

# ChatGPT

 chat.openai.com/c/c96175a5-3e0a-41fc-936d-91025c51fcb1

## ChatGPT

JUnit 5 is a powerful testing framework for Java applications, and it introduces several annotations to facilitate test development. Here are some important annotations in JUnit 5:

### 1. **@Test:**

- Marks a method as a test method.
- Example:

```
java

@Test
void myTestMethod() {
    // Test logic here
}
```

### 2. **@BeforeAll:**

- Used to signal that the annotated method should be executed before all tests in the test class.
- Example:

```
java

@BeforeAll
static void setup() {
    // Initialization logic here
}
```

### 3. **@BeforeEach:**

- Indicates that the annotated method should be executed before each test method in the test class.
- Example:

```
java

@BeforeEach
void init() {
    // Initialization logic here
}
```

### 4. **@AfterEach:**

- Specifies that the annotated method should be executed after each test method in the test class.
- Example:

```
java

@AfterEach
void cleanup() {
    // Cleanup logic here
}
```

### 5. **@AfterAll:**

- Denotes that the annotated method should be executed after all tests in the test class.
- Example:

```
java

@AfterAll
static void tearDown() {
    // Tear-down logic here
}
```

## 6. **@DisplayName:**

- Provides a custom display name for a test class or method.
- Example:

```
java

@Test
@DisplayName("My Custom Test Name")
void myTestMethod() {
    // Test logic here
}
```

## 7. **@Disabled:**

- Marks a test class or method as disabled or ignored.
- Example:

```
java

@Test
@Disabled("This test is currently disabled")
void disabledTest() {
    // Test logic here
}
```

## 8. **@RepeatedTest:**

- Specifies that the annotated method should be executed a specified number of times.
- Example:

```
java

@RepeatedTest(3)
void repeatedTest() {
    // Test logic here
}
```

## 9. **@ParameterizedTest:**

- Indicates that the annotated method should be executed with different sets of parameters.
- Example:

java

```
@ParameterizedTest
@ValueSource(ints = {1, 2, 3})
void parameterizedTest(int value) {
    // Test logic here with parameter 'value'
}
```

## 10. **@Nested:**

- Allows the creation of nested test classes.
- Example:

java

```
@Nested
class InnerTestClass {
    // Nested test methods and classes here
}
```

These are some of the key annotations provided by JUnit 5 to support various testing scenarios. Keep in mind that JUnit 5 offers more annotations and features, so it's recommended to refer to the official documentation for comprehensive information.