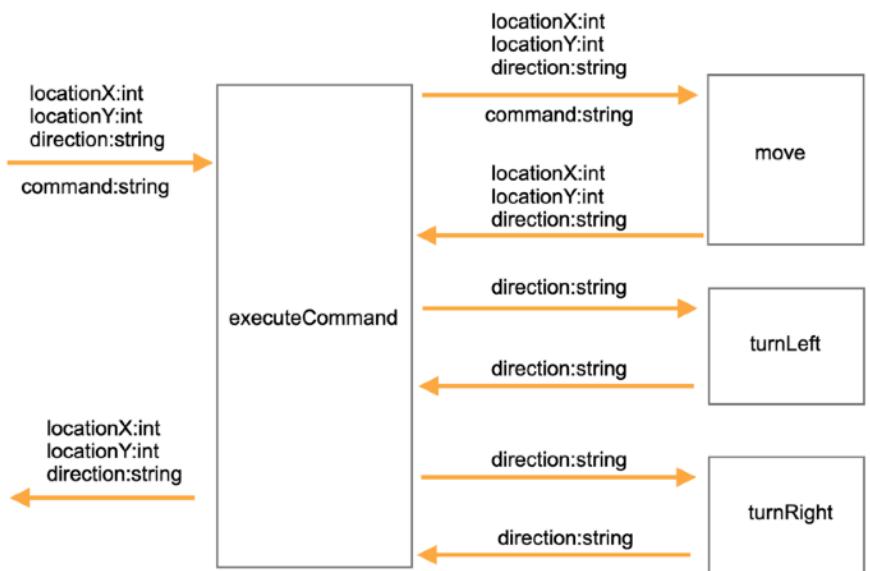
locationX locationY direction	command	M	L	R
0 0 N				
0 0 S				
0 0 E				
0 0 W				

MarsRover - list cases of `executeCommand`



command locationX locationY direction	M	L	R
0 0 N	0 1 N	0 0 W	0 0 E
0 0 S	0 -1 S	0 0 E	0 0 W
0 0 E	1 0 E	0 0 N	0 0 S
0 0 W	-1 0 W	0 0 S	0 0 N

MarsRover - TDD Demo



command locationX locationY direction	M	L	R
0 0 N	0 1 N	0 0 W	0 0 E
0 0 S	0 -1 S	0 0 E	0 0 W
0 0 E	1 0 E	0 0 N	0 0 S
0 0 W	-1 0 W	0 0 S	0 0 N

Practice

- Find the task on learning Platform: Java Engineering -> TDD Basic -> Mars Rover
- Implement Mars Rover using TDD

