

LUXOFT TRAINING

SQA-051 Школа автоматизированного
тестирования. Часть 3. Тестирование с
использованием Cucumber

think.
create.
accelerate.

SQA-051 Школа автоматизированного тестирования. Часть 3. Тестирование с использованием Cucumber

Расширенные возможности Cucumber

СОДЕРЖАНИЕ ТРЕНИНГА

▪ Работа с тегами	4
▪ Hooks	17
▪ Step Definition	27

СЕКЦИЯ 1: РАБОТА С ТЕГАМИ

ПРИМЕНЕНИЕ ТЕГОВ

- Теги используются для организации файлов **Feature** и **Scenarios**. Чтобы добавить тег к **Scenario**, нужно вставить слова с префиксом @ в строке перед ключевым словам Scenario.
- Feature/Scenario могут иметь несколько тегов; просто разделите их пробелами или перенесите на другую строку.

ПРИМЕНЕНИЕ

ЗАЧЕМ НУЖЕН TAG?

Для выборочного запуска сценариев

Или выключения сценариев из
процесса

КОД

@scores

Функция: Снятие денег со счета

@success

Сценарий: Успешное снятие денег со счета

Дано на счете пользователя имеется

120000 рублей Когда пользователь

снимает со счета 20000 рублей Тогда на
счете пользователя имеется 100000 рублей

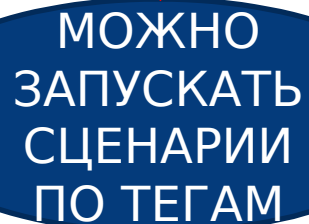
@fail

Сценарий: Снятие денег со счета -
недостаточно денег

Дано на счете пользователя имеется 100
рублей Когда пользователь снимает со

счета 120 рублей Тогда появляется
предупреждение "На счете недостаточно
денег"

МОЖНО
ЗАПУСКАТЬ
СЦЕНАРИИ
ПО ТЕГАМ



ПРИМЕНЕНИЕ ТЕГОВ

@important

Feature: Home Page

Background: flow till home page

Given user is on Application home page

@sanity

Scenario: Home Page Default content

Then user gets a GitHub Bootcamp section

@regression

Scenario: GitHub Bootcamp Section

When user focuses on GitHub Bootcamp Section

Then user gets an option to setup git

@sanity @regression

Scenario: Top Banner content

When user focuses on Top Banner

Then user gets an option of home page

ЗАПУСК ПО ТЕГАМ

ТОЛЬКО С ТЕГОМ SUCCESS

```
@RunWith(Cucumber.class)
@CucumberOptions(
    tags = "@success"
)
public class RunnerTest { }
```

