Front matter

Front matter

lang: ru-RU title: "Математические основы защиты информации и информационной безопасности" subtitle: "Отчёт по лабораторной работе №4:Вычисление наибольшего общего author: "Бекбузарова Роза Алисхановна" institute:

- Российский университет дружбы народов, Москва, Россия

i18n babel

babel-lang: russian babel-otherlangs: english

Formatting pdf

toc: false toc-title: Содержание slide_level: 2 aspectratio: 169 section-titles: true theme: metropolis header-includes:

- \metroset{progressbar=frametitle,sectionpage=progressbar,numbering=fraction}
- '\makeatletter'
- '\beamer@ignorenonframefalse'
- '\makeatother'

Цель работы

Основной целью работы является реализовать разными алгоритмами вычисление наибольшего общего делителя.

Выполнение лабораторной работы

Алгоритм Евклида

[22]:
```
# Классический алгоритм Евклида
function gcd_euclid(a::Int, b::Int)
    r0, r1 = a, b
    while r1 != 0
        r0, r1 = r1, r0 % r1
    end
    return r0
end
println("gcd_euclid(48, 18) = ", gcd_euclid(48, 18))
```

```
gcd_euclid(48, 18) = 6
```

Бинаврный алгоритм Евклида

[24]:
```
function gcd_binary(a::Int, b::Int)
    if a == 0
        return b
    elseif b == 0
        return a
    end
    shift = 0
    # Убираем общие факторы 2
    while (a & 1 == 0) && (b & 1 == 0)
        a >>= 1
        b >>= 1
        shift += 1
    end
    # Делим a на 2 пока оно чётное
    while (a & 1) == 0
        a >>= 1
    end
    while b != 0
        # Делим b на 2 пока оно чётное
        while (b & 1) == 0
            b >>= 1
        end
        # Обеспечиваем a <= b
        if a > b
            a, b = b, a
        end
        b -= a
    end
    return a << shift
end
println("gcd_binary(48, 18) = ", gcd_binary(108, 18))
```

```
gcd_binary(48, 18) = 18
```

Расширенный алгоритм Евклида

[26]:
```
# 1. Расширенный алгоритм Евклида
function extended_euclid(a::Int, b::Int)
    r0, r1 = a, b
    x0, x1 = 1, 0
    y0, y1 = 0, 1

    while r1 != 0
        q = r0 ÷ r1
        r0, r1 = r1, r0 - q * r1
        x0, x1 = x1, x0 - q * x1
        y0, y1 = y1, y0 - q * y1
    end

    return r0, x0, y0  # gcd, x, y
end
a, b = 12, 244
println("Расширенный алгоритм Евклида:")
g, x, y = extended_euclid(a, b)
println("gcd($a, $b) = $g, x = $x, y = $y  =>  $a*$x + $b*$y = $(a*x + b*y)")
```

```
Расширенный алгоритм Евклида:
gcd(12, 244) = 4, x = -20, y = 1  =>  12*-20 + 244*1 = 4
```

Расширенный бинарный алгоритм Евклида

```
                A = (A + b) >> 1
                B = (B - a) >> 1
            end
        end
        # Делим v на 2 пока оно четное
        while (v & 1) == 0
            v >>= 1
            if (C & 1 == 0) && (D & 1 == 0)
                C >>= 1
                D >>= 1
            else
                C = (C + b) >> 1
                D = (D - a) >> 1
            end
        end
        if u >= v
            u -= v
            A -= C
            B -= D
        else
            v -= u
            C -= A
            D -= B
        end
    end
    gcd = v * g
    x = C
    y = D
    return gcd, x, y
end
a, b = 9, 108
println("\nРасширенный бинарный алгоритм Евклида:")
g2, x2, y2 = extended_binary_euclid(a, b)
println("gcd($a, $b) = $g2, x = $x2, y = $y2  =>  $a*$x2 + $b*$y2 = $(a*x2 + b*y2)")
```

```
Расширенный бинарный алгоритм Евклида:
gcd(9, 108) = 9, x = 13, y = -1  =>  9*13 + 108*-1 = 9
```

Вывод

В ходе выполнения лабораторной работы было реализовано разными алгоритмами вычисление наибольшего общего делителя.