**Чек — лист для тестирования приложения «Треугольник».**

Передача исходных данных (набор из трех сторон) предусмотрена через конструктор. Создание объекта и дальнейшая работа приложения возможна только при передаче в конструктор трех чисел типа double. Поэтому тестирование работы приложения при передаче неверного количество входных данных и и неверного типа данных не требуется. Для тестирования работы приложения необходимо:

1. Проверить корректность работы метода checkTriangle()). Метод должен возвращать **true** при одновременном выполнении двух условий:
2. Входные параметры являются положительными числами >0.
3. Сумма двух любых сторон больше третьей (3 тест-кейса).

Проверить, checkTriangle()) возвращает **false** в случаях если:

1. Любая из сторон равна нулю, а две другие положительные числа — 3 тест-кейса
2. Все стороны равны нулю -1 тест-кейс.
3. Любая из сторон отрицательное число, а остальные две положительные числа — 3 тест-кейса.
4. Две стороны отрицательные числа, третья — положительное число — 3 тест-кейса.
5. Три стороны - отрицательные числа -1 тест-кейс.
6. Сумма двух сторон меньше третьей — 3 тест-кейса.
7. Одна сторона равна нулю, вторая отрицательное число, третье положительное число — 3 тест-кейса.
8. Проверить, что метод checkTriangle()) выбрасывает исключение при выполнении любого их условий:
9. Отношение наибольшей стороны треугольника к наименьшей превышает 10 16. В противном случае из-за ограниченной точности типа double выполнение отдельных математических операций в Java будет некорректно. Например результатом сравнения10 18 + 10 и 10 18 будет логическое **true**.
10. В случае превышение максимально допустимого значения для double при выполнения математических операций внутри метода. Проверить, что исключение выбрасывается при нарушении любого из неравенств):

a+b <=Double.MaxValue();

b+c <=Double.MaxValue();

a+c <=Double.MaxValue().

1. Проверить корректность определения типа треугольника (метод detectTriangle()):
2. Проверить, что если метод detectTriangle() вызывается у объекта, который не является треугольником, он выбрасывает исключение.
3. Проверить, что если отношение наибольшей стороны треугольника к наименьшей превышает 10 16 метод detectTriangle()бросает исключение;
4. Проверить корректность определения равностороннего треугольника\*.
5. Проверить корректность определения прямоугольника треугольника\*.
6. Проверить корректность определения равнобедренного треугольника\*.
7. Проверить корректность определения прямоугольного равнобедренного треугольника\*.
8. Проверить корректность определения обычного треугольника (не являющимся ни одним из перечисленных в п. с) – e) )\*.
9. Проверить отсутствие превышения максимально допустимого значения для double при выполнения математических операций внутри метода. Убедиться, что в случае невыполнения любого из этих неравенств, метод detectTriangle() выбрасывает исключение:

a2+b2 <= Double.MaxValue();

c2 < =Double.MaxValue();

где **с** -наибольшая сторона треугольника, **a** и **b –** другие стороны треугольника.

*\*Примечание: проверки правильности определения всех типов треугольника необходимо произвести с различными наборами величин сторон (стороны меньше единицы, стороны больше единицы, граничные значения по условию не переполнения double). Для каждого набора величин нужно выполнить три тест-кейса.**Например для набора 3, 4, 5 тест-кейсы будут следующими:*

*a=3, b=4, c=5 ; a=4, b=5, c=3; a=5 b=3, c=4*

4. Проверка корректности определения площади треугольника (метод getSquare()):

1. Проверить, что если метод getSquare() вызывается у объекта, который не является треугольником, он выбрасывает исключение;
2. Проверить, что если отношение наибольшей стороны треугольника к наименьшей превышает 10 16 , метод getSquare()бросает исключение;
3. Проверить правильность определения площади для трех сторон которые образуют треугольник (метод checkTriangle() возвращает **true)**. Проверку выполнить для различных наборов данных, аналогично как указано в примечании к методу detectTriangle().
4. Проверить отсутствие превышения максимально допустимого значения для double при выполнения математических операций внутри метода. Убедиться, что в случае невыполнения любого из этих неравенств, метод getSquare() выбрасывает исключение:

a+b+c <=Double.MaxValue();

p(p-a)(p-b)(p-c) <=Double.MaxValue().

5. Проверить, что определение типа треугольника (метод detectTriangle()) возможно только для трех сторон, которые образуют треугольник (метод checkTriangle() возвращает **true**).

6. Проверить, что определение площади треугольника (метод getSquare()) возможно только для трех сторон, которые образуют треугольник (метод checkTriangle() возвращает **true**).

7. Тестирование производительности приложения. Убедиться что время создания копии треугольника и последовательного выполнению трех методов класса Triangle

checkTriangle() →detectTriangle() → getSquare() для позитивного и негативных сценариев не превышают 0.2 секунды (для каждого сценария). Цифра 0.2 секунды в качестве ориентира для прохождения дымового тестирования. Она служит индикатором, что в приложении нет критических проблем со скоростью работы.

8. Тестирование безопасности. Убедиться что нет прямого доступа к состоянию объекта (все переменные имеют модификатор доступа **private**).

Выполнил: Раик Евгений, TAT21.