**Cucumber**

1. В Maven добавить зависимости:

<**dependency**>  
 <**groupId**>io.cucumber</**groupId**>  
 <**artifactId**>cucumber-java</**artifactId**>  
 <**version**>7.0.0</**version**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>io.cucumber</**groupId**>  
 <**artifactId**>cucumber-junit-platform-engine</**artifactId**>  
 <**version**>7.0.0</**version**>  
 <**scope**>test</**scope**>  
</**dependency**>

//Для многопоточного запуска тестов

<**dependency**>  
 <**groupId**>org.junit.platform</**groupId**>  
 <**artifactId**>junit-platform-suite-engine</**artifactId**>  
 <**version**>1.9.3</**version**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>org.junit.platform</**groupId**>  
 <**artifactId**>junit-platform-suite</**artifactId**>  
 <**version**>1.10.0</**version**>  
 <**scope**>test</**scope**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>org.junit.platform</**groupId**>  
 <**artifactId**>junit-platform-suite-api</**artifactId**>  
 <**version**>1.10.0</**version**>  
 <**scope**>compile</**scope**>  
</**dependency**>

+ остальное, что будет предлагать автоматически добавить ***Intellij Idea***!

2. В папке “src/test/resources” *(если нет, то создать)* или ее подпапках создать файл/ы с расширением “.feature”. В них написать описание того, что будет тестироваться и как, используя структуру:

**Feature:** - тут пишем, то что будет делать класс (и именно с таким названием он автоматически создаст класс)

**Background:** - описывает какие то шаги, которые будут выполняться только ПЕРЕД каждым сценарием

***Given***

***When***

***Then***

**Scenario/Scenario Outline:** - описываем шаги каждого сценария. “Scenario Outline” применяем если тут будет использован блок “Examples”.

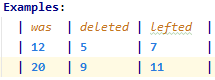
***Given***

***When***

***- // -***

***Then***

***Examles:*** - тут создается маленькая БД для заполнения разными наборами данных для многократного прогона данного сценария



Scenario/Scenario Outline:

и т.д.

При этом IntellijIdea будет предлагать автоматически создать пустые шаблоны класса (с названием как написано в блоке “Feature”) с методами (написанными в блоках *“Background”* u *“Scenario”/”Scenario Outline”*).

***@Before*** *(cucumber.api.java.Before)*, ***@After*** *(cucumber.api.java.After)*– для каждой из этих аннотаций, пишутся отдельные классы с методом/ами *(аннотированым/ми этими аннотацией/ями)*. Причем вначале выполняется код написанный в этих аннотациях и только потом код написанный в блоке “Background”.

Когда в классе несколько подряд методов с аннотацией ***@Before*** или ***@After*** то запускает начиная с нижней к верхней.

Кроме того все эти ***@Before/@After*** распространяются не только на конкретный файл “*\*\*\*.feature*”, а вообще на любой “*\*\*\*.feature*” из соседних папок (подтягивает сволочь все ***@Before/@After отовсюду***)

3. Для каждого файла с расширением “.feature” в папке “src/test/java” или ее подпапках создаем отдельный Java-файл с кодом, реализующим пункты, написанные в этом “.feature” файле.

Java-файл с пустыми методами и связанный с файлом с расширением “.feature”, формируется из самого файла с расширением “.feature”. В нем наводимся мышкой на любой пунктов блоков “Scenario”/”Scenario Outline” или “Background” и тогда выпадет предложение создать файл либо с одним конкретным методом или со всеми методами.

**!!!! В моей IntellijIdea не создается нормально автоматически новый файл с нормально прописанным Given/When/Then и т.д. Нужно вначале в “криво” созданном файле удалить всю внутренность, потом написать минимальный пустой метод “public void xxx(){}” с аннотацией @Given(заполненной ”” внутри скобок) и только потом опять автоматически создать метод уже в этом заранее созданном файле. После чего удалить пустой метод !!!!!!**

4. Запуск тестов в однопоточном режиме производится из каждого файла с расширением “.feature”.

5. Для многопоточного запуска тестов в папке “src/test/java” или ее подпапках создать Java-файл с любым именем, загружающий настройки из файла ***“junit-platform.properties”*** и запускающий с этими настройками тестирование всех тестов из всех файлов с расширением “.feature” и имеющий вид:

@Suite  
@IncludeEngines(**"cucumber"**) – наименование движка  
@SelectClasspathResource(**"/"**) – место расположение файла “junit-platform.properties” относительно папки “src/test/resources”  
**public class** ConfigPropertyRunner {  
}

- при этом все содержимое ненужных файлов с расширением “.feature” закомментировать!

6. Файл настроек ***“junit-platform.properties”*** должен находиться обязательно в папке ***“src/test/resources”*** или ее подпапках.

В этом файле также могут быть написаны настройки для многопоточного чистого запуска тестов *(не для cucumber, а файлов содержащих в названиях слово “Test”)* с помощью команды *“mvn test”*.

Сами настройки описаны в файле ***“Cucumber все дополнительные функциию.doc”***.