

# **Отчёт по лабораторной работе 4. Вычисление наибольшего общего делителя**

Ильин Никита Евгеньевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>

## Список иллюстраций

4.1	Программная реализация алгоритма Эвклида . . . . .	9
4.2	Результат работы программы . . . . .	10
4.3	Программная реализация алгоритма Эвклида . . . . .	10
4.4	Результат работы программы . . . . .	11
4.5	Программная реализация алгоритма Эвклида . . . . .	11
4.6	Результат работы программы . . . . .	12
4.7	Программная реализация алгоритма Эвклида . . . . .	13
4.8	Результат работы программы . . . . .	14

## **Список таблиц**

# 1 Цель работы

Цель данной работы - научиться реализовывать алгоритмы поиска НОД.

## 2 Задание

1. Реализовать алгоритмы поиска НОД.

### 3 Теоретическое введение

ЛАБОРАТОРНАЯ РАБОТА №4 Вычисление наибольшего общего делителя Пусть числа  $a$  и  $b$  целые и  $b \neq 0$ . Разделить  $a$  на  $b$  с остатком - значит представить  $a$  в виде  $a = qb + r$ , где  $q, r \in \mathbb{Z}$  и  $0 \leq r < |b|$ . Число  $q$  называется неполным частным, число  $r$  - неполным остатком от деления  $a$  на  $b$ . Целое число  $d \neq 0$  называется наибольшим общим делителем целых чисел  $a_1, a_2, \dots, a_k$  (обозначается  $d = \text{НОД}(a_1, a_2, \dots, a_k)$ ), если выполняются следующие условия: 1. каждое из чисел  $a_1, a_2, \dots, a_k$  делится на  $d$ ; 2. если  $d_1 \neq 0$  - другой общий делитель чисел  $a_1, a_2, \dots, a_k$ , то  $d$  делится на  $d_1$ . Например,  $\text{НОД}(12345, 24690) = 12345$ ,  $\text{НОД}(12345, 54321) = 3$ ,  $\text{НОД}(12345, 12541) = 1$ . Ненулевые целые числа  $a$  и  $b$  называются ассоциированными (обозначается  $a \sim b$ ), если  $a$  делится на  $b$  и  $b$  делится на  $a$ . Для любых целых чисел  $a_1, a_2, \dots, a_k$  существует наибольший общий делитель  $d$  и его можно представить в виде линейной комбинации этих чисел:  $d = G_1 a_1 + G_2 a_2 + \dots + G_k a_k$ ,  $G_i \in \mathbb{Z}$  ( $\mathbb{Z}$  - множество целых чисел). Например,  $\text{НОД}$  чисел 91, 105, 154 равен 7. В качестве линейного представления можно взять  $7 = 7 \cdot 91 + (-6) \cdot 105 + 0 \cdot 154$ , либо  $7 = 4 \cdot 91 + 1 \cdot 105 - 3 \cdot 154$ . Целые числа  $a_1, a_2, \dots, a_k$  называются взаимно простыми в совокупности, если  $\text{НОД}(a_1, a_2, \dots, a_k) = 1$ . Целые числа  $a$  и  $b$  называются взаимно простыми, если  $\text{НОД}(a, b) = 1$ . Целые числа  $a_1, a_2, \dots, a_k$  называются попарно взаимно простыми, если  $\text{НОД}(a_i, a_j) = 1$  для всех  $1 \leq i < j \leq k$ .

41. Пока и четное, полагать и «»

42. Пока четное, полагать и «» 4.3. При и  $\geq$  и положить и- и- v. В противном случае положить о- г - и. 5. Положить  $d - gv$ . 3. Расширенный алгоритм Евклида. Вход. Целые числа  $a, b$ ;  $0 < b \leq a$ . 1. Положить  $r - a, r - b, x - 1, y - 0$

