

Мироненко Артём Сергеевич ИС231
Практическая работа 8
Функциональное программирование

1. Напишите функцию, которая принимает список на вход, и возвращает сумму всех элементов этого списка.

```
fun main() {  
    val numbers = listOf(5, 7, 3, 2, 8)  
    println(sumOfList(numbers))  
}  
  
fun sumOfList(list: List<Int>): Int {  
    return list.sum()  
}
```

```
"C:\Users\User\AppData\Local\Programs\Java\jdk-17.0.2\bin\java.exe" -Djava.class.path=C:\Users\User\AppData\Local\Programs\Java\jdk-17.0.2\bin\java.exe -jar C:\Users\User\AppData\Local\Programs\Java\jdk-17.0.2\bin\java.exe  
25  
  
Process finished with exit code 0
```

2. Напишите функцию, которая получает на вход список целых чисел и возвращает разность самого большого и самого маленького из них.

```
fun maxMinDifference(numbers: List<Int>): Int {  
    if (numbers.isEmpty()) return 0  
    return numbers.maxOrNull()!! - numbers.minOrNull()!!  
}  
  
fun main() {  
    val numbers = listOf(1, 2, 3, 4, 5, 6, 7)  
    println(maxMinDifference(numbers))  
}
```

```
"C:\Users\User\AppData\Local\Programs\Java\jdk-17.0.2\bin\java.exe" -Djava.class.path=C:\Users\User\AppData\Local\Programs\Java\jdk-17.0.2\bin\java.exe -jar C:\Users\User\AppData\Local\Programs\Java\jdk-17.0.2\bin\java.exe  
6  
  
Process finished with exit code 0
```

3. Создайте функцию для объединения двух списков целых чисел.

```
fun main() {  
    val list1 = listOf(4, 3, 2, 1)  
    val list2 = listOf(8, 7, 6, 5)  
    println(mergeLists(list1, list2))  
}  
  
fun mergeLists(list1: List<Int>, list2: List<Int>): List<Int> {  
    val result = mutableListOf<Int>()  
    result.addAll(list1)  
    result.addAll(list2)  
}
```

```
    return result
}
```

```
"C:\Users\User\AppData\Local\Programs\Microsoft VS Code\bin\code.exe"
[4, 3, 2, 1, 8, 7, 6, 5]

Process finished with exit code 0
```

4. Создайте функцию, которая принимает три аргумента prob, prize, pay и возвращает True, если $\text{prob} * \text{prize} > \text{pay}$, в противном случае возвращает False.

```
fun evaluateBet(prob: Double, prize: Double, pay: Double): Boolean {
    return prob * prize > pay
}

fun main() {
    val prob = 5.2
    val prize = 28.0
    val pay = 42.0

    val result = evaluateBet(prob, prize, pay)
    println(result)
}
```

```
"C:\Users\User\AppData\Local\Programs\Microsoft VS Code\bin\code.exe"
true

Process finished with exit code 0
```

5. Функция получает на вход два числа. Она должна вернуть True, если сумма этих чисел меньше 100 и False в противном случае.

```
fun isSumLessThan100(num1: Int, num2: Int): Boolean {
    return (num1 + num2) < 100
}

fun main() {
    val number1 = 90
    val number2 = 9

    val result = isSumLessThan100(number1, number2)
    println(result)
}
```

```
"C:\Users\User\AppData\Local\Programs\Microsoft VS Code\bin\code.exe"
true

Process finished with exit code 0
```

6. Напишите функцию, которая принимает целое число и возвращает True, если оно делится на 100. В противном случае функция должна вернуть False.

```
fun main() {
    fun isDivisibleBy100(number: Int): Boolean {
        return number % 100 == 0
    }
    println(isDivisibleBy100(100)) // true
    println(isDivisibleBy100(200)) // true
    println(isDivisibleBy100(150)) // false
    println(isDivisibleBy100(1000)) // true
}
```

```
"C:\Users\User\AppData\Local\Programs\Python\Python39\python.exe"
true
true
false
true

Process finished with exit code 0
```

7. Напишите функцию, которая принимает количество минут и частоту кадров (FPS) и возвращает, сколько за это время кадров показывает компьютер при этом FPS.

```
fun calculateFrames(minutes: Int, fps: Int): Int {
    val totalSeconds = minutes * 60
    return totalSeconds * fps
}

fun main() {
    val minutes = 2
    val fps = 144

    val totalFrames = calculateFrames(minutes, fps)
    println(totalFrames)
}
```

```
"C:\Users\User\AppData\Local\Programs\Python\Python39\python.exe"
17280

Process finished with exit code 0
```

