

Изучение подходов к  
построению тестов (TDD,  
BDD, DDT)



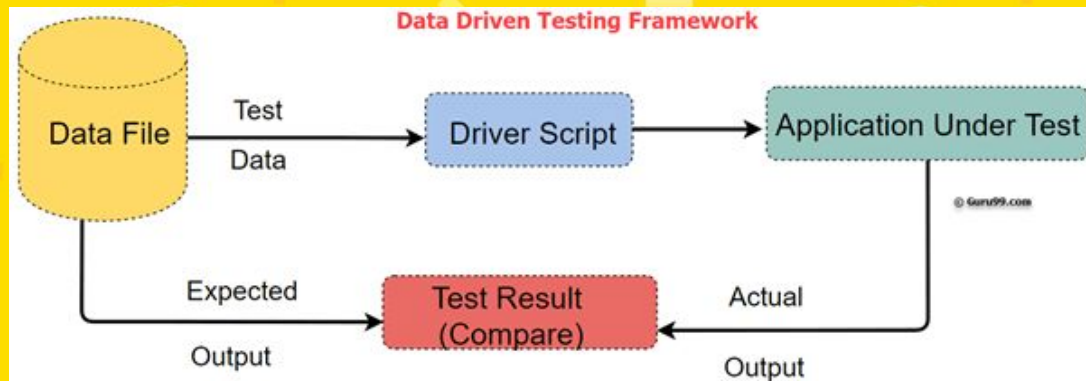
# Data-driven testing

**DDT (Data-Driven Testing)** — это подход к автоматизированному тестированию, при котором тестовые данные хранятся отдельно от тестовых сценариев — в таблицах, файлах или базах данных.

Один и тот же тестовый скрипт выполняется повторно с разными наборами входных данных, что позволяет покрыть множество сценариев позитивного и негативного тестирования.

Где хранятся данные:

- Excel-файлы (.xls, .xlsx)
- CSV-файлы
- XML / JSON
- Базы данных
- Встроенные коллекции (массивы, списки)



# Test-driven development

**TDD, test-driven development** или **разработка через тестирование** — это методология разработки ПО, которая основывается на повторении коротких циклов разработки: изначально пишется тест, покрывающий желаемое изменение, затем пишется программный код, который реализует желаемое поведение системы и позволит пройти написанный тест, а затем проводится рефакторинг написанного кода с постоянной проверкой прохождения тестов.

# Behavior-driven development

**BDD (сокр. от англ. Behavior-driven development, дословно «разработка через поведение»)** — это методология разработки программного обеспечения, являющаяся ответвлением от методологии разработки через тестирование (TDD). Основной идеей данной методологии является совмещение в процессе разработки чисто технических интересов и интересов бизнеса, позволяя тем самым управляющему персоналу и программистам говорить на одном языке. Для общения между этими группами персонала используется предметно-ориентированный язык, основу которого представляют конструкции из естественного языка, понятные неспециалисту, обычно выражающие поведение программного продукта и ожидаемые результаты. Считается, что данный подход эффективен, когда предметная область, в которой работает программный продукт, описывается очень комплексно.

# Немного об огурцах...

cucumber



<https://cucumber.io/docs/cucumber/>



# Пример

Feature: Guess the word

# The first example has two steps

Scenario: Maker starts a game

When the Maker starts a game

Then the Maker waits for a Breaker to join

# The second example has three steps

Scenario: Breaker joins a game

Given the Maker has started a game with the word "silky"

When the Breaker joins the Maker's game

Then the Breaker must guess a word with 5 characters



# Ключевые слова



Ключевые слова:

- i. Feature
- ii. Scenario
- iii. Given
- iv. When
- v. Then
- vi. And/But
- vii. Scenario Outline/Examples
- viii. Background



Документация: <https://cucumber.io/docs/gherkin/reference/>



# Практика



<https://github.com/borodicht/WikipediaSearch.git>

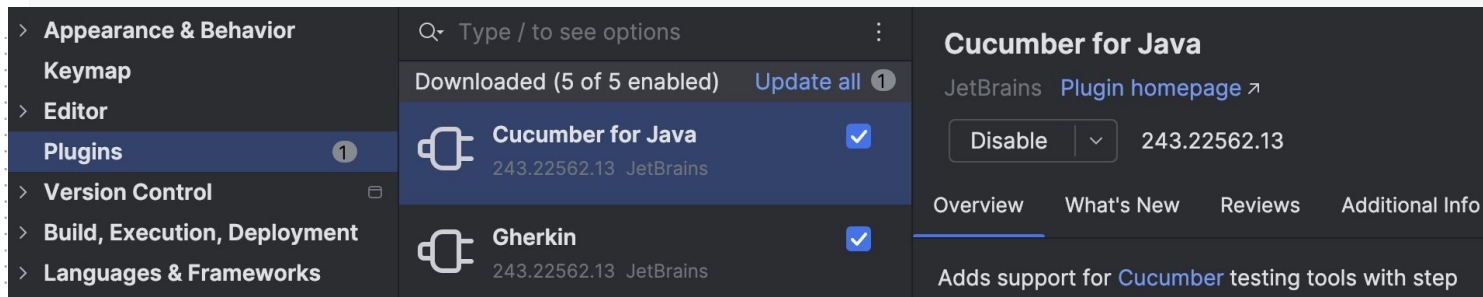


# Установка



1

Добавить плагины в IntelliJ IDEA: Cucumber for Java, Gherkin



2

Добавить зависимости в pom.xml

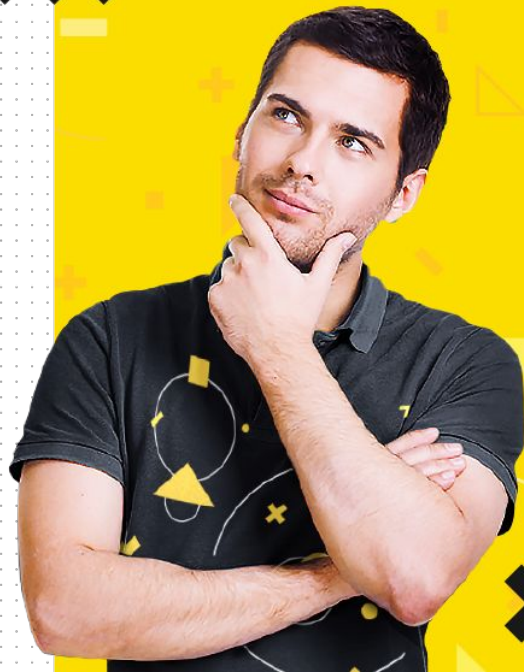
# Зависимости

```
<properties>  
  <cucumber.version>7.23.0</cucumber.version>  
</properties>
```

```
<dependency>  
  <groupId>io.cucumber</groupId>  
  <artifactId>cucumber-core</artifactId>  
  <version>${cucumber.version}</version>  
</dependency>
```

```
<dependency>  
  <groupId>io.cucumber</groupId>  
  <artifactId>cucumber-java</artifactId>  
  <version>${cucumber.version}</version>  
</dependency>
```

```
<dependency>  
  <groupId>io.cucumber</groupId>  
  <artifactId>cucumber-testng</artifactId>  
  <version>${cucumber.version}</version>  
</dependency>
```



**До встречи на следующем занятии!!!**