**Модульное тестирование**

Дано приложение, определяющее «тип» треугольника (равносторонний, равнобедренный, прямоугольный, «обычный») и предоставляющее возможность вычислить площадь треугольника. См. код в архиве «**Triangle.zip**».

Задание:

1. Написать чек-лист для тестирования данного приложения.
2. С помощью JUnit или TestNG (на ваш выбор) реализовать те пункты чек-листа, которые возможно реализовать в модульном тестировании.

После завершения работы поместите целиком ваш проект в папку familiya\_imya (где familiya\_imya – ваши фамилия и имя на английском в нижнем регистре).

**Чек-лист**

**В данном чек-листе в качестве параметров принимаются переменные a, b, c**

**В данном чек листе будут осуществлены проверки на ввод корректных данных (см. пункты 1-16) и на корректное определение типа треугольника (см. пункты 17-25)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Пункты чек листа** | **Название теста** | **Ожидаемый результат** | **Результат**  **выполнения теста** |
| 1. Передача в конструктор в качестве параметров корректных (положительных, разных, сумма двух параметров больше третьего) чисел и вызов метода проверки checkTriangle(). | checkTriangle\_Correct\_tst | Метод проверки checkTriangle() вернёт true | + |
| 1. Передача в конструктор в качестве параметра **а** некорректного (нулевого) числа, остальные параметры корректны (положительны, различны, сумма двух параметров больше третьего) и вызов метода проверки checkTriangle(). | checkTriangle\_Zero\_A\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор в качестве параметра **b** некорректного (нулевого) числа, остальные параметры корректны (положительны, различны, сумма двух параметров больше третьего) и вызов метода проверки checkTriangle(). | checkTriangle\_Zero\_B\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор в качестве параметра **c** некорректного (нулевого) числа, остальные параметры корректны (положительны, различны, сумма двух параметров больше третьего) и вызов метода проверки checkTriangle(). | checkTriangle\_Zero\_C\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор в качестве параметра а некорректного (отрицательного) числа, остальные параметры корректны (положительны, различны, сумма двух параметров больше третьего) и вызов метода проверки checkTriangle(). | checkTriangle\_Negative\_A\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. 8 Передача в конструктор в качестве параметра b некорректного (отрицательного) числа, остальные параметры корректны (положительны, различны, сумма двух параметров больше третьего) и вызов метода проверки checkTriangle(). | checkTriangle\_Negative\_B\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор в качестве параметра c некорректного (отрицательного) числа, остальные параметры корректны (положительны, различны, сумма двух параметров больше третьего) и вызов метода проверки checkTriangle(). | checkTriangle\_Negative\_C\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a+b>c, при этом числа должны быть положительны, и вызов метода проверки checkTriangle(). | checkTriangle\_Three\_More\_C\_tst | Метод проверки checkTriangle() вернет true | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a+c>b, при этом числа должны быть положительны, и вызов метода проверки checkTriangle(). | checkTriangle\_Three\_More\_B\_tst | Метод проверки checkTriangle() вернет true | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы b+c>a, при этом числа должны быть положительны, и вызов метода проверки checkTriangle(). | checkTriangle\_Three\_More\_A\_tst | Метод проверки checkTriangle() вернет true | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a+b<c, при этом числа должны быть положительны, и вызов метода проверки checkTriangle(). | checkTriangle\_Three\_Less\_C\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a+c<b, при этом числа должны быть положительны, и вызов метода проверки checkTriangle(). | checkTriangle\_Three\_Less\_B\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы b+c<a, при этом числа должны быть положительны, и вызов метода проверки checkTriangle(). | checkTriangle\_Three\_Less\_A\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a+b=c, при этом числа должны быть положительны, и вызов метода проверки checkTriangle(). | checkTriangle\_Three\_Equal\_C\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a+c=b, при этом числа должны быть положительны, и вызов метода