8. Напишите функцию, которая возвращает True, если $k^k == n$ для входных данных (n, k), и возвращает False в противном случае.

```
fun isKPowerEqualToN(n: Int, k: Int): Boolean {
    return Math.pow(k.toDouble(), k.toDouble()) == n.toDouble()
}
```

```

}

fun main() {
    val n = 4
    val k = 2

    val result = isKPowerEqualToN(n, k)
    println(result)
}

```

```

"C:\Users\User\AppData\Local\Programs\
true

Process finished with exit code 0

```

9. Создайте рекурсивную функцию, которая принимает два параметра и повторяет строку n количество раз. Первый параметр txt - это строка, которую нужно повторить, а второй параметр - количество повторений строки.

Примеры

```

repetition("ab", 3) → "ababab"
repetition("kiwi", 1) → "kiwi"
repetition("cherry", 2) → "cherrycherry"

```

```

fun repeatString(txt: String, n: Int): String {
    return if (n <= 0) ""
    else txt + repeatString(txt, n - 1)
}

fun main() {
    val text = "bluh"
    val repetitions = 4

    val result = repeatString(text, repetitions)
    println(result)
}

```

```

"C:\Users\User\AppData\Local\Programs\
bluhbluhbluhbluh

Process finished with exit code 0

```

10. Создайте функцию, которая принимает уравнение (например, "1+1") и возвращает ответ.

```

fun main() {
    println("Введите выражение (например 1+1 или 7*4-2):")
    val input = readLine()!!
}

```

```

fun equation(e: String): Int {
    val nums = e.split("+", "-", "*")
    var result = nums[0].toInt()
    var i = 0

    e.forEach { c ->
        when(c) {
            '+' -> { i++; result += nums[i].toInt() }
            '-' -> { i++; result -= nums[i].toInt() }
            '*' -> { i++; result *= nums[i].toInt() }
        }
    }
    return result
}

println("Результат: ${equation(input)}")
}

```

```

"C:\Users\User\AppData\Local\Programs\Eclipse
Введите выражение (например 1+1 или 7*4-2):
2*2+5
Результат: 9
Process finished with exit code 0

```

11. Напишите функцию, которая принимает число `number`, и возвращает слово Google с количеством букв `o`, равным `number`.

Примеры

```

google(5) => "Goooooogle"
google(0) => "Ggle"
google(2) => "Google"

```

```

fun generateGoogleWord(number: Int): String {
    if (number < 0) {
        throw IllegalArgumentException("Количество букв o не может быть отрицательным")
    }

    return "G" + "o".repeat(number) + "gle"
}

fun main() {
    val number = 6

    val result = generateGoogleWord(number)
    println(result)
}

```

```
"C:\Users\User\AppData\Local\Programs\
Go00000ogle

Process finished with exit code 0
```

12. Приветствие: Напишите функцию, которая выводит "Привет, мир!" на экран.

```
fun main() {
    print("Привет мир!")
}
```

```
"C:\Users\User\AppData\Local\Programs\E
Привет мир!
Process finished with exit code 0
```

13. Сумма двух чисел: Создайте функцию, которая принимает два числа и возвращает их сумму.

```
fun main() {
    print("Введите первое число: ")
    val a = readLine()!!.toInt()
    print("Введите второе число: ")
    val b = readLine()!!.toInt()
    println("Сумма: ${a + b}")
}
```

```
"C:\Users\User\AppData\Local\Program
Введите первое число: 3
Введите второе число: 3
Сумма: 6

Process finished with exit code 0
```

14. Сравнение чисел: Напишите функцию, которая принимает два числа и возвращает большее из них.

```
fun main() {
    print("Введите первое число: ")
    val a = readLine()!!.toInt()
    print("Введите второе число: ")
    val b = readLine()!!.toInt()
    println("Большее число: ${if (a > b) a else b}")
}
```

```
"C:\Users\User\AppData\Local\Programs\
Введите первое число: 4
Введите второе число: 6
Большее число: 6

Process finished with exit code 0
```

15. Определение четности: Создайте функцию, которая принимает одно число и возвращает true, если оно четное, и false, если нечетное.

```
fun main() {
    print("Введите число: ")
    val num = readLine()!!.toInt()
    val isEven = num % 2 == 0
    println("Число ${if (isEven) "четное" else "нечетное"}")
}
```

```
"C:\Users\User\AppData\Local\Programs\
Введите число: 10
Число четное

Process finished with exit code 0
```

