# Лабораторная работа №4

Тема: Вычисление наибольшего общего делителя (НОД)

Выполнила: Исламова Сания Маратовна

Группа: НПИмд-01-24

Студ.билет: 1132249576

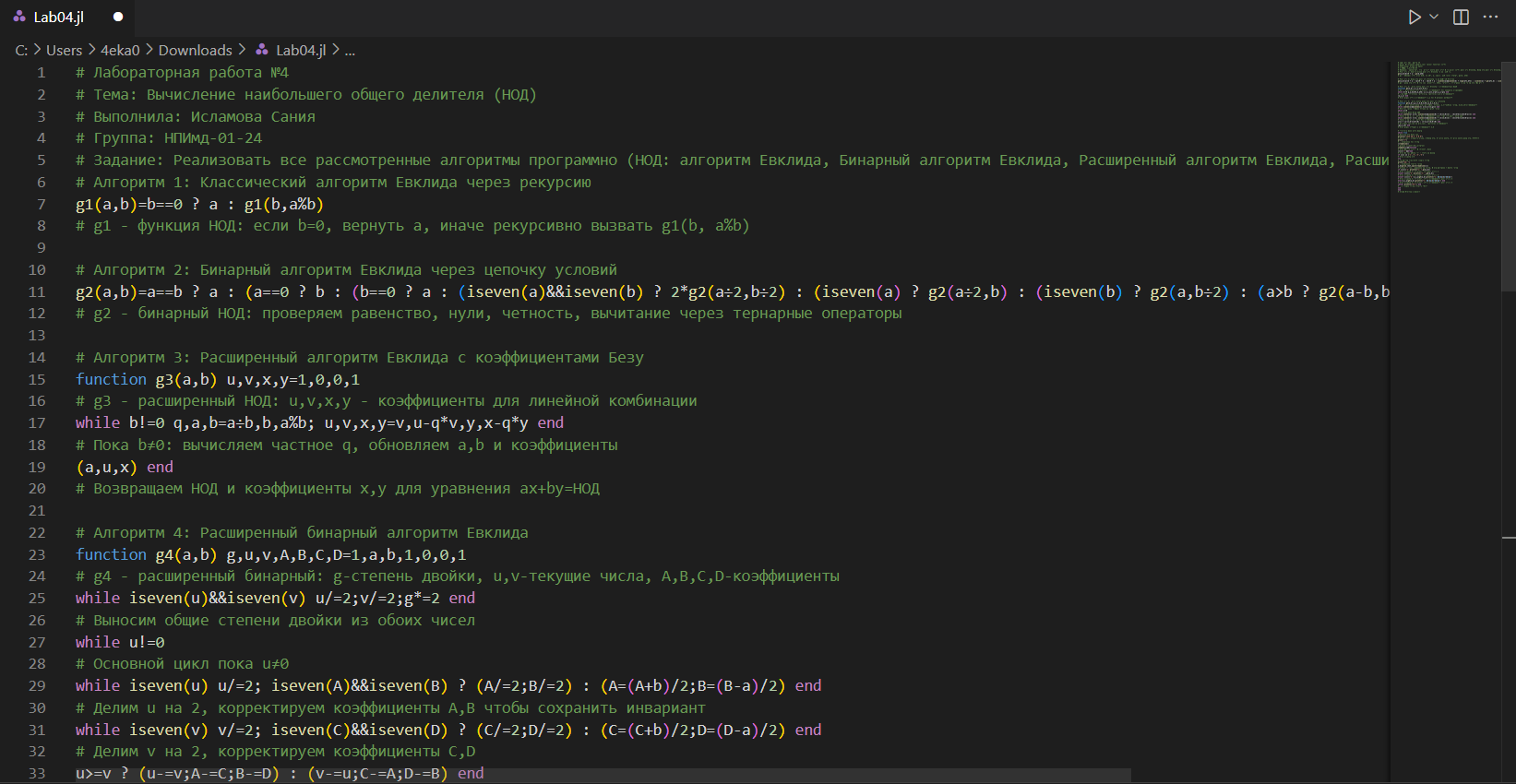
### Задача лабораторной работы:

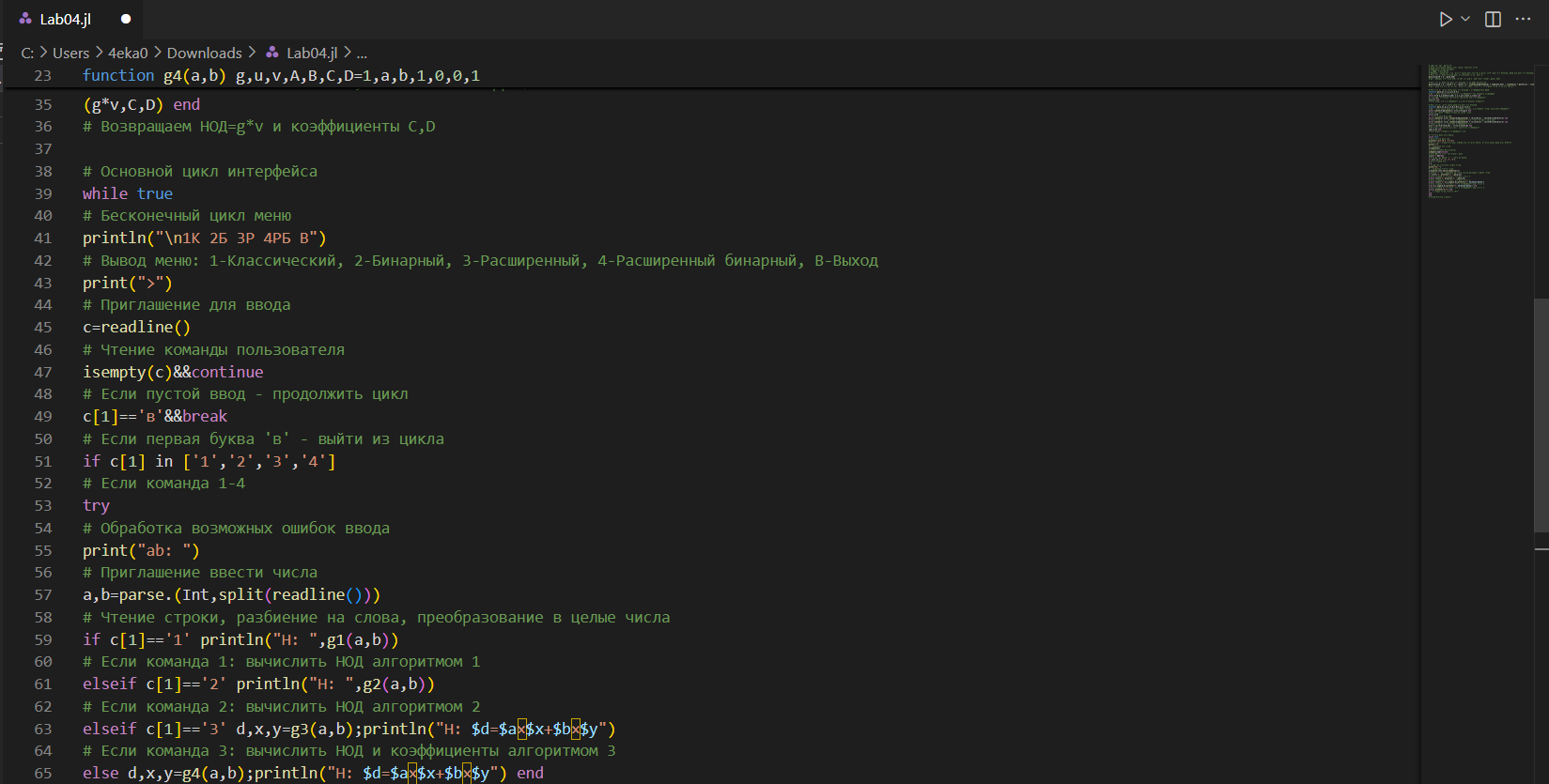
Реализовать все рассмотренные алгоритмы программно (НОД: алгоритм Евклида, Бинарный алгоритм Евклида, Расширенный алгоритм Евклида, Расширенный бинарный алгоритм Евклида)

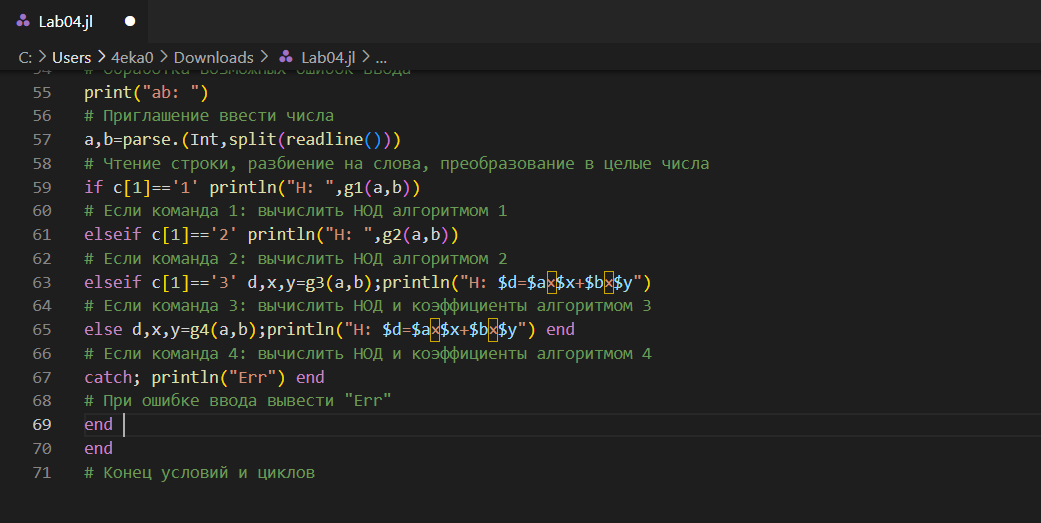
### Описание хода выполнения лабораторной работы:

# Алгоритм 1: Классический алгоритм Евклида через рекурсию  
g1(a,b)=b==0 ? a : g1(b,a%b)  
# g1 - функция НОД: если b=0, вернуть a, иначе рекурсивно вызвать g1(b, a%b)  
  
# Алгоритм 2: Бинарный алгоритм Евклида через цепочку условий  
g2(a,b)=a==b ? a : (a==0 ? b : (b==0 ? a : (iseven(a)&&iseven(b) ? 2\*g2(a÷2,b÷2) : (iseven(a) ? g2(a÷2,b) : (iseven(b) ? g2(a,b÷2) : (a>b ? g2(a-b,b) : g2(a,b-a)))))))  
# g2 - бинарный НОД: проверяем равенство, нули, четность, вычитание через тернарные операторы  
  
# Алгоритм 3: Расширенный алгоритм Евклида с коэффициентами Безу  
function g3(a,b) u,v,x,y=1,0,0,1  
# g3 - расширенный НОД: u,v,x,y - коэффициенты для линейной комбинации  
while b!=0 q,a,b=a÷b,b,a%b; u,v,x,y=v,u-q\*v,y,x-q\*y end  
# Пока b≠0: вычисляем частное q, обновляем a,b и коэффициенты  
(a,u,x) end  
# Возвращаем НОД и коэффициенты x,y для уравнения ax+by=НОД  
  
# Алгоритм 4: Расширенный бинарный алгоритм Евклида  
function g4(a,b) g,u,v,A,B,C,D=1,a,b,1,0,0,1  
# g4 - расширенный бинарный: g-степень двойки, u,v-текущие числа, A,B,C,D-коэффициенты  
while iseven(u)&&iseven(v) u/=2;v/=2;g\*=2 end  
# Выносим общие степени двойки из обоих чисел  
while u!=0  
# Основной цикл пока u≠0  
while iseven(u) u/=2; iseven(A)&&iseven(B) ? (A/=2;B/=2) : (A=(A+b)/2;B=(B-a)/2) end  
# Делим u на 2, корректируем коэффициенты A,B чтобы сохранить инвариант  
while iseven(v) v/=2; iseven(C)&&iseven(D) ? (C/=2;D/=2) : (C=(C+b)/2;D=(D-a)/2) end  
# Делим v на 2, корректируем коэффициенты C,D  
u>=v ? (u-=v;A-=C;B-=D) : (v-=u;C-=A;D-=B) end  
# Вычитаем меньшее из большего, обновляем коэффициенты  
(g\*v,C,D) end  
# Возвращаем НОД=g\*v и коэффициенты C,D  
  
# Основной цикл интерфейса  
while true   
# Бесконечный цикл меню  
println("\n1К 2Б 3Р 4РБ В")  
# Вывод меню: 1-Классический, 2-Бинарный, 3-Расширенный, 4-Расширенный бинарный, В-Выход  
print(">")   
# Приглашение для ввода  
c=readline()  
# Чтение команды пользователя  
isempty(c)&&continue  
# Если пустой ввод - продолжить цикл  
c[1]=='в'&&break  
# Если первая буква 'в' - выйти из цикла  
if c[1] in ['1','2','3','4']   
# Если команда 1-4  
try   
# Обработка возможных ошибок ввода  
print("ab: ")   
# Приглашение ввести числа  
a,b=parse.(Int,split(readline()))  
# Чтение строки, разбиение на слова, преобразование в целые числа  
if c[1]=='1' println("Н: ",g1(a,b))  
# Если команда 1: вычислить НОД алгоритмом 1  
elseif c[1]=='2' println("Н: ",g2(a,b))  
# Если команда 2: вычислить НОД алгоритмом 2  
elseif c[1]=='3' d,x,y=g3(a,b);println("Н: $d=$a×$x+$b×$y")  
# Если команда 3: вычислить НОД и коэффициенты алгоритмом 3  
else d,x,y=g4(a,b);println("Н: $d=$a×$x+$b×$y") end   
# Если команда 4: вычислить НОД и коэффициенты алгоритмом 4  
catch; println("Err") end   
# При ошибке ввода вывести "Err"  
end   
end  
# Конец условий и циклов

### Код реализации вычисление наибольшего общего делителя (НОД) алгоритмами Евклида:







### Результат работы вычисление наибольшего общего делителя (НОД) алгоритмами Евклида:

