

Математические основы защиты информации и информационной безопасности

Колчева Юлия Вячеславовна

23 Октября 2024

РУДН, Москва, Россия

Лабораторная работа 4

```
In [4]: 1 number1 = 15
        2 number2 = 61
        3
        4 num1 = number1
        5 num2 = number2
        6 while num1 != 0 && num2 != 0
        7     if num1 >= num2
        8         num1 = num1 % num2
        9     else
       10         num2 = num2 % num1
       11     end
       12 end
       13
       14 nod1 = num1 + num2
       15 println(nod1)
```

1

Рис. 1: Реализация программы

Бинарный алгоритм Евклида

```
In [5]: 1 num1 = number1
        2 num2 = number2
        3 shift = 0
        4
        5 while (num1 | num2) & 1 == 0
        6     shift += 1
        7     num1 >>= 1
        8     num2 >>= 1
        9 end
       10
       11 while num1 & 1 == 0
       12     num1 >>= 1
       13 end
       14
       15 while num2 != 0
       16     while num2 & 1 == 0
       17         num2 >>= 1
       18     end
       19     if num1 > num2
       20         num1, num2 = num2, num1
       21     end
       22     num2 -= num1
       23 end
       24
       25 println(num1 << shift)
```

Расширенный алгоритм Евклида

```
In [21]: 1 function ext_evk(a, b)
2         if b == 0
3             return a, 1, 0
4         else
5             nod, x1, y1 = ext_evk_2(b, a % b)
6             x = y1
7             y = x1 - div(a, b) * y1
8             return nod, x, y
9         end
10      end
11      x, y, g = ext_evk(number1, number2)
12      println(x, '*', number1, '+', '(' , y, ')', '*', number2, '=', g)

1*15+(-4)*61=1
```

Рис. 3: Реализация программы

Расширенный бинарный алгоритм Евклида

```
def ext_gcd(a, b):  
    if a == 0: return 1, 1, b  
    if b == 0: return 1, 1, a  
    if not a & 1 | b & 1:  
        x, y, g = ext_gcd(a >> 1, b >> 1)  
        return x, y, g << 1  
    elif not a & 1:  
        x, y, g = ext_gcd(a >> 1, b)  
        return (x - b >> 1, y + (a >> 1), g) if x & 1 else (x >> 1, y, g)  
    elif not b & 1:  
        x, y, g = ext_gcd(a, b >> 1)  
        return (x + (b >> 1), y - a >> 1, g) if y & 1 else (x, y >> 1, g)  
    elif b > a:  
        x, y, g = ext_gcd(a, b - a)  
        return x - y, y, g  
    else:  
        x, y, g = ext_gcd(a - b, b)  
        return x, y - x, g
```

Рис. 4: Реализация программы

```
1  
1  
-4 * 15 + ( 1 ) * 61 = 1  
57 * 15 + ( -14 ) * 61 = 1
```

Рис. 5: Вывод программ

- Познакомилась с алгоритмами вычисления наибольшего общего делителя.
- Применила алгоритмы на практике.

Спасибо за внимание!