



Урок 10. Python: алгоритм Евклида

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 10 из 34 · 45 минут · Модуль «Циклы»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Из двух лент длиной 48 и 18 см нужны одинаковые целые отрезки без остатка. Как найти наибольшую возможную длину без перебора всех чисел?

Результат урока: Вычислить НОД алгоритмом Евклида с остатками; проверить оба порядка аргументов и случаи с нулём.

01. Проблемная ситуация

Ленты 48 и 18 см режут на одинаковые целые отрезки. Проверьте кандидаты 2,3,6,9: какие делят обе длины? Как убедиться, что найденный общий делитель действительно наибольший, а не просто один из подходящих?

02. Понятия и разобранный пример

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

Общий делитель сохраняется

Наибольший общий делитель двух положительных целых — наибольшее положительное число, делящее оба без остатка. Если $a=q*b+r$, общие делители a, b совпадают с общими делителями b, r : из двух слагаемых равенства можно выразить третье.

Алгоритм: пока b не равно нулю, заменяем пару (a, b) на $(b, a \% b)$. Строка $a, b = b, a \% b$ вычисляет обе правые части по старым значениям, затем присваивает новые. Нельзя бездумно заменить её двумя последовательными присваиваниями.

При положительном b остаток меньше b и неотрицателен, поэтому второе число уменьшается до нуля. $\text{НОД}(a, 0) = a$ при $a > 0$, $\text{НОД}(0, b) = b$ при $b > 0$. Для пары $(0, 0)$ в этом уроке НОД не определяем и выводим отдельное сообщение. Все входы — неотрицательные целые.

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

$48 = 2 \cdot 18 + 12$; $18 = 1 \cdot 12 + 6$; $12 = 2 \cdot 6 + 0$. Пары: $(48, 18) \rightarrow (18, 12) \rightarrow (12, 6) \rightarrow (6, 0)$. Ответ 6 см. Последний ненулевой делитель — НОД. Код сначала отделяет $(0, 0)$, чтобы не выдавать 0 за положительный наибольший делитель.

```
a = int(input())
b = int(input())
if a == 0 and b == 0:
    print("не определён")
else:
    while b != 0:
        a, b = b, a % b
    print(a)
```



Python: алгоритм Евклида

Пара 1	$48 = 2 \times 18 + 12$
Пара 2	$18 = 1 \times 12 + 6$
Пара 3	$12 = 2 \times 6 + 0$
Ответ	Последнее ненулевое число: 6

Пары алгоритма Евклида: $(48,18) \rightarrow (18,12) \rightarrow (12,6) \rightarrow (6,0)$; НОД равен 6.

03. Практика 1. Проверяем материал

Строим таблицу остатков

Сначала трассируйте 18,48: первый остаток 18, хотя первое число меньше второго. Это допустимый вход, переставлять вручную не нужно.

Ввод a,b: две строки	Ожидаемый вывод
48; 18	6
18; 48	6
7; 13	1
24; 24	24

Создайте gcd.py по полному примеру. Для каждой пары таблицы запишите последовательность пар и итог. Независимо проверьте, что ответ делит оба исходных числа; для 7,13 перечислите положительные делители 7. Критерии: пары обновляются одновременно, цикл завершается, четыре теста, независимая проверка.

04. Практика 2. Перенос способа

Обрабатываем нулевой аргумент

Новые тесты для той же программы: 0,15; 15,0; 0,0. Предскажите число повторений и ответ. Затем самостоятельно найдите НОД для 84 и 30 и объясните, сколько отрезков длиной НОД получится из каждой ленты. Код не переписывайте ради нового набора данных.

05. Обсуждение и границы вывода

Ученик сначала пишет $a=b$, затем $b=a\%b$. Почему это не то же самое, что одновременное присваивание? Разберите на паре 48,18 и предложите альтернативу с временной переменной.

06. Индивидуальная проверка

Найдите НОД(35,14) с парами и остатками. Назовите ответ программы для 0,9 и объясните, почему $b!=0$ проверяется до вычисления остатка. Критерии: трасса/ответ, нулевой вход, защита от деления на ноль.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Для 18,48: (18,48)→(48,18)→(18,12)→(12,6)→(6,0). У 7 делители 1,7; 7 не делит 13, значит НОД 1. Для равных 24 первый остаток 0, ответ 24. Полный правильный код приведён в примере. По 1 баллу за четыре критерия.

Практика 2: разбор и критерии

0,15: один шаг к (15,0), ответ 15; 15,0: ноль шагов, ответ 15; 0,0: сообщение «не определён», цикл не запускается. 84,30→30,24→24,6→6,0: НОД 6, отрезков 14 и 5. Критерии: три нулевых случая, трасса новой пары, два количества отрезков.

Обсуждение: возможный вывод

После $a=b$ оба числа равны 18; остаток $18\%18=0$, ошибочный ответ 18. Нужен остаток от старых значений: $g=a\%b$; $a=b$; $b=g$. Одновременное присваивание короче, временная переменная нагляднее для трассировки.

Индивидуальная проверка: ответ и критерии

(35,14)→(14,7)→(7,0), НОД 7. Для 0,9 ответ 9. Проверка исключает вычисление $a\%0$, которое недопустимо. 3 балла.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, локальную папку с правом записи, Python/IDLE, распечатку ученической части PDF, бумагу и ручку. Внешние библиотеки не нужны: только стандартный Python 3.11 или новее. Все программы, входные данные, таблицы и ключи находятся в статье и PDF. Детские аккаунты, сеть и регистрация не нужны.

Уровень — начинающий, 7 класс. Предпосылки: //, %, while и условный оператор. Работа в паре: автор прогнозирует и вводит код, проверяющий ведёт журнал; затем роли меняются. На бумаге трассируйте готовый код и входные данные; для графики используйте координатную сетку и схему. Учитель может показать заранее полученный вывод/PNG. Бумажный вариант проверяет понимание, но не выдаётся за реальный запуск учеником. Чтение, ввод и проверка включены в план. Матрица программы — материал учителя.

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
Алгоритм Евклида	Повторяет переход к делителю и остатку	НОД и трассировка
Цикл с условием	Проверяет нулевые аргументы	Нет операции %0

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Официальное руководство Python 3.11: управление выполнением

<https://docs.python.org/3.11/tutorial/controlflow.html>

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: 17.10.2026.