ВСЕ СЦЕНАРИИ, КРОМЕ @SUCCESS

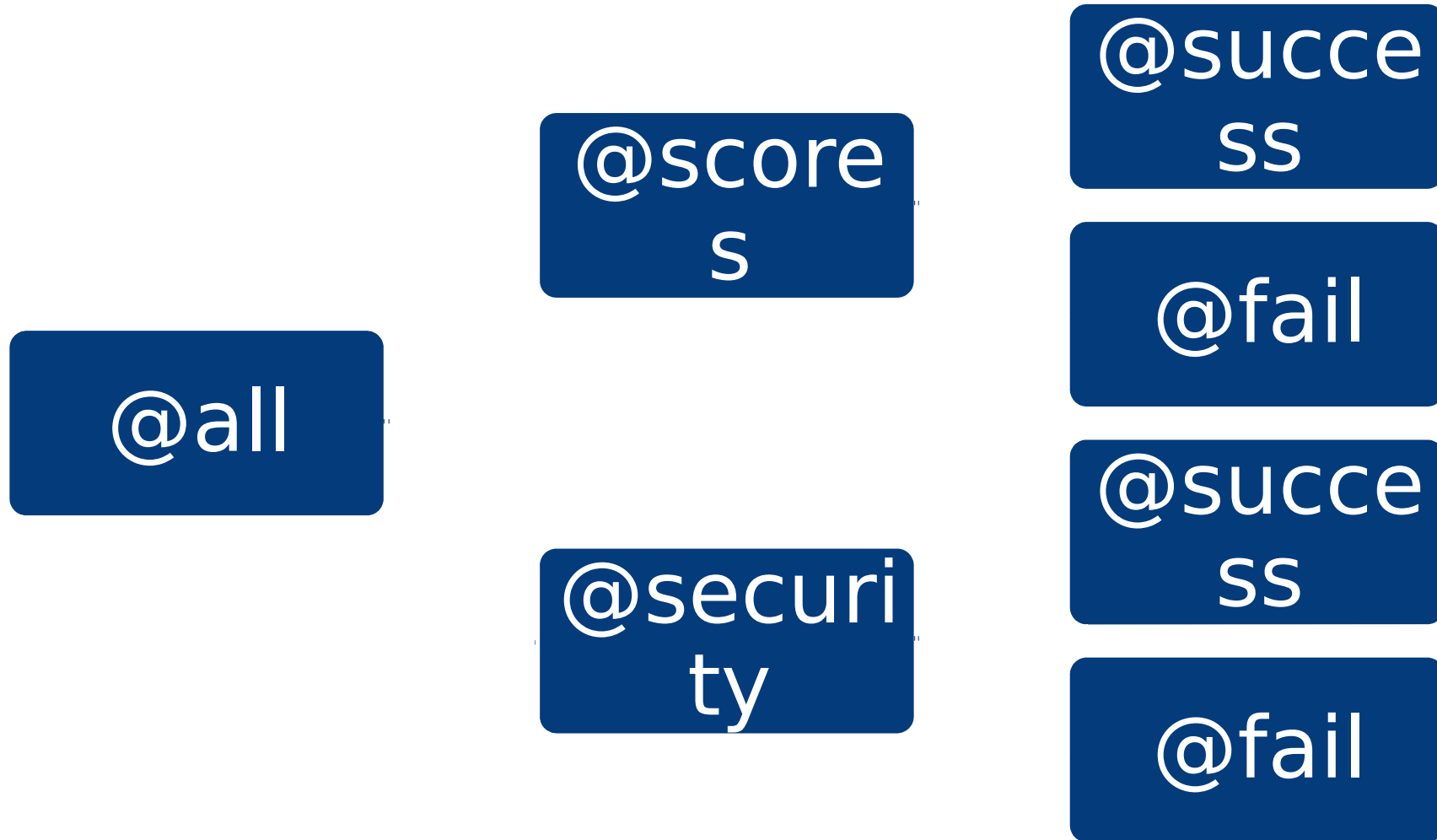
```
@RunWith(Cucumber.class)
@CucumberOptions(
    tags = "~@success"
)
public class RunnerTest { }
```


ЗАПУСК ПО ТЕГАМ

СЦЕНАРИИ @SCORES И
@SECURITY

```
@RunWith(Cucumber.class)
@CucumberOptions(
    tags = "@scores, @security"
)
public class RunnerTest { }
```

ПРИМЕР КЛАССИФИКАЦИИ



ПРИМЕНЕНИЕ ТЕГОВ

- `mvn test -Dcucumber.options="--tags @important"`

```
-----
T E S T S
-----
Running com.CucumberOptions.RunCukeTest
@important
Feature: Home Page

    Background: flow till home page          # sample.feature:4
        Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()

    @sanity
    Scenario: Home Page Default content      # sample.feature:8
        Then user gets a github bootcamp section # LoginSteps.user_gets_a_github_bootcamp_section()

    Background: flow till home page          # sample.feature:4
        Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()

    @regression
    Scenario: GitHub Bootcamp Section        # sample.feature:12
        When user focuses on GitHub Bootcamp Section # LoginSteps.user_focuses_on_GitHub_Bootcamp_Section()
        Then user gets an option to setup git      # LoginSteps.user_gets_an_option_to_setup_git()

    Background: flow till home page          # sample.feature:4
        Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()

    @sanity @regression
    Scenario: Top Banner content              # sample.feature:17
        When user focuses on Top Banner        # LoginSteps.user_focuses_on_Top_Banner()
        Then user gets an option of home page # LoginSteps.user_gets_an_option_of_home_page()

3 Scenarios (3 passed)
8 Steps (8 passed)
0m0.204s

Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.926 sec
```

