

## Лабораторная работа №4

Математические основы защиты информации и информационной безопасности

---

Мануэл Марсия Педру

2026

Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

::::::::::: {.columns align=center} ::: {.column width="70%"}

- Мануэл Марсия Педру
- Студент группы НФИМд-02-25
- Студ. билет 1032255503
- Российский университет дружбы народов имени Патриса Лумумбы

## 2.1 Цель работы

---

## 2.1 Цель работы

---

Получение практических навыков работы в консоли с атрибутами файлов для групп пользователей.

## 2 Выполнение лабораторной работы

---

## 2.1 Реализация алгоритма Евклида

---

## 2.1 Реализация алгоритма Евклида

---

Алгоритм Евклида — эффективный алгоритм для нахождения наибольшего общего делителя двух целых чисел (или общей меры двух отрезков). Алгоритм назван в честь греческого математика Евклида (III век до н. э.), который впервые описал его в VII и X книгах «Начал»

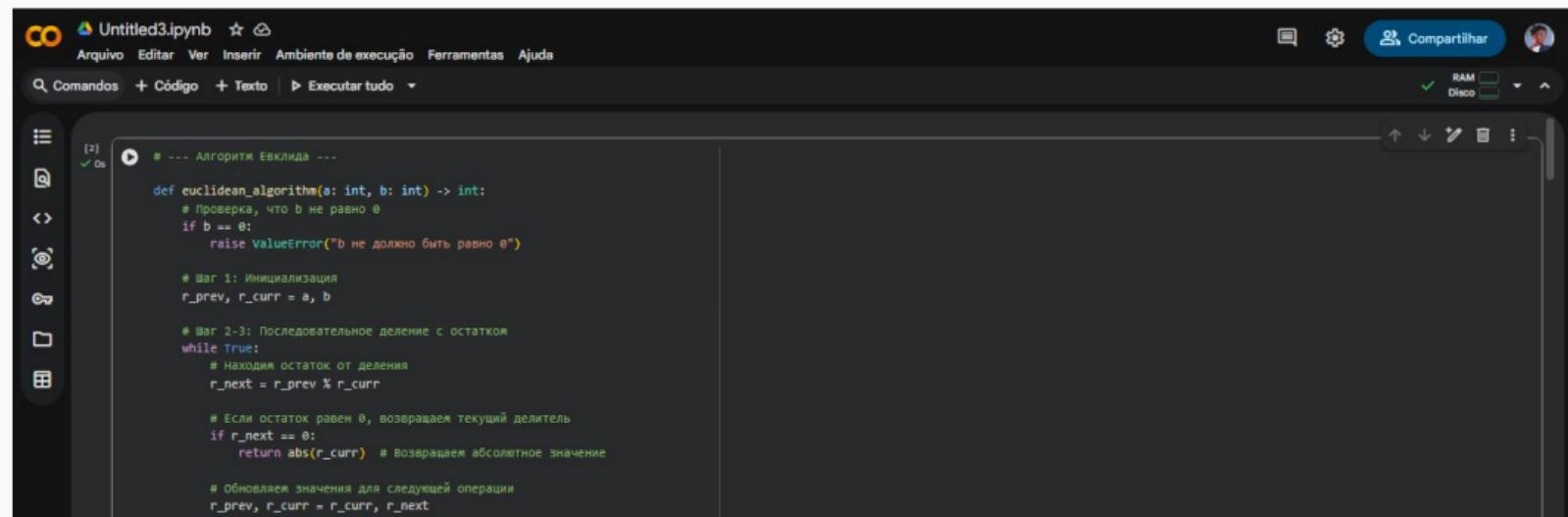
## 2.3 Реализация расширенного алгоритма Евклида

---

## 2.3 Реализация расширенного алгоритма Евклида

Расширенный алгоритм Евклида — модификация алгоритма Евклида, вычисляющая, кроме наибольшего общего делителя (НОД) целых чисел  $a$  и  $b$ , ещё и коэффициенты соотношения Безу, то есть такие целые  $x$  и  $y$ , что  $xa + yb = \text{НОД}(a, b)$ .

Выполним реализацию этого алгоритма на языке Python (рис. (fig:001?)):



The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar includes the file name "Untitled3.ipynb", navigation menus like Arquivo, Editar, Ver, Inserir, Ambientes de execução, Ferramentas, Ajuda, and a toolbar with various icons. On the left, there's a sidebar with notebook, cell, and file/folder icons. The main area contains a code cell with the following Python code:

```
[2] # --- Алгоритм Евклида ---
def euclidean_algorithm(a: int, b: int) -> int:
    # Проверка, что b не равно 0
    if b == 0:
        raise ValueError("b не должно быть равно 0")

    # Шаг 1: Инициализация
    r_prev, r_curr = a, b

    # Шаг 2-3: Последовательное деление с остатком
    while True:
        # Находим остаток от деления
        r_next = r_prev % r_curr

        # Если остаток равен 0, возвращаем текущий делитель
        if r_next == 0:
            return abs(r_curr) # Возвращаем абсолютное значение

        # Обновляем значения для следующей операции
        r_prev, r_curr = r_curr, r_next
```