Лабораторная работа №5

Тема: Вероятностные алгоритмы проверки чисел на простоту

Выполнила: Исламова Сания Маратовна

Группа: НПИмд-01-24

Студ.билет: 1132249576

Задача лабораторной работы:

Реализовать все рассмотренные алгоритмы программно: тест Ферма, тест Соловэя-Штрассена, тест Миллера-Рабина, тест символа Якоби

Описание хода выполнения лабораторной работы:

```Julia
using Random  # Подключаем модуль для работы со случайными числами (нужен для генерации случайных оснований a)

# Функция вычисления символа Якоби (a/n) - используется в тесте Соловэя-Штрассена
function jacobi_symbol(a, n)
    a == 0 && return 0  # Если a = 0, символ Якоби равен 0 (числа не взаимно просты)
    a == 1 && return 1  # Если a = 1, символ Якоби всегда равен 1

    # Шаг 1: Выносим все степени 2 из числа a (представляем a = 2^e * a1, где a1 - нечетное)
    e = 0  # Счетчик степени двойки
    a1 = a  # Копируем a для обработки
    while iseven(a1)  # Пока число четное
        e += 1  # Увеличиваем счетчик степени
        a1 ÷= 2  # Делим на 2 (целочисленное деление)
    end

    # Шаг 2: Вычисляем множитель для степени 2 по формуле (2/n) = (-1)^((n²-1)/8)
    s = 1  # Начальное значение множителя
    if e % 2 == 1  # Если степень двойки нечетная
        if n % 8 == 1 || n % 8 == 7  # Проверяем остаток n по модулю 8
            s = 1   # Если n ≡ ±1 (mod 8), то (2/n) = 1
```

```
        elseif n % 8 == 3 || n % 8 == 5  # Если n ≡ ±3 (mod 8)
            s = -1  # То (2/n) = -1
        end
    end

    # Шаг 3: Применяем квадратичный закон взаимности: если оба числа ≡ 3 (mod 4), меняем знак
    if n % 4 == 3 && a1 % 4 == 3  # Если n ≡ 3 (mod 4) и a1 ≡ 3 (mod 4)
        s = -s  # Меняем знак на противоположный
    end

    # Шаг 4: Рекурсивно вызываем функцию с новыми параметрами (меняем местами a1 и n mod a1)
    n1 = n % a1  # Вычисляем n по модулю a1
    if a1 == 1  # Базовый случай рекурсии: если a1 = 1
        return s  # Возвращаем накопленный множитель
    else  # Иначе продолжаем рекурсию
        return s * jacobi_symbol(n1, a1)  # Рекурсивный вычет с переставленными аргументами
    end
end

# Тест Ферма - простейший вероятностный тест на простоту
function fermat_test()
    println("\n=== ТЕСТ ФЕРМА ===")  # Заголовок теста
    println("Введите число для проверки:")  # Запрос числа n
    n = parse(Int, readline())  # Чтение и преобразование ввода в целое число
    println("Введите количество тестов:")  # Запрос количества проверок k
    k = parse(Int, readline())  # Чтение количества тестов

    # Выполняем k независимых тестов
    for i in 1:k
        a = rand(2:(n-2))  # Генерируем случайное основание a в диапазоне [2, n-2]
        if powermod(a, n-1, n) != 1  # Проверяем условие малой теоремы Ферма: a^(n-1) ≡ 1 (mod n)
            println("Число $n - СОСТАВНОЕ (тест $i с основанием $a)")  # Если условие нарушено - число составное
            return  # Завершаем функцию досрочно
        end
    end
    # Если все тесты пройдены
    println("Число $n - ВЕРОЯТНО ПРОСТОЕ (пройдено $k тестов)")  # Выводим
```

