**Модульное тестирование Triangle**

Классы эквивалентности: Min.Double - Max.Double

**Корректные значения** – значения вида :

Все стороны > 0 и удовлетворяют условию: сумма двух сторон > третьей **.**

Считаем , что методы detectTriangle() и getSquare() не должны производить проверку значений на корректность.

Т.к. в программе не предусмотрено обработки исключений , будем считать , что исключения являются классом Exception

Тестирование метода **checkTriangle():**

**Позитивное тестирование :**

|  |  |
| --- | --- |
| **Сценарий** | **Ожидаемый результат** |
| **1.Вводим значения в пределах класса эквивалентности . Все стороны > 0 и удовлетворяют условию: сумма двух сторон > третьей .** | **true** |
| **2. Вводим корректные значения (см. п. 1) равные Double.MaxValue ,**  **Значения равные Double.MinValue** | **true** |
| **3.Вводим корректные значения (см. п. 1) приближённые к Double.MaxValue и Double.MinValue** | **true** |

**Негативное тестирование**

|  |  |
| --- | --- |
| **Сценарий** | **Ожидаемый результат** |
| **Вводим отрицательные или нулевые или комбинация отрицательных и нулевых значений** | **false** |
| **Вводим значения больше Double.Max , меньше Double.Min** | **Исключение** |
| **Вводим значение а < 0 , a = 0** | **gertMessage() возвращает “a <=0”** |
| **Вводим значение b < 0 , b = 0** | **gertMessage() возвращает “b <=0”** |
| **Вводим значение c < 0, c = 0** | **gertMessage() возвращает “c <=0”** |
| **Вводим значения так: a + b < c, a + b = c** | **gertMessage() возвращает “a + b <=c”** |
| **Вводим значения так: a + c < b, a + c = b** | **gertMessage() возвращает “a + c <= b”** |
| **Вводим значения так: b + c < a, b + c = a** | **gertMessage() возвращает “b + c <=a”** |

Тестирование метода **detectTriangle(). Считаем , что этот метод не должен проводить проверку на правильность введённых значений**

**Позитивное тестирование :**

|  |  |
| --- | --- |
| **Сценарий** | **Ожидаемый результат** |
| **Вводим корректные значения для равностороннего треугольника. Делаем проверку на то , что a=b=c** | **Выводится 3** |
| **Вводим корректные значения для равнобедренного треугольника. Делаем проверку на то , что a=b || a=c || b=c** | **Выводится 2** |
| **Вводим корректные значения для прямоугольного треугольника Делаем проверку на то , что a^2 + b^2 = c^2||b^2 + c^2=a^2 || c^2 + a^2 = b^2** | **Выводится 8** |
| **Вводим корректные значения для обычного треугольника** | **Выводится 4** |
| **Вводим корректные значения для равнобедренного прямоугольного треугольника (Math.sqrt(32),Math.sqrt(32),8.0)** | **Выводится 10** |
| **Вводим корректные значения равные Double.MaxValue, Double.MinValue** | **Выводится соответсвующий тип треугольника** |
| **Вводим корректные значения приближённые к Double.MaxValue и Double.MinValue** | **Выводится соответсвующий тип треугольника** |

**Негативное тестирование :**

|  |  |
| --- | --- |
| **Сценарий** | **Ожидаемый результат** |
| **Вводим корректные значения больше Double.MaxValue , Меньше Double.MinValue** | **Исключение** |

Тестирование метода **getSquare() . Считаем , что этот метод не должен проводить проверку на правильность введённых значений**

**Позитивное тестирование:**

|  |  |
| --- | --- |
| **Сценарий** | **Ожидаемый результат** |
| **Вводим корректные целые значения (3,4,5)** | **Результат = 6** |
| **Вводим корректные дробные значения (0.1,0.1,0.1)** | **Результат =** 0.00433012701  (Для вычислений использовался инженерный калькулятор) |
| **Вводим значения приближённые к Double.MaxValue (10E100,10E80,10E100)** | **Число** |
| **Вводим отрицательные значения , нулевые,**  **Комбинация отрицательных и нулевых** | **NaN** |