



Урок 14. Классификатор треугольников

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 14 из 33 · 45 минут · Модуль «Повторение и обобщение: марафон по ветвлениям»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Три одинаковых числа 0 нельзя назвать равносторонним треугольником. Сначала выясним, существует ли фигура, и только потом дадим ей название.

Результат урока: Проверить допустимость сторон и классифицировать треугольник по равенству сторон, сохранив правильный порядок проверок.

01. Проблемная ситуация

0–5 минут. Даны тройки (3,3,3), (0,0,0), (1,2,3), (3,4,5). Какие можно собрать из отрезков в невырожденный треугольник? Объясните, что происходит при $1+2=3$: получается прямая, а не треугольник.

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Сначала существование, затем вид

Контракт: три целые длины в условных единицах, каждая $1 \dots 1000$. Нарушение диапазона — error. Отсортируем стороны $x \leq y \leq z$. Для невырожденного треугольника нужно $x+y > z$; равенство и меньшая сумма дают no triangle. Другие две суммы автоматически больше противоположной стороны, потому что стороны положительны.

Среди допустимых треугольников сначала проверяем все три равные стороны: equilateral. Затем хотя бы одну равную пару: isosceles. Иначе scalene. В геометрии равносторонний является частным случаем равнобедренного; здесь программа выдаёт наиболее конкретную из трёх взаимоисключающих меток.

Сортировка меняет порядок представления, но не длины. Поэтому все перестановки одной тройки должны получать одинаковый ответ. Для задач по углам сравнивают z^2 с x^2+y^2 , где z — именно наибольшая сторона, и только после проверки существования.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Для (5,3,4) сортировка даёт 3,4,5; $3+4 > 5$, равных сторон нет: scalene. Для (2,2,4) сумма двух меньших равна 4: no triangle. До проверки равенства пар дело не доходит.

```
def triangle(a, b, c):
    if min(a,b,c) < 1 or max(a,b,c) > 1000:
        return "error"
    x, y, z = sorted([a,b,c])
    if x + y <= z:
        return "no triangle"
    if x == z:
        return "equilateral"
    if x == y or y == z:
        return "isosceles"
    return "scalene"

for t in [(3,3,3), (0,0,0), (1,2,3), (5,3,4), (2,2,3)]:
    print(t, triangle(*t))
```



Диапазон 1...1000	иначе error
$x \leq y \leq z$; $x+y>z$	иначе no triangle
$x=z$	equilateral
$x=y$ или $y=z$; иначе	isosceles; иначе scalene

Проверка диапазона и существования предшествует классификации; x, y, z — стороны по возрастанию.

03. Практика 1. Проверяем материал

Перестановка не меняет фигуру

14–25 минут. Проверьте десять входов таблицы. Запишите отдельно проверку диапазона, сумму и классификацию.

Стороны	Метка
3,3,3	equilateral
2,2,3;3,2,2;2,3,2	isosceles для всех
3,4,5;5,3,4	scalene для обеих
2,2,4;1,2,4	no triangle для обеих
0,2,2;1001,2,2	error для обеих

2 минуты — прогноз;4 — запуски;3 — временно замените $\leq$ на $<$ в проверке суммы, найдите пограничный контрпример;2 — восстановите код. Критерии: десять ответов, устойчивость к перестановке, обнаруженное вырождение и правильный порядок.

04. Практика 2. Перенос способа

Классификация по углам через квадраты

25–33 минуты. Теперь классифицируйте допустимый треугольник по углам: $z^2=x^2+y^2 \rightarrow \text{right}$, меньше $\rightarrow \text{acute}$, больше $\rightarrow \text{obtuse}$. Диапазон и отказ сохраняются. За 2 минуты рассчитайте(3,4,5),(4,4,5),(2,3,4);за 4 реализуйте;за 2 добавьте(1,2,3),(0,4,5). Критерии: сортировка, существование, квадраты, пять тестов.

05. Обсуждение и границы вывода

33–40 минут. Можно ли сразу искать равные стороны, а существование проверять в конце? За 2 минуты подготовьте контрпример,за 3 обсудите также метки equilateral/isosceles,за 2 сформулируйте договорённость о единственном ответе.

06. Индивидуальная проверка

40–45 минут. Для(5,5,6) запишите проверку существования, метку по сторонам, сравнение квадратов и метку по углам. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Ошибка проявляется на (2,2,4): получится isosceles вместо no triangle. Положительные стороны и равная пара не гарантируют существования. Для всех перестановок метка одна; полный правильный исходник выше.

Практика 2: разбор и критерии

right, acute, obtuse, no triangle, error. Сравнения: $25=9+16$; $25<16+16$; $16>4+9$. Квадраты целых здесь считаются точно в Python.

```
def angle_type(a, b, c):
    if min(a,b,c) < 1 or max(a,b,c) > 1000:
        return "error"
    x, y, z = sorted([a,b,c])
    if x + y <= z:
        return "no triangle"
    q = x*x + y*y
    if z*z == q:
        return "right"
    if z*z < q:
        return "acute"
    return "obtuse"

for t in [(3,4,5), (4,4,5), (2,3,4), (1,2,3), (0,4,5)]:
    print(t, angle_type(*t))
```

Обсуждение: возможный вывод

Ранний возврат по равным сторонам ошибочно принимает (0,0,0) или (2,2,4). Сначала проверяются предпосылки. Метки взаимоисключающие по договорённости программы, хотя геометрические классы могут включать друг друга.

Индивидуальная проверка: ответ и критерии

$5+5>6$; isosceles; $36<25+25$; acute. Единицы длины условные, квадраты сравниваются в одинаковых квадратных единицах.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Python 3.11+ и IDLE или редактор; запуск `python lesson.py` (либо `python3 lesson.py`). Внешние библиотеки не нужны. Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 8 класс. Предпосылки: сравнения, and/or, Python-функции и сортировка трёх чисел. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание. Матрица программы предназначена учителю.

Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ



07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Повторение и обобщение: марафон по ветвлениям	Проверить допустимость сторон и классифицировать треугольник по равенству сторон, сохранив правильный порядок проверок.	Исходник, таблица прогнозов и фактических результатов
Составлять и отлаживать программы; использовать условные операторы	Классификация по углам через квадраты	Самостоятельное решение и индивидуальная проверка по четырём критериям

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Python: управление выполнением

<https://docs.python.org/3/tutorial/controlflow.html>

Евклид, I.20: неравенство треугольника

<https://mathcs.clarku.edu/~djoyce/elements/bookI/propI20.html>

OpenStax: связь сторон и углов; источник для учителя

<https://openstax.org/books/algebra-and-trigonometry-2e/pages/10-2-non-right-triangles-law-of-cosines>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 18.10.2026.