16. Факториал числа: Реализуйте функцию, которая вычисляет факториал введенного числа.

```
fun main() {
    print("Введите число: ")
    val num = readLine()!!.toInt()

    fun factorial(n: Int): Int {
        return if (n <= 1) 1 else n * factorial(n - 1)
    }

    println("Факториал числа $num = ${factorial(num)}")
}
```

```
"C:\Users\User\AppData\Local\Programs\
Введите число: 3
Факториал числа 3 = 6

Process finished with exit code 0
```

17. Проверка на простоту: Напишите функцию, которая проверяет, является ли заданное число простым.

```
fun main() {
    print("Введите число: ")
```

```

val num = readLine()!!.toInt()

fun isPrime(n: Int): Boolean {
    if (n <= 1) return false
    if (n == 2) return true

    for (i in 2..kotlin.math.sqrt(n.toDouble()).toInt()) {
        if (n % i == 0) return false
    }
    return true
}

println("Число $num ${if (isPrime(num)) "простое" else "не простое"}")
}

```

```

"C:\Users\User\AppData\Local\Programs\B
Введите число: 3
Число 3 простое

Process finished with exit code 0

```

18. Сумма чисел в массиве: Создайте функцию, которая принимает массив целых чисел и возвращает их сумму.

```

fun main() {
    print("Введите числа через пробел: ")
    val numbers = readLine()!!.split(" ").map { it.toInt() }
    val sum = numbers.sum()

    println("Массив: ${numbers.joinToString()}")
    println("Сумма всех чисел: $sum")
}

```

```

"C:\Users\User\AppData\Local\Programs\Ecl
Введите числа через пробел: 1 2 3
Массив: 1, 2, 3
Сумма всех чисел: 6

Process finished with exit code 0

```

19. Наибольшее число в массиве: Напишите функцию, которая находит и возвращает максимальное значение в массиве.

```

fun main() {
    print("Введите числа через пробел: ")
    val numbers = readLine()!!.split(" ").map { it.toInt() }
}

```



```

val max = numbers.maxOrNull()

println("Массив: ${numbers.joinToString()}")
println("Максимальное число: $max")
}

```

```

"C:\Users\User\AppData\Local\Programs\Ec
Введите числа через пробел: 1 2 3
Массив: 1, 2, 3
Максимальное число: 3

Process finished with exit code 0

```

20. Сортировка массива: Создайте функцию, которая сортирует массив чисел в порядке возрастания.

```

fun main() {
    print("Введите числа через пробел: ")
    val numbers = readLine()!!.split(" ").map { it.toInt() }
        .toMutableList()

    fun bubbleSort(arr: MutableList<Int>) {
        for (i in 0 until arr.size - 1) {
            for (j in 0 until arr.size - i - 1) {
                if (arr[j] > arr[j + 1]) {
                    val temp = arr[j]
                    arr[j] = arr[j + 1]
                    arr[j + 1] = temp
                }
            }
        }
    }

    println("Исходный массив: ${numbers.joinToString()}")
    bubbleSort(numbers)
    println("Отсортированный массив: ${numbers.joinToString()}")
}

```

```

C:\Users\User\AppData\Local\Programs\Eclipse Adoptium\jdk
Введите числа через пробел: 3 5 6 4 3 7 8 9 1 3 4
Исходный массив: 3, 5, 6, 4, 3, 7, 8, 9, 1, 3, 4
Отсортированный массив: 1, 3, 3, 3, 4, 4, 5, 6, 7, 8, 9

Process finished with exit code 0

```

21. Проверка палиндрома: Напишите функцию, которая проверяет, является ли строка палиндромом.

```

fun main() {
    print("Введите строку: ")
    val input = readLine()!!.lowercase().replace(Regex("[^a-za-я0-9]"),
    "")

    fun isPalindrome(str: String): Boolean {
        return str == str.reversed()
    }

    println("Строка ${if (isPalindrome(input)) "является" else "не
является"} палиндромом")
}

```

```

"C:\Users\User\AppData\Local\Programs\Eclipse
Введите строку: заказ
Строка является палиндромом

Process finished with exit code 0

```

22. Количество символов: Создайте функцию, которая принимает строку и возвращает количество символов в ней.

```

fun main() {
    print("Введите строку: ")
    val input = readLine()!!

    fun countCharacters(str: String): Int {
        return str.length
    }

    val length = countCharacters(input)
    println("Количество символов в строке: $length")
}

```

```

"C:\Users\User\AppData\Local\Program
Введите строку: привет
Количество символов в строке: 6

Process finished with exit code 0

```

23. Конвертация в верхний регистр: Напишите функцию, которая принимает строку и возвращает её в верхнем регистре.

```

fun main() {
    print("Введите строку: ")
    val input = readLine()!!