ПРИМЕНЕНИЕ ТЕГОВ

- `mvn test -Dcucumber.options="--tags @sanity"`

```
-----
T E S T S
-----
Running com.CucumberOptions.RunCukeTest
@important
Feature: Home Page

    Background: flow till home page          # sample.feature:4
        Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()

    @sanity
    Scenario: Home Page Default content        # sample.feature:8
        Then user gets a github bootcamp section # LoginSteps.user_gets_a_github_bootcamp_section()

    Background: flow till home page          # sample.feature:4
        Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()

    @sanity @regression
    Scenario: Top Banner content                # sample.feature:17
        When user focuses on Top Banner        # LoginSteps.user_focuses_on_Top_Banner()
        Then user gets an option of home page # LoginSteps.user_gets_an_option_of_home_page()

2 Scenarios (2 passed)
5 Steps (5 passed)
0m0.236s

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.981 sec

Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0
```

ПРИМЕНЕНИЕ ТЕГОВ

- `mvn test -Dcucumber.options="--tags @important"` запустит все Сценарии (так как мы запускаем Тег, связанный с Функцией).
- `mvn test -Dcucumber.options="--tags @sanity"` запустит Сценарии, связанные с @sanity.

ПРИМЕНЕНИЕ ОПЕРАЦИЙ AND И OR К ТЕГАМ

Если вы хотите запустить Сценарии с тегами sanity и regression, выполните в командной строке следующую команду:

```
mvn test -Dcucumber.options="--tags '@sanity and @regression'"
```

```
-----  
T E S T S  
-----  
Running com.CucumberOptions.RunCukeTest  
@important  
Feature: Home Page  
  
    Background: flow till home page          # sample.feature:4  
        Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()  
  
    @sanity @regression  
    Scenario: Top Banner content              # sample.feature:17  
        When user focuses on Top Banner      # LoginSteps.user_focuses_on_Top_Banner()  
        Then user gets an option of home page # LoginSteps.user_gets_an_option_of_home_page()  
  
1 Scenarios (1 passed)  
3 Steps (3 passed)  
0m0.431s  
  
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.011 sec
```

ПРИМЕНЕНИЕ ОПЕРАЦИЙ AND И OR К ТЕГАМ

@important

Feature: Home Page

Background: flow till home page

Given user is on Application home page

@sanity @wip

Scenario: Home Page Default content

Then user gets a GitHub Bootcamp section

@regression @wip

Scenario: GitHub Bootcamp Section

When user focuses on GitHub Bootcamp Section

Then user gets an option to setup git

@sanity @regression

Scenario: Top Banner content

When user focuses on Top Banner

Then user gets an option of home page

ПРИМЕНЕНИЕ ОПЕРАЦИЙ AND И OR К ТЕГАМ

- Если вы хотите запустить Сценарии с тегами @sanity **AND** @wip или @regression **AND** @wip, выполните следующую команду:
- `mvn test -Dcucumber.options="--tags '(@sanity or @regression) and @wip'"`

```
-----
T E S T S
-----
Running com.CucumberOptions.RunCukeTest
@important
Feature: Home Page

    Background: flow till home page          # sample.feature:4
        Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()

    @sanity @wip
    Scenario: Home Page Default content      # sample.feature:8
        Then user gets a github bootcamp section # LoginSteps.user_gets_a_github_bootcamp_section()

    Background: flow till home page          # sample.feature:4
        Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()

    @regression @wip
    Scenario: GitHub Bootcamp Section        # sample.feature:12
        When user focuses on GitHub Bootcamp Section # LoginSteps.user_focuses_on_GitHub_Bootcamp_Section()
        Then user gets an option to setup git      # LoginSteps.user_gets_an_option_to_setup_git()

2 Scenarios (2 passed)
5 Steps (5 passed)
0m0.180s

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.844 sec
```