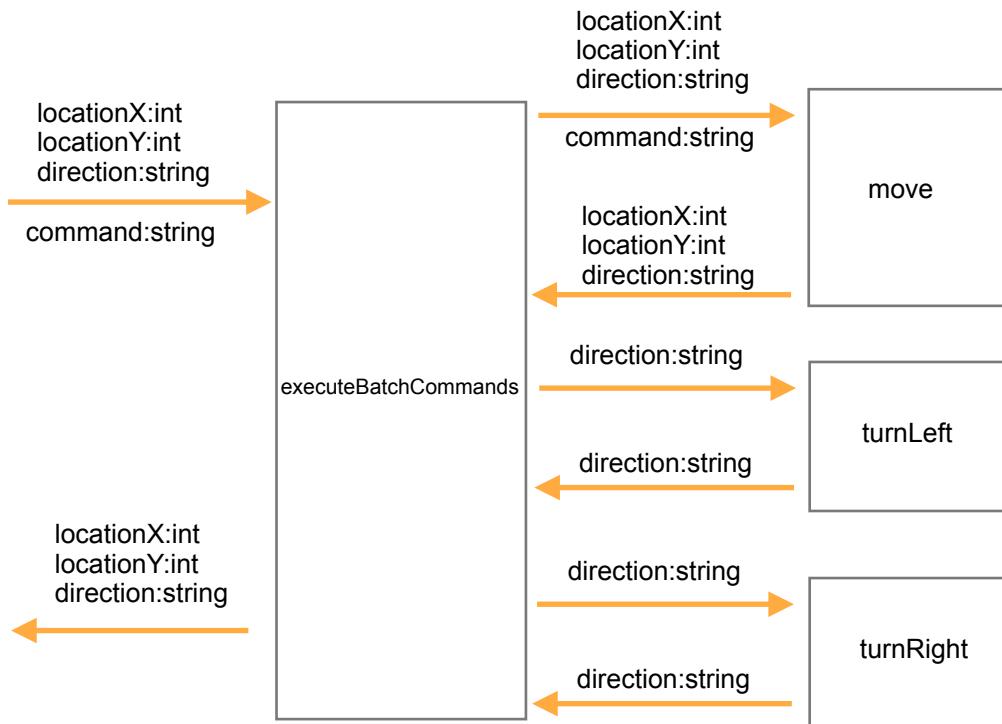
Requirement

1. Baby step commit with meaningful message as **type: detail message**
 - Commit after writing a test with expectation failure
 - Commit after writing code to make failed test pass
 - Commit after refactoring
2. Format code before commit
3. Using meaningful names