```

```
fun toUpperCase(str: String): String {  
    return str.uppercase()  
}  
  
val result = toUpperCase(input)  
println("Результат: $result")  
}
```

```
"C:\Users\User\AppData\Local\Program  
Введите строку: как дела  
Результат: КАК ДЕЛА  
  
Process finished with exit code 0
```

24. Объединение строк: Создайте функцию, которая принимает две строки и возвращает их объединение.

```
fun main() {
    print("Введите первую строку: ")
    val string1 = readLine()!!

    print("Введите вторую строку: ")
    val string2 = readLine()!!

    fun concatenateStrings(str1: String, str2: String): String {
        return str1 + str2
    }

    val result = concatenateStrings(string1, string2)
    println("Результат объединения: $result")
}
```

```
"C:\Users\User\AppData\Local\Programs\Eclipse Adoptium\
Введите первую строку: все нормально,
Введите вторую строку: а у тебя как?
Результат объединения: все нормально, а у тебя как?
|
Process finished with exit code 0
```

25. Возвращение последнего элемента массива: Напишите функцию, которая возвращает последний элемент переданного массива.

```
fun main() {
    print("Введите элементы массива через пробел: ")
    val array = readLine()!!.split(" ")

    fun getLastElement(arr: List<String>): String {
        if (arr.isEmpty()) {
            return "Массив пуст"
        }
        return arr.last()
    }

    println("Массив: ${array.joinToString()}")
    println("Последний элемент: ${getLastElement(array)}")
}
```

```
"C:\Users\User\AppData\Local\Programs\Eclipse Adoptium\
Введите элементы массива через пробел: 1 2 3 4 5 6
Массив: 1, 2, 3, 4, 5, 6
Последний элемент: 6
Process finished with exit code 0
```

26. Проверка наличия элемента: Создайте функцию, которая проверяет, присутствует ли элемент в массиве.

```
fun main() {
    print("Введите элементы массива через пробел: ")
    val array = readLine()!!.split(" ")

    print("Введите элемент для поиска: ")
    val searchElement = readLine()!!

    fun containsElement(arr: List<String>, element: String): Boolean {
        return element in arr
    }

    val result = containsElement(array, searchElement)
    println("Массив: ${array.joinToString()}")
    println("Элемент '$searchElement' ${if (result) "найден" else "не найден"} в массиве")
}
```

```
"C:\Users\User\AppData\Local\Programs\Eclipse Adopti
Введите элементы массива через пробел: 2 4 6 8 10
Введите элемент для поиска: 4
Массив: 2, 4, 6, 8, 10
Элемент '4' найден в массиве

Process finished with exit code 0
```

27. Создание массива от 1 до N: Напишите функцию, которая создает массив целых чисел от 1 до N.

```
fun main() {
    print("Введите число N: ")
    val n = readLine()!!.toInt()

    fun createArray(n: Int): List<Int> {
        return (1..n).toList()
    }

    val result = createArray(n)
    println("Массив от 1 до $n: ${result.joinToString()}")
}
```

```
"C:\Users\User\AppData\Local\Programs\Ec
Введите число N: 5
Массив от 1 до 5: 1, 2, 3, 4, 5

Process finished with exit code 0
```

28. Максимум и минимум: Создайте функцию, которая находит одновременно максимальное и минимальное значения в массиве.

```
fun main() {
    print("Введите элементы массива через пробел: ")
    val array = readLine()!!.split(" ").map { it.toInt() }

    fun findMinMax(arr: List<Int>): Pair<Int, Int> {
        val min = arr.minOrNull() ?: 0
        val max = arr.maxOrNull() ?: 0
        return Pair(min, max)
    }

    val (min, max) = findMinMax(array)
    println("Минимальное значение: $min")
    println("Максимальное значение: $max")
}
```

```
"C:\Users\User\AppData\Local\Programs\Eclipse Adoptium
Введите элементы массива через пробел: 2 5 7 9 15
Минимальное значение: 2
Максимальное значение: 15

Process finished with exit code 0
```