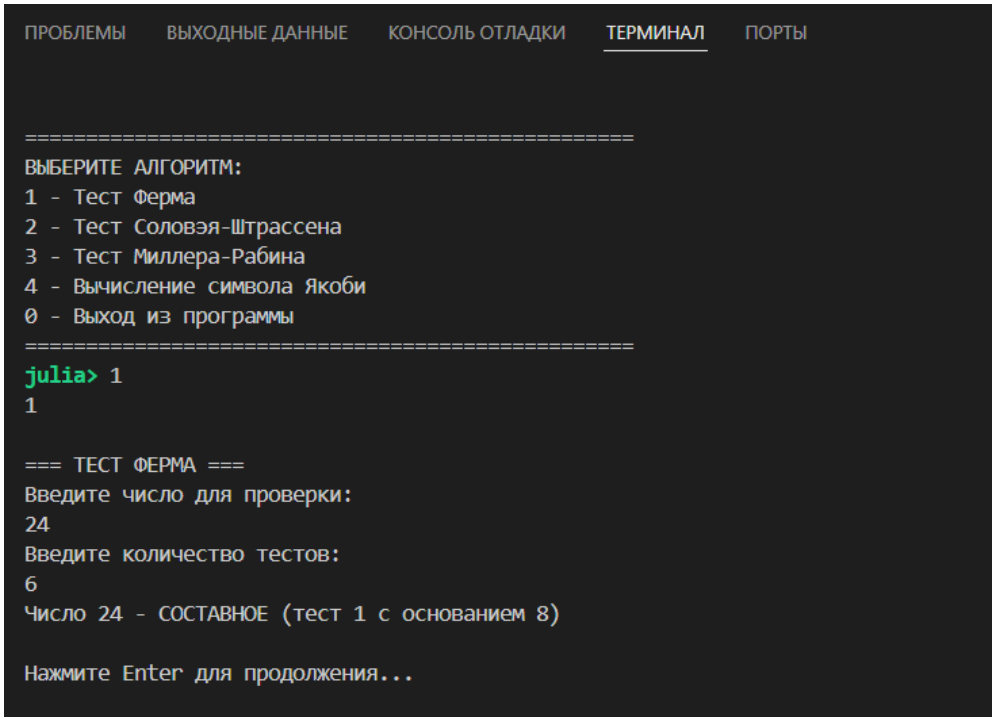
```
вероятностный результат
end

# Тест Соловэя-Штрассена - более надежный тест, использующий символ Якоби
function solovay_strassen_test()
    println("\n=== ТЕСТ СОЛОВЭЯ-ШТРАССЕНА ===")  # Заголовок теста
    println("Введите число для проверки:")  # Запрос числа n
    n = parse(Int, readline())  # Чтение числа n
    println("Введите количество тестов:")  # Запрос количества проверок
    k = parse(Int, readline())  # Чтение количества тестов

    # Выполняем k независимых тестов
    for i in 1:k
        a = rand(2:(n-2))  # Генерируем случайное основание a
        r = powermod(a, (n-1)÷2, n)  # Вычисляем a^((n-1)/2) mod n (критерий Эйлера)
        s = jacobi_symbol(a, n)  # Вычисляем символ Якоби (a/n)

        # Проверяем условия простоты
        if r != 1 && r != n-1  # Если r ≠ 1 и r ≠ n-1
            println("Число $n - СОСТАВНОЕ (тест $i с основанием $a)")  # Число
составное
            return
        elseif r % n != s % n  # Если r не равно символу Якоби по модулю n
            println("Число $n - СОСТАВНОЕ (тест $i с основанием $a)")  # Число
составное
            return
        end
    end
    println("Число $n - ВЕРОЯТНО ПРОСТОЕ (пройдено $k тестов)")  # Все тесты
пройдены
end

# Тест Миллера-Рабина - наиболее надежный вероятностный тест
function miller_rabin_test()
    println("\n=== ТЕСТ МИЛЛЕРА-РАБИНА ===")  # Заголовок теста
    println("Введите число для проверки:")  # Запрос числа n
    n = parse(Int, readline())  # Чтение числа n
    println("Введите количество тестов:")  # Запрос количества проверок
    k = parse(Int, readline())  # Чтение количества тестов

    # Представляем n-1 в виде 2^s * d, где d - нечетное
    s, d = 0, n-1  # Инициализация: s - степень двойки, d - нечетная часть
```

