Упражнение 1  
Unit-тестирование

## Проект

calculator

## Длительность

Самостоятельная работа: 30 мин

Обсуждение: 5 мин

## Цели упражнения

Ознакомиться с базовыми принципами и возможностям JUnit. Научиться покрывать функционал unit-тестами.

## Описание

Любой функционал зависит от других частей проекта. Эти части могут меняться и функционал может при этом «сломаться». Чтобы регулярно проверять корректность постоянного функционала- существуют т.н. unit-тесты. Самым популярным фреймворком unit-тестирования является JUnit.

Вам дан класс простейшего калькулятора. «Покройте» его функционал unit-тестами с помощью JUnit.

## Задачи упражнения

1. Создать методы инициализации калькулятора.
2. Вызвать методы калькулятора.
3. Сверить полученные результаты с ожидаемыми.

## Подробное руководство

### Часть 1. Создание класса Unit-теста

1. Откройте проект calculator;
2. Перейдите в класс Calculator. Изучите его методы;
3. Встаньте курсором на имя класса. Кликните на значок лампы и выберите – Create Unit test
4. В появившемся диалоговом окне- выберите вид JUnit фреймворка- 5. Нажмите ОК
5. Перейдите в сгенерированный класс.
6. В классе CalculatorTest- создайте метод инициализации калькулятора

@BeforeAll  
**public static void** init(){  
 *calculator* = **new** Calculator();  
}

1. В классе CalculatorTest- создайте метод тестирования суммации

@Test  
**public void** testSum() {  
 **int** result = *calculator*.sum(2, 2);  
 **if** (result != 4) { *// if 2 + 2 != 4* Assert.*fail*();  
 }  
}

1. В классе CalculatorTest- создайте метод тестирования минуса

@Test  
**public void** testMinus() {  
 Calculator calculator = **new** Calculator();  
 Assert.*assertEquals*(0, calculator.minus(2, 2));  
}

1. В классе CalculatorTest- создайте метод тестирования деления

@Test  
**public void** testDivide() {  
 Calculator calculator = **new** Calculator();  
 Assert.*assertEquals*(2, calculator.divide(6, 3));  
}

1. В классе CalculatorTest- создайте метод тестирования деления c проверкой на реакцию деления на ноль

@Test(expected = ArithmeticException.**class**)  
**public void** testDivideWillThrowExceptionWhenDivideOnZero() {  
 Calculator calculator = **new** Calculator();  
 calculator.divide(6, 0);  
}

1. Запустите тесты и пронаблюдайте результат