- ,  $Y_0 = 0$ ,  $Y_1 = 1$ , ...,  $Y_{i-1}$ . 2. Разделить с остатком  $r_{i-1}$  на  $r_i$ :  $r_{i-1} = a_i r_i + r_{i+1}$ . 3. Если  $r_{i+1} = 0$ , то положить  $d = r_i$ ,  $x = X_i$ ,  $y = Y_i$ . В противном случае Положить  $X_{i+1} = X_i - \Phi_i X_i$ ,  $Y_{i+1} = Y_i - \Phi_i Y_i$ ,  $i = i + 1$  и вернуться на шаг 2.
43. Результат:  $d, x, y$ .
44. Расширенный бинарный алгоритм Евклида. Вход. Целые числа  $a, b$ ;  $0 < b \leq a$ . Выход.  $d = \text{НОД}(a, b)$ .
45. Положить  $d = 1$ . 2. Пока  $a$  и  $b$  нечетные, выполнять  $a \leftarrow a - 4$ ,  $b \leftarrow b + 4$  до получения хотя бы одного нечетного значения  $a$  или  $b$ .
46. Положить  $i = a, v = b, A = 1, B = 0, C = 0, D = 1$ .
47. Пока  $i \neq 0$  выполнять следующие действия: 4.1. Пока  $i$  четное: 4.1.1. Положить  $i = i / 2$ . 4.1.2. Если оба числа  $A$  и  $B$  четные, то положить  $A \leftarrow A / 2, B \leftarrow B / 2$ . В противном случае положить  $A \leftarrow A + b, B \leftarrow B + a$ . 4.2. Пока  $i$  нечетное: 4.2.1. Положить  $0 < i < 2$ .
48.  $17 \cdot 2, B < B - a$

Саратовский государственный университет имени Н.И. Чернышевского

Саратовский государственный университет имени Н.И. Чернышевского

4.2.2. Если оба числа  $S$  и  $D$  четные, то положить  $S \leftarrow S / 2, D \leftarrow D / 2$ . В противном случае положить  $S \leftarrow S - 4$ . 3. При  $i \geq v$  положить  $i = i - 0, A \leftarrow A - C, B \leftarrow B - D$ . В противном случае положить  $0 = v - i, S \leftarrow S - A D, D \leftarrow B$ . 5. Положить  $d = \gcd(x - C, y + D)$ . 6. Результат:  $d, x, y$ .



## 4 Выполнение лабораторной работы

1. Для начала реализуется функция алгоритма Эвклида (рис. 4.1).

```
def euclid(a, b):  
    while a!=0 and b!=0:  
        if (a >= b):  
            a %= b  
        else:  
            b %= a  
    return a or b
```

Рис. 4.1: Программная реализация алгоритма Эвклида

2. Результат работы функции (рис. 4.2).

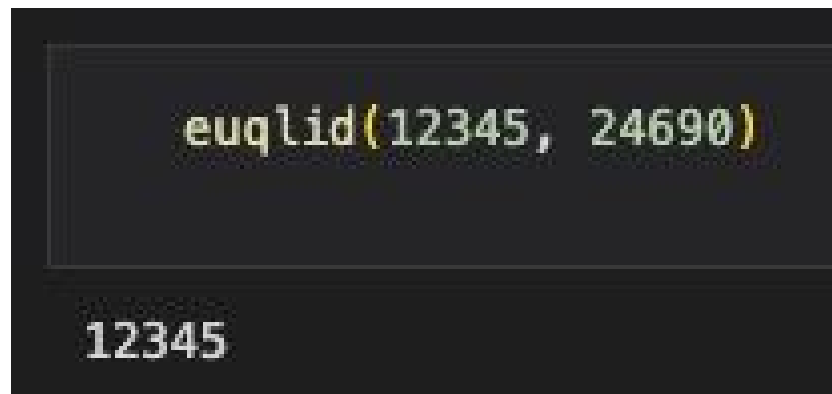


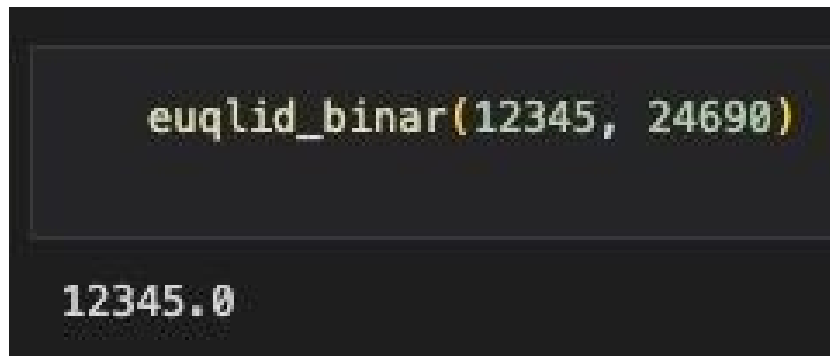
Рис. 4.2: Результат работы программы

3. Для начала реализуется функция бинарного алгоритма Евклида. (рис. 4.3).

```
def euclid_binar(a, b):  
    g = 1  
    while a % 2 == 0 and b % 2 == 0:  
        a /= 2  
        b /= 2  
        g *= 2  
    u, v = a, b  
    while u != 0:  
        if u % 2 == 0:  
            u /= 2  
        if v % 2 == 0:  
            v /= 2  
        if u >= v:  
            u = u - v  
        else:  
            v = v - u  
    d = g * v  
    return d
```

Рис. 4.3: Программная реализация алгоритма Эвклида

4. Результат работы функции (рис. 4.4).