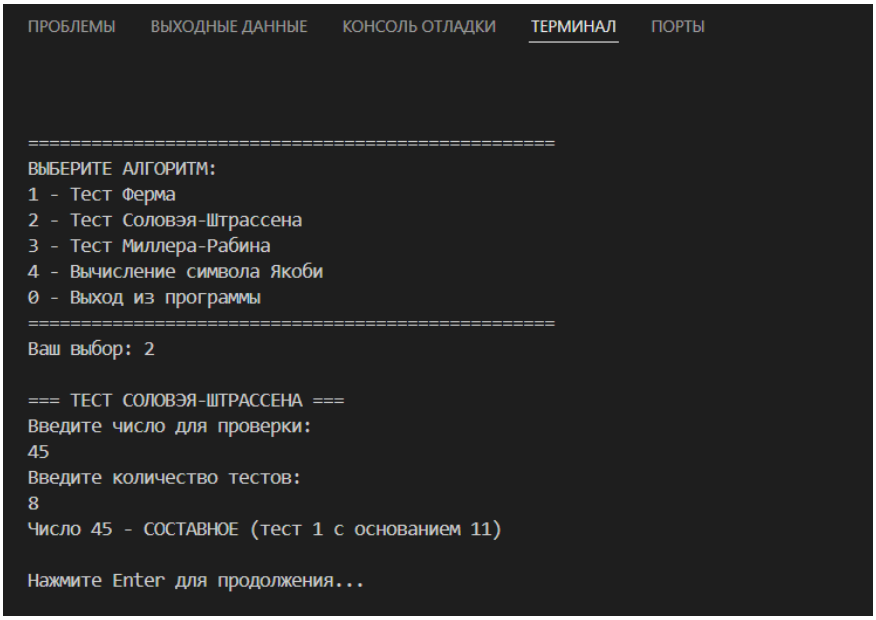
```
    while iseven(d)  # Пока d четное
        s += 1  # Увеличиваем счетчик степени
        d ÷= 2  # Делим d на 2
    end

    # Выполняем k независимых тестов
    for i in 1:k
        a = rand(2:(n-2))  # Генерируем случайное основание a
        x = powermod(a, d, n)  # Вычисляем x = a^d mod n

        # Проверяем тривиальные случаи
        if x != 1 && x != n-1  # Если x ≠ 1 и x ≠ n-1, нужна дополнительная проверка
            composite = true  # Предполагаем, что число составное
            for j in 1:s-1  # Проверяем последовательные квадраты
                x = (x * x) % n  # Возводим в квадрат по модулю n
                if x == n-1  # Если нашли n-1
                    composite = false  # Число вероятно простое
                    break  # Прерываем внутренний цикл
                end
            end
            if composite  # Если все проверки провалились
                println("Число $n - СОСТАВНОЕ (тест $i с основанием $a)")  # Число
составное
                return
            end
        end
    end
    println("Число $n - ВЕРОЯТНО ПРОСТОЕ (пройдено $k тестов)")  # Все тесты
пройдены
end

# Функция для вычисления символа Якоби как отдельная операция
function jacobi_calculation()
    println("\n=== ВЫЧИСЛЕНИЕ СИМВОЛА ЯКОБИ ===")  # Заголовок
    println("Введите число a:")  # Запрос числа a
    a = parse(Int, readline())  # Чтение числа a
    println("Введите нечетное число n ≥ 3:")  # Запрос модуля n
    n = parse(Int, readline())  # Чтение модуля n

    # Проверка корректности входных данных
    if n < 3 || iseven(n)  # Если n < 3 или четное
        println("Ошибка: n должно быть нечетным числом ≥ 3")  # Сообщение об
```

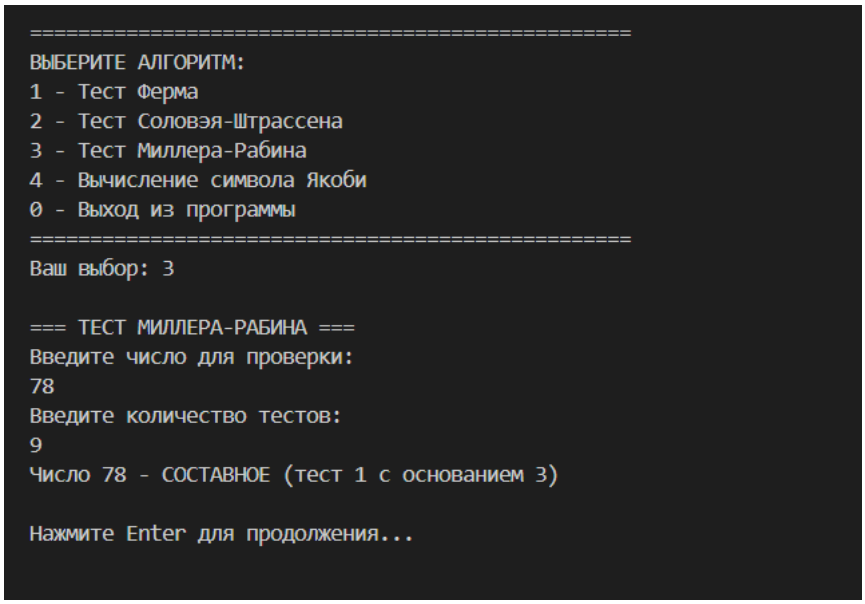
```
ошибке
        return  # Завершаем функцию
    end

    result = jacobi_symbol(a, n)  # Вычисляем символ Якоби
    println("Символ Якоби ($a/$n) = $result")  # Выводим результат

    # Дополнительная интерпретация результата
    if result == 1
        println("Это означает, что a является квадратичным вычетом по модулю n")  #
a - квадратичный вычет
    elseif result == -1
        println("Это означает, что a является квадратичным невычетом по модулю n")
# a - невычет
    else
        println("Это означает, что a и n не взаимно просты")  # Числа имеют общие
делители
    end
end

# Главная функция программы с интерактивным меню
function main()
    while true  # Бесконечный цикл для многократного использования
        println("\n" * "="^50)  # Разделительная линия
        println("ВЫБЕРИТЕ АЛГОРИТМ:")  # Заголовок меню
        println("1 - Тест Ферма")  # Опция 1
        println("2 - Тест Соловэя-Штрассена")  # Опция 2
        println("3 - Тест Миллера-Рабина")  # Опция 3
        println("4 - Вычисление символа Якоби")  # Опция 4
        println("0 - Выход из программы")  # Опция выхода
        println("="^50)  # Разделительная линия

        print("Ваш выбор: ")  # Приглашение для ввода
        choice = parse(Int, readline())  # Чтение выбора пользователя

        # Обработка выбора пользователя
        if choice == 0
            println("Выход из программы...")  # Сообщение о выходе
            break  # Прерываем цикл - завершаем программу
        elseif choice == 1
            fermat_test()  # Вызов теста Ферма
        elseif choice == 2
```

