

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

---

Ильин Никита Евгеньевич

## Цель выполнения лабораторной работы

---

Цель данной работы - научиться реализовывать алгоритмы поиска НОД.

1. Для начала реализуется функция алгоритма Эвклида (рис. (fig:001?)).

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

Figure 1: Программная реализация алгоритма Эвклида

2. Результат работы функции (рис. (fig:002?)).

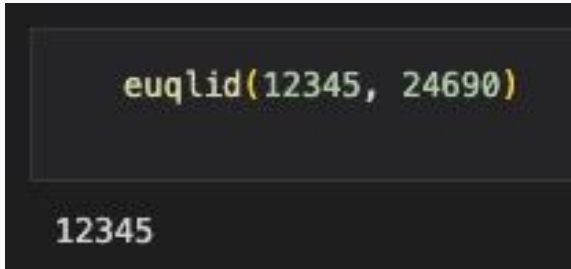


Figure 2: Результат работы программы

## Выполнение лабораторной работы

3. Для начала реализуется функция бинарного алгоритма Евклида.  
(рис. (fig:003?)).

```
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:
```

4. Результат работы функции (рис. (fig:004?)).



```
euqlid_binar(12345, 24690)  
  
12345.0
```

Figure 4: Результат работы программы

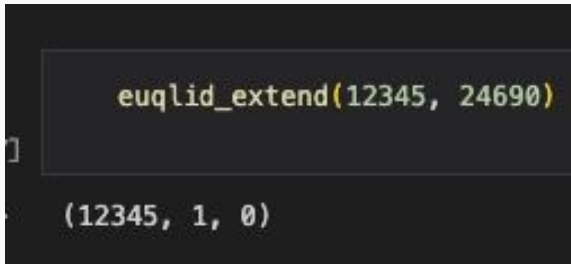
5. Для начала реализуется функция алгоритма Эвклида (рис. (fig:005?)).

```
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
```

Figure 5: Программная реализация алгоритма Эвклида



6. Результат работы функции (рис. (fig:006?)).



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

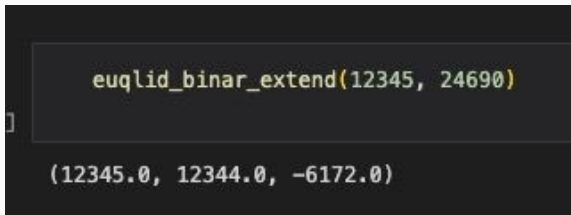
Figure 6: Результат работы программы

# Выполнение лабораторной работы

7. Для начала реализуется функция алгоритма Эвклида (рис. (fig:007?)).

```
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 -= u
```

8. Результат работы функции (рис. (fig:008?)).

A screenshot of a code editor with a dark background. The first line of code is `euclid_binar_extend(12345, 24690)` in a yellow-green font. Below it, the output is displayed in white font as `(12345.0, 12344.0, -6172.0)`.

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

Figure 8: Результат работы программы

## Выводы

---

В ходе работы были реализованы алгоритмы вычисления НОД.