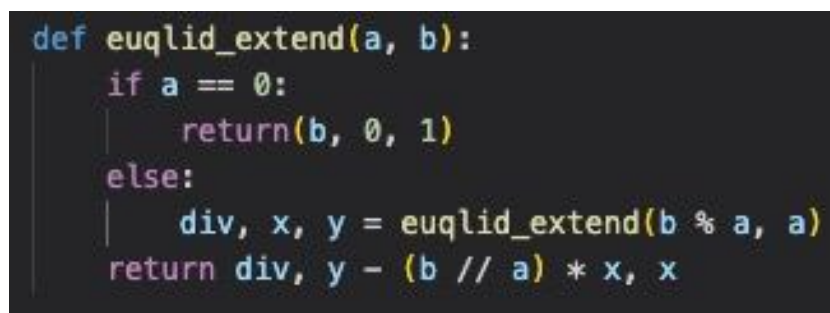


```
euclid_binar(12345, 24690)

12345.0
```

Рис. 4.4: Результат работы программы

5. Для начала реализуется функция алгоритма Эвклида (рис. 4.5).



```
def euclid_extend(a, b):
    if a == 0:
        return(b, 0, 1)
    else:
        div, x, y = euclid_extend(b % a, a)
        return div, y - (b // a) * x, x
```

Рис. 4.5: Программная реализация алгоритма Эвклида

6. Результат работы функции (рис. 4.6).

```
euclid_extend(12345, 24690)  
  
(12345, 1, 0)
```

Рис. 4.6: Результат работы программы

7. Для начала реализуется функция алгоритма Эвклида (рис. 4.7).

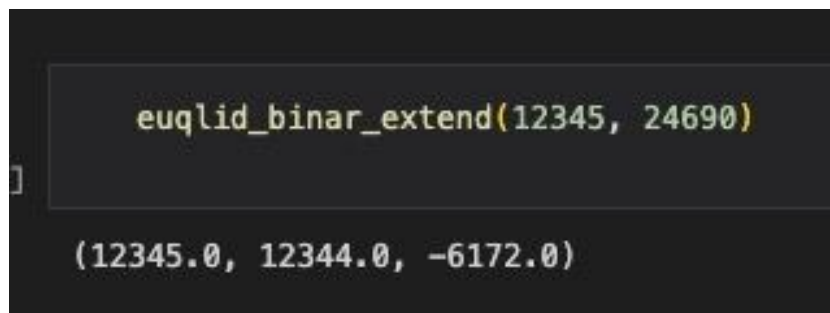
```

def euclid_binar_extend(a, b):
    g = 1
    while a % 2 == 0 and b % 2 == 0:
        a /= 2
        b /= 2
        g *= 2
    u, v = a, b
    A, B, C, D = 1, 0, 0, 1
    while u != 0:
        if u % 2 == 0:
            u /= 2
            if A % 2 == 0 and B % 2 == 0:
                A /= 2
                B /= 2
            else:
                A = (A + b) / 2
                B = (B - a) / 2
        if v % 2 == 0:
            v /= 2
            if C % 2 == 0 and D % 2 == 0:
                C /= 2
                D /= 2
            else:
                C = (C + b) / 2
                D = (D - a) / 2
        if u >= v:
            u -= v
            C -= A
            D -= B
        else:
            v = v - u
            C -= A
            D -= B
    d = g * v
    x = C
    y = D
    return d, x, y

```

Рис. 4.7: Программная реализация алгоритма Эвклида

8. Результат работы функции (рис. 4.8).



```
euclid_binar_extend(12345, 24690)  
  
(12345.0, 12344.0, -6172.0)
```

A screenshot of a terminal window with a dark background. The first line shows the function call `euclid_binar_extend(12345, 24690)` in a light green monospace font. The second line shows the output tuple `(12345.0, 12344.0, -6172.0)` in a light gray monospace font. A small white cursor is visible on the left side of the terminal.

Рис. 4.8: Результат работы программы

## 5 Выводы

В ходе работы были реализованы алгоритмы вычисления НОД.