СЕКЦИЯ 2: HOOKS

ОПРЕДЕЛЕНИЕ

HOOK ЭТО

Метод

Который должен выполняться ДО
или ПОСЛЕ

тестового сценария

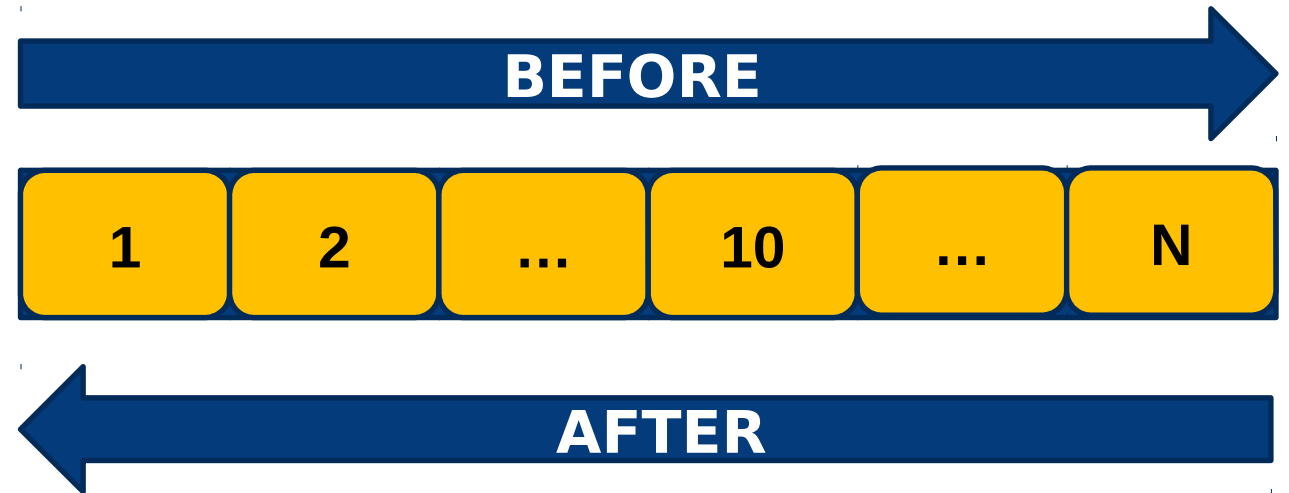
ПРИМЕР

```
import cucumber.api.java.After;  
import cucumber.api.java.Before;  
  
public class Hooks {  
    @Before  
    public void prepareData() {  
        //подготовить данные  
    }  
  
    @After public void clearData() {  
        //очистить данные  
    }  
}
```

ПОРЯДОК ВЫПОЛНЕНИЯ

КОД

```
@Before(order = 10)
public void connectToServer() {
    //подключиться к серверу
}
@Before(order = 20)
public void prepareData() {
    //подготовить данные
}
```



ПО
УМОЛЧАНИЮ
ORDER = 1000

ДОБАВЛЕНИЕ HOOKS В КОД CUCUMBER

```
public class Hooks {  
    public static WebDriver driver = null;  
    public static String browser = "chrome";  
    public static String baseUrl = "https://GitHub.com/";  
  
    @Before  
    public static void createDriver() {  
        System.out.println("Inside Before");  
        createDriver(browser);  
        OpenURL(baseUrl);  
    }  
  
    public static void createDriver(final String browserId) {  
        if (browserId.equalsIgnoreCase("chrome")) {  
            driver = new ChromeDriver();  
        }  
    }  
  
    public static void OpenURL(String baseUrl) {  
        driver.manage().window().maximize();  
        driver.get(baseUrl);  
    }  
    @After  
    public void tearDown() {  
        System.out.println("Inside After");  
        driver.quit();  
    }  
}
```

ДОБАВЛЕНИЕ HOOKS В КОД CUCUMBER

- Теперь запустим home_page.feature

```
Feature: Home Page
  Inside Before
    Background: flow till home page # C:/Users/user/Documents/Xebia/Docs/cucumber/Book/Project/src
    Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()
  Inside After
  Inside Before
    @sanity @wip
    Scenario: Home Page Default content # C:/Users/user/Documents/Xebia/Docs/cucumber/Book/Project/src
      Then user gets a github bootcamp section # LoginSteps.user_gets_a_github_bootcamp_section()

    Background: flow till home page # C:/Users/user/Documents/Xebia/Docs/cucumber/Book/Project/src
    Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()
  Inside After
    @regression @wip
    Scenario: GitHub Bootcamp Section # C:/Users/user/Documents/Xebia/Docs/cucumber/Book/Project/src
      When user focuses on GitHub Bootcamp Section # LoginSteps.user_focuses_on_GitHub_Bootcamp_Section()
      Then user gets an option to setup git # LoginSteps.user_gets_an_option_to_setup_git()

2 Scenarios (2 passed)
5 Steps (5 passed)
1m10.816s
```

HOOKS + TAGS

КОД

```
@Before(value = "@scores", order  
= 20)  
public void connectToServer() {  
}  
@Before(value = "~@fail", order =  
10)  
public void prepareData() {  
}
```


