# Author

Старикова Евгения Дмитриевна

email: 1032259343@pfur.ru

Российский университет дружбы народов

Российская Федерация

Москва

**"Лабораторная работа №3" "Шифрование гаммированием"**

**Цель работы**

Цель работы -- изучить и реализовать шифрование гаммированием.

# Задание

С помощью языка программирования Julia реализовать:

шифрование гаммированием.

# Теоретическое введение

Julia — высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений[@julialang]. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

Для выполнения заданий была использована официальная документация Julia[@juliadoc].

# Выполнение лабораторной работы

Шифрование гаммированием — это симметричный метод шифрования, при котором к открытым данным (тексту) применяется операция наложения (обычно XOR) с последовательностью случайных чисел, называемой гаммой. Эта гамма должна быть не короче сообщения и обеспечивает обратимость операции, позволяя расшифровать данные при наличии той же гаммы. Такой метод обеспечивает высокую стойкость при условии использования случайной и одноразовой гаммы.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | function gamma\_cypher(text, gamma; mod=33) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | russian\_alphabet = collect("абвгдежзийклмнопрстуфхцчшщьыъэюя") | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | # Фильтруем текст, оставляя только буквы и приводим к нижнему регистру | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | filtered\_text | | | | | = [c for c in lowercase(text) if c in russian\_alphabet] | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | # Преобразуем | | | | | | | | | гамму | | | | | | | | в | | числовые | | | | | | | значения | | | | | | | | gamma\_nums = [findfirst(==(c), russian\_alphabet) for c in gamma] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | # Создаем | | | | | | повторяющуюся | | | | | | | | | | | | | | гамму | | | | | нужной | | | | | длины | | | | | repeated\_gamma = repeat(gamma\_nums, ceil(Int, length(filtered\_text) / length(gamma\_nums))) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | | cyphered\_text = "" | | | | | | | | | | | | |  | | for (a, b) in zip(filtered\_text, repeated\_gamma) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | # Находим | | | | | | | | индекс | | | | | | | | буквы | | | | | в | алфавите | | | | | | | | char\_index = findfirst(==(a), russian\_alphabet) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | # Вычисляем | | | | | | | | | | | новый | | | | | | | индекс | | | | | | new\_index = (char\_index + b) % mod | | | | | | | | | | | | | | | | | | | | | | | | | | | | if new\_index == 0 | | | | | | | | | | | | | | new\_index = mod | | | | | | | | | | | | | | | | end | | | | | | # Добавляем | | | | | | | | | | соответствующую | | | | | | | | | | | | | | букву | | | | | cyphered\_text \*= russian\_alphabet[new\_index] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | end | | | | |  | | return cyphered\_text | | | | | | | | | | | | | | | end | | | |

Также реализуем простую программу для проверки работы шифра:

# Выводы

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | function main() | | | text = "приказ | | | " | | | gamma = "гамма | | | | | | " | | | cyphered\_text = gamma\_cypher(text, gamma) | | | | | | | | | | | | | | | println("Исходный | | | | | | | текст | | : ", text) | | | | | println("Гамма | | | | : ", gamma) | | | | | | | | println("Зашифрованный | | | | | | | | | | текст | | : ", cyphered\_text) | | | | end |        |  | | --- | | Результат: | | Исходный текст: приказ | | | | Гамма: гамма | | | Зашифрованный текст: усхчбл | | | | |

С помощью языка программирования Julia были реализованы: шифрование гаммированием.

# Список литературы{.unnumbered}

::: {#refs} :::