

Name : Mahima Bisht  
EmailId : mahima.bisht@  
capgemini.com  
EmpId : 46010069

## 4/Sept/2020 Java BDD USING CUCUMBER AND GHERKIN

### ▷ Test Domain Layer

- here we <sup>call</sup> ~~calls~~ webdriver and  
REST API.

- we should not put these calls  
to glue code and make it  
complex.

### ▷ Test Runner Class

- finds feature files and read them

Glue code



4/Sept/2020

## △ Test Runner Class

Name = Mahima Bisht

Email = mahima.bisht@  
calogemini.com

EmpId = 46010069

- finds feature files and read them
- find glue code.

Here, @RunWith(Cucumber.class) tells the junit to hand over the control to Cucumber

- And, @CucumberOptions Annotation helps us to configure such as where cucumber should look for feature files

Here, in "glue" options, cucumber finds the step definitions

- Default feature directory is src/test/features.



Name = Mahima Bish  
EmailId = mahima.bish@  
Capgemini.com  
EmpId = 46010069

4/Sept/2020

## △ STEP DEFINITIONS

→ Passing parameters through feature

for ex: Given I have 5 files

Step def created for this step in <sup>step def</sup> ~~feature~~  
file will be as

```
@Given ("I have {int} files")  
public void ihavefiles (int arg) {  
}
```

→ we can pass different types of parameters

ex: @Given I have a "pizza"

```
@Given ("I have a {String}")  
public void ihavea (String arg) { --- }
```

or, Given I have a 2.22 m scale

```
@Given ("I have a {float} m scale")  
public void ihaveamScale (float arg) {  
  // ---  
}
```



4/Sept/2020

Regular Expression

Name = Mahima Bisht

EmailId = mahima.bisht@capgemini

EmpId = 46010069

① @ Given ("^ I have ordered (.\*) pineapples\$")  
Start of string      matches anything      any no. of times      end of string

② Use \d+ for digits and \w+ for strings  
③ s?he



4 Sept 2020

Name = Mahima Bisht  
Email Id = mahima.bisht@calpagemin.com  
EmpId = 46010669

## BDD Process

### 1) ILLUSTRATE PHASE

- team talks about examples how feature would work
- how it is beneficial.

### 2) FORMULATE PHASE

- turns that examples into specifications or acceptance tests
- these tests/specification are created so as to execute as test suite
- ~~- language used Gherkin.~~

### 3) Automate Automate

- write those acceptance tests in gherkin in a test automation tool.
- once tests passed. →

### 4) Validate

- we validate that the feature does what is expected.
- and provide feedback.