

План тестирования приложения «Треугольник»

Андрей Сандлер, 097 гр. МФТИ

Для тестирования работы приложения нужно запустить его на наборе тестов, покрывающем все классы эквивалентности длин сторон треугольника. Ниже перечислены выделенные классы:

- Неправильное число аргументов (*)
- Хотя бы одна сторона нулевая (*)
- Отрицательная длина сторон(ы) (*)

(далее все стороны считаем положительными)

- Равносторонний треугольник
- Равнобедренный треугольник
- Прямоугольный треугольник
- Прямоугольный равнобедренный треугольник
- Несуществующий треугольник (*)
- Несуществующий треугольник с двумя одинаковыми сторонами (*)
- Треугольник с нецелыми длинами сторон (*)
- Треугольник с длинами сторон, близкими к погрешности измерений
- Треугольник с очень большими сторонами (*)
- Обычный треугольник с маленькими сторонами

Для классов эквивалентности (*) нужно сделать группу тестов, и считать её не пройденной, если хотя бы один тест из группы не пройден.

Для удобного и быстрого тестирования сразу на многих входах был написан простой скрипт `run.rb` на Ruby, который парсит файл с тестами и запускает приложение `triangle` с заданным набором аргументов. Формат файла `tests` несложный: все строки, начинающиеся с символа «#», считаются комментариями, остальные подаются на вход программе как аргументы. Файл `tests` должен лежать там же, где и скрипт `run.rb`. Результаты тестирования появятся в файле `answer`, который выглядит следующим образом (отрывок):

```
...
68 4    4    5          <--- текущий тест
69 Треугольник существует <--- ответ программы
70 Равнобедренный треугольник <--- ответ программы
71 ----- <--- место для вердикта
72 ----- <--- (заполняется тестирующим)
73 1    1    1.414213562
74 Треугольник существует
75 Равнобедренный треугольник
76 Прямоугольный треугольник
77 -----
78 -----
```

Между двух строк из символов «-» тестировщик вписывает вердикт (правильно или неправильно отработала программа, а также замечания и (по желанию) правильный ответ)

Поскольку ответ, полученный на полном наборе тестов, а также мой собственный вердикт занимают вместе много строк (по сравнению с этим документом), то я его приложил отдельно в файле `testing_answer_asandler_097.pdf`.

Тестировщик при желании может написать заново / дописать новые тесты в файл `tests`.

Для того, чтобы запустить автоматическое тестирование, нужно установить Ruby 2.0 на компьютер и сказать в командной строке:

```
>ruby run.rb
```

Если Ruby не ставится, можно запустить руками каждый тест (будет долго).