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

Лабораторная работа №2

Реализация шифров перестановки

АВТОР
Кюнкриков Д.С. -

ПРИНАДЛЕЖНОСТЬ
Российский университет дружбы народов, Москва, Россия
-

Информация

Докладчик

- Кюнкриков Даниил Саналович
- студент уч. группы НПИмд-01-24
- Российский университет дружбы народов
- 1132249574@pfur.ru
- https://github.com/DanzanK/2025-2026_math-sec/tree/main

Объект и предмет исследования

- Объект исследования - Шифры перестановки
- Предмет исследования - Шифры: маршрутное шифрование, шифрование с помощью решеток, таблица Виженера

Цели и задачи

- Цель работы: Реализовать шифры перестановки: маршрутное шифрование, шифрование с помощью решеток, таблица Виженера

Маршрутное шифрование

```
function marshr_main()
    # Бесконечный цикл для работы программы до команды выхода

    while true
        # Выводим меню с доступными командами
        println("ш - шифрование, р - расшифровка, в - выход")
        #menu = lowercase(strip(readline()))
        cmd = lowercase(strip(readline()))
        cmd == "в" && (println("выход");
        break)
```

```
        cmd in ["ш","р"] || (println("Ошибка команды"); continue)
        # Запрашиваем текст для шифрования/расшифрования и пароля шифра
        print("Введите сообщение:")
        text = readline()
        print("Введите пароль :")
        password = readline()

        # подготовка текста
        # удаление пробелов и приводим к верхнему регистру
        clean_text = replace(uppercase(text), " " => "")
        # преобразуем пароль в массив символов (для избежания проблем с индексацией русских си
        pass_chars = collect(uppercase(password))
        # n - количество столбцов (равно длине пароля)
        # m - количество строк
        n, m = length(pass_chars), ceil(Int, length(clean_text) / length(pass_chars))

        padded = clean_text * "А"^(m*n - length(clean_text))
        table = reshape(collect(padded), (m, n))
        # создание таблицы
        column_pairs = [(pass_chars[i], i) for i in 1:length(pass_chars)]
        # Сортируем пары по символам пароля (алфавитный порядок)
        sort!(column_pairs, by = x -> x[1])
        sorted_cols = [idx for (char, idx) in column_pairs]
        result = join([table[i,j] for j in sorted_cols for i in 1:m])
        # вывод результата
        println("Result: $result")
    end
end

marshr_main()
```

Шифрование с помощью решеток

```
function cellbased_main()
    while true
        println("ш - шифрование, р - расшифровка, в - выход")

        cmd = lowercase(strip(readline()))
        cmd == "в" && (println("выход");
        break)

        cmd in ["ш","р"] || (println("Ошибка команды"); continue)
        print("Введите сообщение:")
        # Запрашиваем текст для шифрования/расшифрования и пароля шифра

        text = readline()
        # должен содержать 4 символа для решетки 2x2
        print("Введите пароль (4 символа):")
        password = readline()
```

```
        clean_chars = collect(replace(uppercase(text), " " => ""))
        pass_chars = collect(uppercase(password))

        k = 2 # размер решетки
        # Размер большой решетки (2k × 2k = 4x4)
        size_2k = 2k
        # Создаем булеву маску (false - закрыто, true - прорезь)
        grille = falses(size_2k, size_2k)
        # Заполняем маску прорезями в 4 угловых квадратах 2x2
        for i in 1:k, j in 1:k
            grille[i,j] = grille[i,k+j] = grille[k+i,j]  = grille[k+i,k+j] = true
        end

        total = size_2k^2

        length(clean_chars) < total && append!(clean_chars, fill('A', total - length(clean_cha
        table, idx, mask = fill(' ', size_2k, size_2k), 1, copy(grille)

        for _ in 1:4
            for i in 1:size_2k, j in 1:size_2k
                mask[i,j] && idx <= length(clean_chars) && (table[i,j] = clean_chars[idx]; idx
            end
            mask = reverse(mask,dims=1)
        end

        sorted_cols = sort(1:length(pass_chars), by = i ->pass_chars[i])
        result = join([table[i,j] for j in sorted_cols for i in 1:size_2k])

        println("Result: $result")
    end
end

cellbased_main()
```

Таблица Виженера

```
function vigenere()
    alphabet = collect("АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ")
    n = length(alphabet)

    while true
        println("ш - шифрование, р - расшифровка, в - выход")

        cmd = lowercase(strip(readline()))
        cmd == "в" && (println("выход"); break)

        cmd in ["ш","р"] || (println("Ошибка команды"); continue)
        print("Введите сообщение:")
        text = readline()
        print("Введите пароль :")
        password = readline()
```

```
        clean_chars = collect(replace(uppercase(text), " " => ""))
        pass_chars = collect(replace(uppercase(password), " " => ""))

        # Создаем пустой массив символов для ключа
        key_chars = Char[]
        # Для каждого символа текста определяем соответствующий символ ключа
        for i in 1:length(clean_chars)
            push!(key_chars, pass_chars[(i-1) % length(pass_chars) + 1])
        end

        result_chars = Char[]
        # Обрабатываем каждый символ текста
        for i in 1:length(clean_chars)
            text_char = clean_chars[i]
            key_char = key_chars[i]
            # Находим позиции символов в алфавите
            text_idx = findfirst(==(text_char), alphabet)
            key_idx = findfirst(==(key_char), alphabet)

            if text_idx !== nothing && key_idx !== nothing
                if cmd == "ш"
                    new_idx = (text_idx + key_idx - 1) % n
                    new_idx == 0 && (new_idx = n)
                else
                    new_idx = (text_idx + key_idx) % n
                    new_idx == 0 && (new_idx += n)
                end
                # Добавляем преобразованный символ к результату
                push!(result_chars, alphabet[new_idx])
            else
                push!(result_chars, text_char)
            end
        end
        result = String(result_chars)

        println("Result: $result")
    end
end

vigenere()
```

Вывод

Реализованы шифры перестановки: маршрутное шифрование, шифрование с помощью решеток, таблица Виженера