29. Сумма чисел от 1 до N: Напишите функцию, которая рассчитывает сумму всех чисел от 1 до N.

```
fun main() {
    print("Введите число N: ")
    val n = readLine()!!.toInt()

    fun sumUpTo(n: Int): Int {
        return (1..n).sum()
    }

    val result = sumUpTo(n)
    println("Сумма чисел от 1 до $n: $result")
}
```

```
"C:\Users\User\AppData\Local\Programs\Ecl
Введите число N: 6
Сумма чисел от 1 до 6: 21

Process finished with exit code 0
```

30. Преобразование Celsius в Fahrenheit: Создайте функцию, которая конвертирует температуру из Цельсия в Фаренгейт.

```
fun main() {
    print("Введите температуру в градусах Цельсия: ")
    val celsius = readLine()!!.toDouble()

    fun celsiusToFahrenheit(celsius: Double): Double {
        return (celsius * 9/5) + 32
    }

    val fahrenheit = celsiusToFahrenheit(celsius)
    println("$celsius градусов Цельсия = $fahrenheit градусов
Фаренгейта")
}
```

```
"C:\Users\User\AppData\Local\Programs\Eclipse Adopt
Введите температуру в градусах Цельсия: 5
5.0 градусов Цельсия = 41.0 градусов Фаренгейта

Process finished with exit code 0
```

31. Обратный порядок строки: Напишите функцию, которая принимает строку и возвращает её в обратном порядке.

```
fun main() {
    print("Введите строку: ")
    val input = readLine()!!
    println("Исходная строка: $input")
    println("Строка в обратном порядке: ${input.reversed()}")
}
```

```
"C:\Users\User\AppData\Local\Programs\Ecl
Введите строку: хелоу мир
Исходная строка: хелоу мир
Строка в обратном порядке: рим уолех

Process finished with exit code 0
```

32. Поиск элемента по индексу: Создайте функцию, которая находит элемент по индексу в массиве.

```
fun main() {  
    val array = listOf(10, 20, 30, 40, 50)  
    print("Введите индекс элемента, который вы хотите найти: ")  
    val index = readLine()!!.toInt()  
  
    if (index in array.indices) {  
        println("Элемент на индексе $index: ${array[index]}")  
    } else {  
        println("Индекс $index вне диапазона массива.")  
    }  
}
```

```
"C:\Users\User\AppData\Local\Programs\  
Введите индекс элемента, который вы хо  
Элемент на индексе 3: 40  
  
Process finished with exit code 0
```

33. Удаление пробелов из строки: Напишите функцию, которая удаляет все пробелы из переданной строки.

```
fun main() {  
    print("Введите строку: ")  
    val input = readLine()!!  
    println("Исходная строка: $input")  
    println("Строка без пробелов: ${input.replace(" ", "")}")  
}
```

```
"C:\Users\User\AppData\Local\Programs\Eclipse Ad  
Введите строку: пример с пробелом п р и в е т  
Исходная строка: пример с пробелом п р и в е т  
Строка без пробелов: примерспробеломпривет  
  
Process finished with exit code 0
```

34. Сумма первых N натуральных чисел: Создайте функцию, которая возвращает сумму первых N натуральных чисел.

```
fun main() {  
    print("Введите число N: ")  
    val n = readLine()!!.toInt()  
    println("Сумма первых $n натуральных чисел: ${sumUpToN(n)}")  
}
```



```

}

fun sumUpToN(n: Int): Int {
    return (1..n).sum()
}

```

```

"C:\Users\User\AppData\Local\Programs\Ecl
Введите число N: 5
Сумма первых 5 натуральных чисел: 15

Process finished with exit code 0

```

35. Проверка строки на наличие подстроки: Напишите функцию, которая проверяет, содержится ли одна строка в другой.

```

fun main() {
    print("Введите исходную строку: ")
    val mainString = readLine()!!
    print("Введите подстроку для поиска: ")
    val subString = readLine()!!

    if (isSubstringPresent(mainString, subString)) {
        println("Подстрока '$subString' найдена в строке '$mainString'")
    } else {
        println("Подстрока '$subString' не найдена в строке '$mainString'")
    }
}

fun isSubstringPresent(mainString: String, subString: String): Boolean
{
    return mainString.contains(subString)
}

```

```

"C:\Users\User\AppData\Local\Programs\Eclipse Adoptium\
Введите исходную строку: первая строка
Введите подстроку для поиска: первая
Подстрока 'первая' найдена в строке 'первая строка'

Process finished with exit code 0