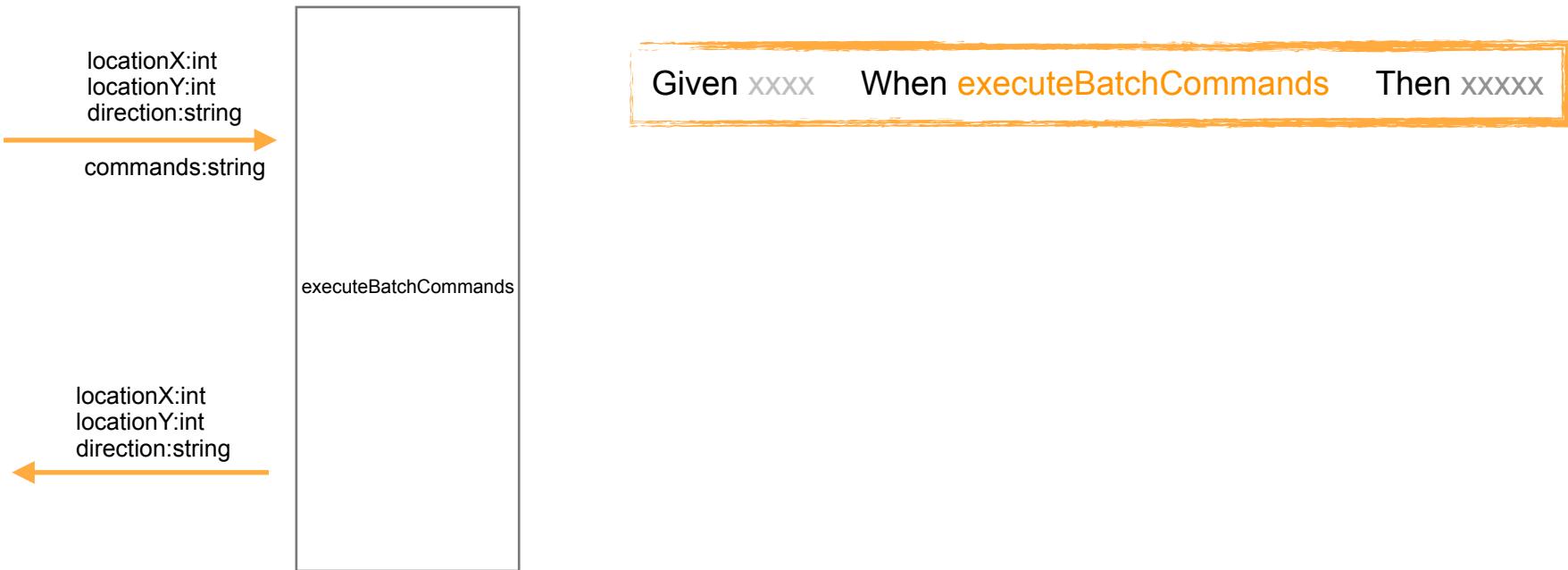
Mars Rover

executeCommand —> executeBatchCommands

Mars Rover Context Map



MarsRover - list cases



MarsRover - list cases

Given xxxx	When executeBatchCommands	Then xxxxx
Given locationX: 0 locationY: 0 direction: N command: MLMR	When executeBatchCommands	Then locationX: -1 locationY: 1 direction: N

Homework

1. Diary using ORID



22:00pm

2. Finish Mars Rover using TDD

- Finish coding implementation of **13 cases**
- Baby step commit, commit **once with one test case or related implementation**, commit message format is **type: detail message**
- Submit your homework with your GitHub repository address

Extra points:

- No obvious access modifier usage errors, e.g., no incorrect use of private, public
- No obvious duplicate code
- Use Java stream instead of for loop
- Use meaningful names for variables, methods
- No more than 4 if/else branches inside one method

MarsRover - list cases of `executeBatchCommands`

	M	L	comm locationX locationY direction
0 0 N	0 1 N	0 0 W	0 0 E
0 0 S	0 -1 S	0 0 E	0 0 W
0 0 E	1 0 E	0 0 N	0 0 S
0 0 W	-1 0 W	0 0 S	0 0 N

Given xxxx	When <code>executeBatchCommands</code>	Then xxxxx
Given locationX: 0 locationY: 0 direction: N command: M	When <code>executeBatchCommands</code>	Then locationX: 0 locationY: 1 direction: N
Given locationX: 0 locationY: 0 direction: N command: MLMR	When <code>executeBatchCommands</code>	Then locationX: -1 locationY: 1 direction: N

Teamwork

- **Session Requirement**

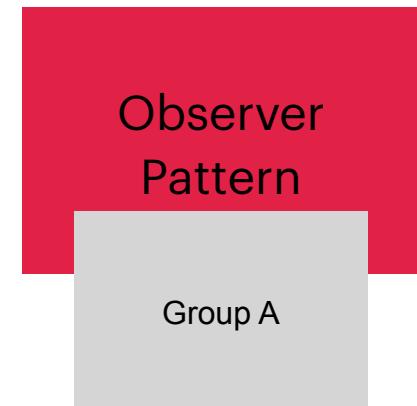
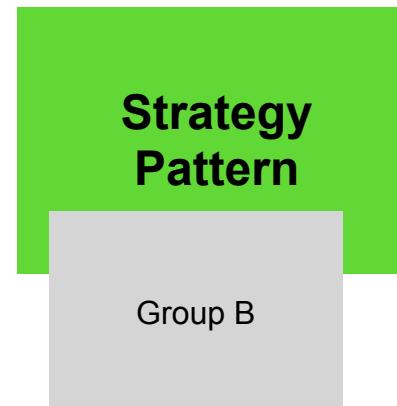
- **Group work** in collect material, design slides, code implementation and presentation
- Showcase Time: ***10:40 am next Monday***
- Duration: **20 minutes** per group
- **Everyone** should participate in presentation

- **Session should answer the following questions**

- What is it?
- What kind of problems does It solve?
- When do we need to use it?
- How to implement it?(Simple code sample.)

Topic Assignment

Group 1	Group 2
ALAN WAN	ANTONY CHOI
ALVIN LEUNG	CHRIS T C WONG
HEINRICH SIU	JOYCE LI
JENNY WONG	KELVIN TO
MARIE CHOW	POLLY LEUNG
MICHAEL NAM	VINCENT TAM
THOMAS K Y KWOK	





AFS

Agile Full Stack
Developer Bootcamp
Thoughtworks ©



Thank You !

/thoughtworks