ПРИМЕНЕНИЕ ТЕГОВ К HOOKS

```
public class Hooks {  
    public static WebDriver driver = null;  
    public static String browser = "firefox";  
    public static String baseUrl = "https://GitHub.com/";  
  
    @Before("@web")  
    public static void createDriver() {  
        System.out.println("Inside Web Hook");  
    }  
  
    @Before("@rest")  
    public static void createRestBuilder() {  
        System.out.println("Inside REST Hook");  
    }  
  
    @After  
    public void tearDown() {  
  
        System.out.println("Inside After");  
    }  
}
```

ПРИМЕНЕНИЕ ТЕГОВ К HOOKS

Feature: Home Page

Inside Web Hook

Before Hook code for @web

Background: flow till home page

C:/Users/user/Documents/Xebia/Docs/cucumber/Book/Project/src/

Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()

Inside After

Inside rest Hook

Before Hook code for @rest

@web

Scenario: Home Page Default content on Web # C:/Users/user/Documents/Xebia/Docs/cucumber/Book/Project/src/

Then user gets a github bootcamp section # LoginSteps.user_gets_a_github_bootcamp_section()

Background: flow till home page

C:/Users/user/Documents/Xebia/Docs/cucumber/Book/Project/src/

Given user is on Application home page # LoginSteps.user_is_on_Application_home_page()

Inside After

@rest

Scenario: GitHub account REST Service # C:/Users/user/Documents/Xebia/Docs/cucumber/Book/Project/src/test/

When user sends a GET request # LoginSteps.user_sends_a_GET_request()

2 Scenarios (2 passed)

4 Steps (4 passed)

0m0.112s

ПРИМЕНЕНИЕ ТЕГОВ К HOOKS

- Tagged Hooks – это сочетание Hooks и Tags. Tagged Hooks используются, когда необходимо выполнить какое-то действие не для всех, а только для конкретных Сценариев. Мы добавляем Tag в скобках после Hook, чтобы преобразовать его в tagged Hook.

ПРИМЕНЕНИЕ ОПЕРАЦИЙ AND И OR К TAGGED HOOKS

- Точно так же, как мы применяли операции AND и OR к тегам, мы можем применить операции AND и OR к комбинации Tags и Hooks. Рассмотрим ситуации, когда нам необходимо выполнить определенные Шаги для Функций, например для feature1 и feature2, но не для других Функций. Как это сделать в Cucumber?

СЕКЦИЯ 3: ОПРЕДЕЛЕНИЯ ШАГОВ

ОПРЕДЕЛЕНИЯ ШАГОВ

- В этом модуле мы рассмотрим следующие темы:
 - Выявление дублирующих и двусмысленных определений шагов
 - Использование регулярных выражений для оптимизации определений шагов
 - Использование опциональных групп Capture/Noncapture (с записью / без записи)
 - Преобразование таблиц данных для анализа тестовых данных
 - Реализация Data Table diffs для сравнения таблиц
 - Использование Doc Strings для анализа большого объема данных одной порцией
 - Объединение Doc Strings и Scenario Outlines
 - Определение преобразований String для улучшения конверсии

ОПРЕДЕЛЕНИЯ ШАГОВ

- Когда Cucumber начинает выполнение программы, он ищет все классы на Classpath, которые находятся в указанном связующем пакете (или потомке).
- Определения шагов (Step Definitions) – это небольшая часть кода с прилагаемым паттерном. Паттерн используется для привязки Step Definitions ко всем соответствующим Шагам, и коды будут выполняться Cucumber при обнаружении Gherkin Step.
- Для создания Step Definitions используются аннотации Cucumber – например, @Given, @When и @Then.
- В Шаге Given пользователь находится на странице входа в приложение, текст после ключевого слова Given (user is on Application landing page) соотносится Cucumber с текстом в файле Step Definitions (@Given("^user is on Application landing page\$")). И когда Cucumber обнаруживает соответствие, он

НЕ ОПРЕДЕЛЕННЫЕ ШАГИ

- При первом выполнении файла Feature, Cucumber не нашел никаких соответствий Step Definitions. Именно поэтому Cucumber выдал ошибку Undefined Steps Error.
- Cucumber также выдаст свое Предложение (Suggestion) о том, как освободиться от Объединенных Шагов (Undefined Steps). Все последующие Шаги после Undefined Step в Сценарии пропускаются, и Сценарий отмечается как провальный (Fail).

