С версии JUnit4 тесты создаются с помощью аннотаций.

Список аннотаций:

**@Test**

**@Before @After**

**@BeforeClass @AfterClass**

**@Ignore**

**@Rule**

**@Parameterized**

Бывает такое, что для выполнения каждого тестового сценария вам необходим некоторый контекст, например, заранее созданные экземпляры классов. А после выполнения нужно освободить зарезервированные ресурсы. В этом случае вам понадобятся аннтоации @Before и @After. Метод, помеченный @Before будет выполняться перед каждым тестовым случаем, а метод, помеченный @After - после каждого тестового случая.

Если же инициализацию и освобождение ресурсов нужно сделать всего один раз - соответственно до и после всех тестов - то используйте пару аннотаций @BeforeClass и @AfterClass.

@Test — указывает на тестирующий метод.

Если вы хотите указать, что определенный тест необходимо пропустить, то пометьте его аннотацией @Ignore. Хотя можно просто удалить аннотацию @Test.

@Rule – здесь можно указать правила для всех тестов (нужно унаследовать TestRule, пример:

**public TestRule timeout = new Timeout(100);**)

@Parameterized – здесь можно указать параметры, при указании параметров – нужно добавить перед классом аннотацию **@RunWith(Parameterized.class)**, она укажет классу что нужно начинать работу теста с указанными параметрами. Пример:

**@RunWith(Parameterized.class)**

**public class MyMathTest{**

**int a, b, c;**

public **MyMathTest(int a, int b, int c){**

**this.a=a;**

**this.b=b;**

**this.c=c;**

**}**

**public void testAdd() trows InterruptedException{**

**int result = MyMath.add(a, b);**

**assertEquals(c, result);**

**}**

**@Parameterized.Parameters**

**Public static Collection numbers(){**

**return Arrays.asList(new Object [] [] {{1,2,3}, {5,5,10}, {10,5,15}});**

**}**

Здесь указывается многократность выполнения тестов. К примеру тестируется сумирование, первак цифра сумируется со второй, третья это ожидаемый результат (1+2=3, 5+5=10, 10+5=15).

**“@RunWith” и “@Suite”** аннотации используются для запуска тест сьюта. Класс, написанный ниже, означает, что классы “JunitTest1” и “JunitTest2” будут запущены после запуска JunitTest5. Все остальные декларации будут внутри класса.

**@RunWith(Suite.class)**

**@Suite.SuiteClasses({**

**JunitTest1.class,**

**JunitTest2.class**

**})**

**public class JunitTest5 {**

**}**

У аннотации могут быть проставлены такие параметры:

**expected** – здесь можно указать какое исключение мы ожидаем

(пример: **@Test (expected = AriphmeticException.class)**

**timeout** - через какое время в милисекундах прекратить выполнение теста и засчитать его как неуспешный.

Создание теста в IDE:

Правой клавишей мыши кликаем на проекте, выбираем создание теста.

Выбираем нужное:

1. методы доступа
2. генерацию кода
3. генерацию комента

Хорошее видео <https://www.youtube.com/watch?reload=9&v=z9jEVLCF5_w>