

Аннотированные статьи и ресурсы по языку программирования Julia:

- **Официальный сайт Julia** - На официальном сайте вы найдете множество учебных материалов, включая вводные курсы, интенсивные мастер-классы и руководства по использованию Julia в науке и инженерии.
- **Julia. Язык программирования. Быстрый старт** - автором данной книги выступает Вадим Никитин. В своей книге он простым языком объясняет главные и важные темы, а так же есть примеры решения задач.
- **Язык Julia как инструмент исследователя** - В. А. Антонюк, написал это пособие, чтобы познакомить читателей с новым языком программирования Julia, присутствуют примеры решений задач.
- **Язык программирования математических вычислений Julia. Базовое руководство** - учебное-методическое пособие Шиндина содержит сведения по установке интерпретатора языка Julia, принципах работы, основных конструкциях и возможностях языка.
- **Осваиваем язык Julia** - Малькольм Шеррингтон написал эту книгу для для специалистов в области анализа данных, а также разработчиков, желающих познакомиться с новым языком программирования.
- **Научное программирование на языке JULIA** - Евгения Александровна Оконешникова привела решения решения задач математической физики средствами языка программирования Julia.
- **Научное программирование на языке Julia** - Камиль Хайруллин сделал презентацию содержащую особенности языка, средства разработки.

Примеры решения:

1. Сумма элементов массива:

```
1  function sum_array(arr)
2      sum = 0
3      for elem in arr
4          sum += elem
5      end
6      return sum
7  end
8
9  arr = [1, 2, 3, 4, 5]
10 println("Сумма элементов массива: ", sum_array(arr))
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

Сумма элементов массива: 15

○ julia>

1. Функция `sum_array(arr)`: Определяет функцию, которая принимает массив `arr` в качестве аргумента.
2. Инициализация суммы: Переменная `sum` инициализируется значением 0.
3. Цикл `for`: Проходит по каждому элементу массива `arr`.
4. Суммирование элементов: Каждый элемент массива добавляется к переменной `sum`.
5. Возврат суммы: Функция возвращает итоговую сумму элементов массива.
6. Пример использования: Создается массив `[1, 2, 3, 4, 5]`, и функция `sum_array` вызывается для этого массива.
7. Вывод результата: Результат (сумма элементов массива) выводится на экран с помощью `println`.

2. Факториал числа:

```
1  function factorial(n)
2      if n == 0
3          return 1
4      else
5          return n * factorial(n - 1)
6      end
7  end
8
9  num = 5
10 println("Факториал числа $num: ", factorial(num))
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

Факториал числа 5: 120

Julia> █

1. Функция factorial(n):

- Если n равно 0, функция возвращает 1 (потому что факториал 0 равен 1).
- В противном случае, функция рекурсивно вызывает саму себя с аргументом $n - 1$ и умножает результат на n .

2. Переменная num: Присваивается значение 5.

3. Вывод результата: Функция println выводит строку "Факториал числа 5: " и результат вызова функции factorial(5).

3. Решение квадратного уравнения:

```
1 function solve_quadratic(a, b, c)
2     discriminant = b^2 - 4*a*c
3     if discriminant < 0
4         return "Нет действительных корней"
5     elseif discriminant == 0
6         x = -b / (2*a)
7         return "Один корень: $x"
8     else
9         x1 = (-b + sqrt(discriminant)) / (2*a)
10        x2 = (-b - sqrt(discriminant)) / (2*a)
11        return "Два корня: $x1 и $x2"
12    end
13 end
14
15 a, b, c = 1, -3, 2
16 println("Решение квадратного уравнения: ", solve_quadratic(a, b, c))
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

Решение квадратного уравнения: Два корня: 2.0 и 1.0

○ julia> █

1. Функция `solve_quadratic(a, b, c)`: Эта функция решает квадратное уравнение вида $(ax^2 + bx + c = 0)$.
2. Дискриминант: Вычисляется дискриминант $(b^2 - 4ac)$.
3. Проверка дискриминанта:
 - Если дискриминант меньше 0, возвращается сообщение "Нет действительных корней".
 - Если дискриминант равен 0, вычисляется единственный корень $(x = -\frac{b}{2a})$ и возвращается сообщение "Один корень: \$x".
 - Если дискриминант больше 0, вычисляются два корня $(x1)$ и $(x2)$ по формулам $(x1 = \frac{-b + \sqrt{\text{discriminant}}}{2a})$ и $(x2 = \frac{-b - \sqrt{\text{discriminant}}}{2a})$, и возвращается сообщение "Два корня: \$x1 и \$x2".

4. Пример использования: В примере функции передаются значения $(a = 1)$, $(b = -3)$, $(c = 2)$. Функция вычисляет корни и выводит результат.