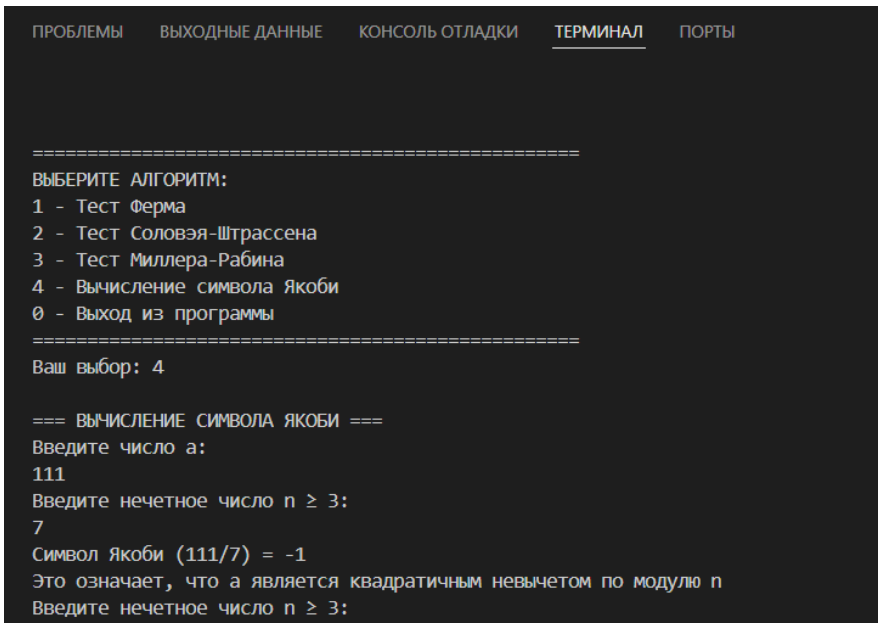
```
            solovay_strassen_test()  # Вызов теста Соловэя-Штрассена
        elseif choice == 3
            miller_rabin_test()  # Вызов теста Миллера-Рабина
        elseif choice == 4
            jacobi_calculation()  # Вызов вычисления символа Якоби
        else
            println("Неверный выбор! Попробуйте снова.")  # Сообщение об ошибке
        end

        # Пауза перед возвратом в меню
        println("\nНажмите Enter для продолжения...")  # Приглашение продолжить
        readline()  # Ожидание нажатия Enter
    end
end

main()  # Запуск главной функции программы
```