```

36. Печать таблицы умножения: Создайте функцию, которая выводит таблицу умножения от 1 до 10 для заданного числа.

```

fun main() {
    print("Введите число для таблицы умножения: ")
    val number = readLine()!!.toInt()
}

```

```

        printMultiplicationTable(number)
    }

    fun printMultiplicationTable(number: Int) {
        println("Таблица умножения для числа $number:")
        for (i in 1..10) {
            println("$number x $i = ${number * i}")
        }
    }
}

```

```

"C:\Users\User\AppData\Local\Programs\Eclips
Введите число для таблицы умножения: 5
Таблица умножения для числа 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

Process finished with exit code 0
|

```

37. Нахождение длины строки: Напишите функцию, которая возвращает длину переданной ей строки.

```

fun main() {
    print("Введите строку: ")
    val input = readLine()!!
    println("Длина строки: ${getStringLength(input)}")
}

fun getStringLength(str: String): Int {
    return str.length
}

```

```

"C:\Users\User\AppData\Local\Programs\Ec
Введите строку: мяук
Длина строки: 4

Process finished with exit code 0

```

38. Переворот массива: Создайте функцию, которая переворачивает массив целых чисел.

```
fun main() {
    val numbers = intArrayOf(1, 2, 3, 4, 5)
    println("Исходный массив: ${numbers.contentToString()}")
    println("Перевернутый массив:
    ${numbers.reversedArray().contentToString()}")
}
```

```
"C:\Users\User\AppData\Local\Programs\Ec
Исходный массив: [1, 2, 3, 4, 5]
Перевернутый массив: [5, 4, 3, 2, 1]

Process finished with exit code 0
```

39. Копирование массива: Напишите функцию, которая копирует массив и возвращает новый массив.

```
fun main() {
    val originalArray = intArrayOf(1, 2, 3, 4, 5)
    println("Исходный массив: ${originalArray.contentToString()}")

    val copiedArray = copyArray(originalArray)
    println("Скопированный массив: ${copiedArray.contentToString()}")
}

fun copyArray(arr: IntArray): IntArray {
    return arr.copyOf()
}
```

```
"C:\Users\User\AppData\Local\Programs\Ec
Исходный массив: [1, 2, 3, 4, 5]
Скопированный массив: [1, 2, 3, 4, 5]

Process finished with exit code 0
```

40. Количество гласных в строке: Создайте функцию, которая считает количество гласных в строке.

```
fun main() {
    println("Введите строку:")
    val input = readLine() ?: ""
    println("Количество гласных в строке: ${countVowels(input)}")
}

fun countVowels(str: String): Int {
```

```

    val vowels = setOf('a', 'e', 'ё', 'и', 'о', 'у', 'ы', 'э', 'ю',
'я', 'А', 'Е', 'Ё', 'И', 'О', 'У', 'Ы', 'Э', 'Ю', 'Я')
    return str.count { it in vowels }
}

```

```

"C:\Users\User\AppData\Local\Programs\
Введите строку:
лалолулелелялиэлэлюлы
Количество гласных в строке: 9

Process finished with exit code 0

```

41. Индекс первого вхождения: Напишите функцию, которая возвращает индекс первого вхождения элемента в массиве, или -1, если элемент не найден.

```

fun main() {
    println("Введите элементы массива через пробел:")
    val inputArray = readLine()?.split(" ")?.mapNotNull {
it.toIntOrNull() }?.toIntArray() ?: intArrayOf()

    println("Введите элемент для поиска:")
    val element = readLine()?.toIntOrNull()

    if (element != null) {
        val index = findFirstIndex(inputArray, element)
        println("Индекс первого вхождения элемента $element: $index")
    } else {
        println("Введено некорректное значение для поиска.")
    }
}

fun findFirstIndex(arr: IntArray, target: Int): Int {
    return arr.indexOf(target)
}

```

<pre> "C:\Users\User\AppData\Local\Programs\Eclips Введите элементы массива через пробел: 1 2 3 4 5 6 Введите элемент для поиска: 7 Индекс первого вхождения элемента 7: -1 Process finished with exit code 0 </pre>	<pre> "C:\Users\User\AppData\Local\Programs\Eclips Введите элементы массива через пробел: 1 2 3 4 5 6 Введите элемент для поиска: 3 Индекс первого вхождения элемента 3: 2 Process finished with exit code 0 </pre>
--	---