проверки checkTriangle(). | checkTriangle\_Three\_Equal\_B\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы b+c=a, при этом числа должны быть положительны, и вызов метода проверки checkTriangle(). | checkTriangle\_Three\_Equal\_A\_tst | Метод проверки checkTriangle() вернет false | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a\*a+b\*b == c\*c, при этом числа должны быть положительны, и вызов метода определения detectTriangle. | detectTriangle\_Restangular\_1\_tst | Метод определения detectTriangle вернет 8 | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы b\*b + c\*c == a\*a, при этом числа должны быть положительны, и вызов метода определения detectTriangle. | detectTriangle\_Restangular\_2\_tst | Метод определения detectTriangle вернет 8 | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a\*a + c\*c == b\*c, при этом числа должны быть положительны, и вызов метода определения detectTriangle. | detectTriangle\_Restangular\_3\_tst | Метод определения detectTriangle вернет 8 | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a==b && b==c && a==c, при этом числа должны быть положительны, и вызов метода определения detectTriangle. | detectTriangle\_Equilateral\_tst | Метод определения detectTriangle вернёт 0 | - |
| 1. Передача в конструктор трех параметров таким образом, чтобы a==b, при этом числа должны быть положительны, и вызов метода определения detectTriangle. | detectTriangle\_Isosceles\_1\_tst | Метод определения detectTriangle вернёт 2 | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы b==c, при этом числа должны быть положительны, и вызов метода определения detectTriangle. | detectTriangle\_Isosceles\_1\_tst | Метод определения detectTriangle вернёт 2 | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a==c, при этом числа должны быть положительны, и вызов метода определения detectTriangle. | detectTriangle\_Isosceles\_1\_tst | Метод определения detectTriangle вернёт 2 | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы выполнялись одновременно пункт из 17-18-19 пунктов и пункт из 21-22-23 пунктов | detectTriangle\_Isosceles\_Restangular\_tst | Метод определения detectTriangle вернёт 0 | - |
| 1. Передача в конструктор трех параметров таким образом, чтобы они были положительные, разные и не выполнялось ни одно из условий равносторонности, равнобедренности или прямоугольности, при этом числа должны быть положительны, и вызов метода определения detectTriangle. | detectTriangle\_Ordinary\_tst | Метод определения detectTriangle вернет 4 | + |
| 1. Передача в конструкто р в качестве параметров корректных (положительных, разных, сумма двух параметров больше третьего) чисел (a=15, b=12, c=14) и вызов метода вычисления площади getSquare. | getSquare\_Correct\_tst | Метод вычисления площади getSquare вернет 78.926786 ± 0.000001 | + |
| 1. Передача в конструктор в качестве параметра **а** некорректного (нулевого) числа, остальные параметры корректны (положительны, различны, сумма двух параметров больше третьего), (a=0, b=12, c=14) и вызов метода вычисления площади getSquare. | getSquare\_Zero\_A\_tst | Метод вычисления площади getSquare вернет 0 | - |
| 1. Передача в конструктор трех параметров таким образом, чтобы a==b && b==c && a==c, при этом числа должны быть положительны, сумма двух параметров больше третьего (a=b=c=5), и вызов метода вычисления площади getSquare. | getSquare\_Equilateral\_tst | Метод вычисления площади getSquare вернет 10.825318 ± 0.000001 | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a\*a+b\*b == c\*c, при этом числа должны быть положительны, сумма двух параметров больше третьего (a=3,b=4,c=5), и вызов метода вычисления площади getSquare. | getSquare\_Rectangular\_tst | Метод вычисления площади getSquare вернет 6 ± 0.000001 | + |
| 1. Передача в конструктор трех параметров таким образом, чтобы a==b, при этом числа должны быть положительны, сумма двух параметров больше третьего (a=2,b=2,c=3),и вызов метода вычисления площади getSquare. | getSquare\_Isosceles\_tst | Метод вычисления площади getSquare вернет 1.984313 ± 0.000001 | + |
| 1. Передача в конструктор в качестве параметра **а** некорректного (отрицательного) числа, остальные параметры корректны (положительны, различны, сумма двух параметров больше третьего), (a=-15,b=12,c=14) и вызов метода вычисления площади getSquare. | getSquare\_Negative\_A\_tst | Метод вычисления площади getSquare вернет 0 | - |