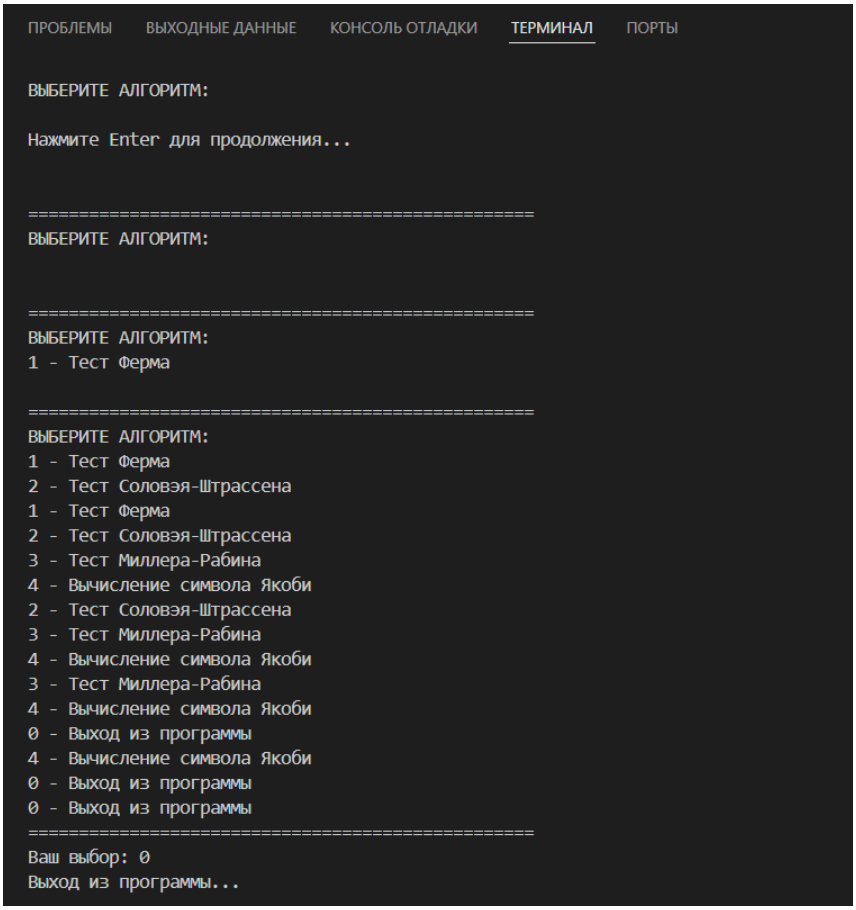
Результат реализации всех рассмотренных алгоритмов (тестов)




ПРОБЛЕМЫ
ВЫХОДНЫЕ ДАННЫЕ
КОНСОЛЬ ОТЛАДКИ
ТЕРМИНАЛ
ПОРТЫ
==========
ВЫБЕРИТЕ АЛГОРИТМ:
1 - Тест Ферма
2 - Тест Соловэя-Штрассена
3 - Тест Миллера-Рабина
4 - Вычисление символа Якоби
0 - Выход из программы
==========
Ваш выбор: 2
=== ТЕСТ СОЛОВЭЯ-ШТРАССЕНА ===
Введите число для проверки:
45
Введите количество тестов:
8
Число 45 - СОСТАВНОЕ (тест 1 с основанием 11)
Нажмите Enter для продолжения...


==========
ВЫБЕРИТЕ АЛГОРИТМ:
1 - Тест Ферма
2 - Тест Соловэя-Штрассена
3 - Тест Миллера-Рабина
4 - Вычисление символа Якоби
0 - Выход из программы
==========
Ваш выбор: 3
=== ТЕСТ МИЛЛЕРА-РАБИНА ===
Введите число для проверки:
78
Введите количество тестов:
9
Число 78 - СОСТАВНОЕ (тест 1 с основанием 3)
Нажмите Enter для продолжения...


ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ
==========
ВЫБЕРИТЕ АЛГОРИТМ:
1 - Тест Ферма
2 - Тест Соловэя-Штрассена
3 - Тест Миллера-Рабина
4 - Вычисление символа Якоби
0 - Выход из программы
==========
Ваш выбор: 4
=== ВЫЧИСЛЕНИЕ СИМВОЛА ЯКОБИ ===
Введите число a:
111
Введите нечетное число n ≥ 3:
7
Символ Якоби (111/7) = -1
Это означает, что a является квадратичным невычетом по модулю n
Введите нечетное число n ≥ 3:


ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ
ВЫБЕРИТЕ АЛГОРИТМ:
Нажмите Enter для продолжения...
==========
ВЫБЕРИТЕ АЛГОРИТМ:
==========
ВЫБЕРИТЕ АЛГОРИТМ:
1 - Тест Ферма
==========
ВЫБЕРИТЕ АЛГОРИТМ:
1 - Тест Ферма
2 - Тест Соловэя-Штрассена
1 - Тест Ферма
2 - Тест Соловэя-Штрассена
3 - Тест Миллера-Рабина
4 - Вычисление символа Якоби
2 - Тест Соловэя-Штрассена
3 - Тест Миллера-Рабина
4 - Вычисление символа Якоби
3 - Тест Миллера-Рабина
4 - Вычисление символа Якоби
0 - Выход из программы
4 - Вычисление символа Якоби
0 - Выход из программы
0 - Выход из программы
==========
Ваш выбор: 0
Выход из программы...