НЕЗАВЕРШЕННЫЕ ШАГИ (PENDING STEPS)

- Когда мы используем код, предлагаемый Cucumber, в нашем файле Step Definition и выполняем файл Feature, то получаем исключение Pending Steps Exception.
- Исключение Pending Steps Exception возникает из-за следующего кода:
- `throw new PendingException();`
- Когда Cucumber сталкивается с таким выражением, он понимает, что эти Шаги остаются незавершенными.
- Статус сценария - незавершенный, первый шаг с `PendingException()` будет отмечен как незавершенный, и все остальные шаги в этом сценарии будут пропущены.

РЕАЛИЗОВАННЫЕ ШАГИ

- Когда мы заменяем команду `throw` функциональным кодом, ошибка `pending exception` исчезает.
- Теперь статус Шагов будет зависеть от кода, исполняемого в данном Шаге.

ДУБЛИРУЮЩИЕ И ДВУСМЫСЛЕННЫЕ ОПРЕДЕЛЕНИЯ ШАГОВ

- Поскольку Cucumber-JVM ищет все классы на Classpath, которые находятся в указанном связующем пакете (или потомке), есть вероятность дублирования шагов (частичного/полного) в Step Definitions.
- **Дублирующие шаги:** когда Cucumber сталкивается с несколькими полностью одинаковыми определениями шагов, он выдает исключение Duplicate Step Definitions.
- **Двусмысленные шаги:** когда Cucumber сталкивается с несколькими частично одинаковыми определениями шагов, он выдает исключение Ambiguous Step Definitions.

ДУБЛИРУЮЩИЕ И ДВУСМЫСЛЕННЫЕ ОПРЕДЕЛЕНИЯ ШАГОВ

- Если Cucumber сталкивается с дублирующими/двусмысленными шагами, все остальные шаги в таких сценариях пропускаются и эти сценарии отмечаются как провальные (Fail).
- Cucumber даже указывает два объекта, которые вызывают эту ошибку.
- При одном выполнении идентифицируются только первые два случая ошибочных Шагов; если имеется больше случаев дублирования одного и того же Шага, они будут выявлены при следующем выполнении кода.

ШАГИ В CUCUMBER

- Помимо запуска сценариев в Cucumber можно кастомизировать и шаги
- Например, передавая в них класс сценария
- Который позволяет контролировать процесс работы теста

КЛАСС SCENARIO

@After

```
public void  
getScenarioInfo(Senario  
scenario) {  
  
}
```

- getId()
 - Получить идентификатор сценария
- getName()
 - Получить имя сценария
- getStatus()
 - Получить статус сценария
- isFailed()
 - Провалился ли сценарий
- getSourceTagNames()
 - По каким тегам запускался сценарий

АНТИПАТТЕРНЫ

- Не вкладывайте step-ы друг в друга
- Не используйте сложные регулярные выражения
- Не пытайтесь адаптировать step-ы на разных языках программирования
- Не соединяйте сценарии из разных сфер в одном файле
- Не соединяйте в одном сценарии множество условий - пишите разные сценарии

АНТИПАТТЕРНЫ (ПРИМЕР)

Given *I authenticated myself with an enabled card*
And *The available balance in my account is positive*
And *the ATM has enough money*
And *the ATM has enough paper to print receipts*
When *I put the card in the ATM*
And *I input into the keyboard my card's pin*
And *I press the confirm pin button*
And *I press the button next to the option to withdraw money*
And *I enter an amount less than or equal to my available balance*
And *I press the button to confirm the withdrawal*
And *I press the button to print the receipt*

АНТИПАТТЕРНЫ (АЛЬТЕРНАТИВА)

Scenario: *As an existing and enabled ATM user, I want to make a withdrawal to get money.*

Given *I authenticated with a card enabled*

And *The available balance in my account is positive*

When *I select the option to withdraw money*

And I *enter the amount of money that is less than the amount I have available and the ATM's available balance*

Then *I get the money*

And *The money I get is subtracted from the available balance of my account*

And *The system returns the card automatically*

And *The system displays the transaction completed message*

Спасибо за внимание!