

## Практическая работа 8

### Функциональное программирование

Ермолаева Наталья ИС232

1

```
println("Введите размер списка:")

val size = scanner.nextInt()

val randomList = List(size) { Random.nextInt(1, 101) }

println("Сгенерированный список: $randomList")

println("Сумма всех элементов: ${sumList(randomList)}")

}
```

2

```
println("Введите размер списка:")

val numbers = MutableList(scanner.nextInt()) {

    println("Введите элемент ${it + 1}:")

    scanner.nextInt()

}

println("Разность между максимальным и минимальным:

${maxMinDifference(numbers)}")

}
```

3

```
fun readList(name: String): List<Int> {

    println("Введите размер списка $name:")

    return MutableList(scanner.nextInt()) {

        println("Введите элемент ${it + 1} для списка

$name:")

        scanner.nextInt()

    }

}
```

```

    }

    val list1 = readList("1")

    val list2 = readList("2")

    println("Объединенный список: ${combineLists(list1, list2)}")
}

```

4

```

    println("Введите prob, prize, pay через пробел:")

    val (prob, prize, pay) = List(3) { scanner.nextDouble() }

    println("Результат: ${checkProfit(prob, prize, pay)}")
}

```

5

```

    println("Введите два числа через пробел:")

    val (a, b) = List(2) { scanner.nextDouble() }

    println("Результат: ${sumLessThan100(a, b)}")
}

```

6

```

    println("Введите число:")

    println("Результат: ${divisibleBy100(scanner.nextInt())}")
}

```

7

```

    println("Введите минуты и FPS через пробел:")

    val (minutes, fps) = List(2) { scanner.nextDouble() }

    println("Количество кадров: ${calculateFrames(minutes,
fps}}")

}

```

8

```

    println("Введите n и k через пробел:")

```

```

        val (n, k) = List(2) { scanner.nextInt() }

        println("Результат: ${checkPowerEquality(n, k)}")
    }
9
    scanner.nextLine()

    println("Введите строку:")

    val txt = scanner.nextLine()

    println("Введите количество повторений:")

    val n = scanner.nextInt()

    println("Результат: ${repeatString(txt, n)}")
}
10
    scanner.nextLine()

    println("Введите уравнение (только с +):")

    val expr = scanner.nextLine()

    try {

        println("Результат: ${evaluateExpression(expr)}")

    } catch (e: Exception) {

        println("Ошибка вычисления: ${e.message}")

    }

}
11
    println("Введите количество 'о':")

    val oCount = scanner.nextInt()

    println(createCustomGoogle(oCount))

}
12

```

```
helloWorld()
```

13

```
println("Введите два числа через пробел:")  
  
val (a, b) = List(2) { scanner.nextDouble() }  
  
println("Сумма: ${sumTwoNumbers(a, b)}")  
  
}
```

14

```
println("Введите два числа через пробел:")  
  
val (a, b) = List(2) { scanner.nextDouble() }  
  
println("Большее число: ${findLargerNumber(a, b)}")  
  
}
```

15

```
println("Введите число:")  
  
println("Результат: ${isEven(scanner.nextInt())}")  
  
}
```

16

```
println("Введите число:")  
  
val n = scanner.nextInt()  
  
println("Факториал: ${factorial(n)}")  
  
}
```

17

```
println("Введите число:")  
  
val n = scanner.nextInt()  
  
println("Результат: ${isPrime(n)}")  
  
}
```

18

```
println("Введите размер массива:")
```

```

        val size = scanner.nextInt()

        val arr = List(size) { Random.nextInt(1, 101) }

        println("Массив: $arr\nСумма: ${sumArray(arr)}")
    }
19

    println("Введите размер массива:")

    val size = scanner.nextInt()

    val arr = List(size) { Random.nextInt(1, 101) }

    println("Массив: $arr\nМаксимум: ${findMax(arr)}")
}

20

    println("Введите размер массива:")

    val size = scanner.nextInt()

    val arr = MutableList(size) { Random.nextInt(1, 101) }

    println("Исходный массив: $arr")

    println("Отсортированный: ${sortArray(arr)}")
}

21

    scanner.nextLine()

    println("Введите строку:")

    val s = scanner.nextLine()

    println("Результат: ${isPalindrome(s)}")
}

22

    scanner.nextLine()

    println("Введите строку:")

    val s = scanner.nextLine()

```

```
println("Длина: ${stringLength(s)}")
}
```

23

```
scanner.nextLine()
println("Введите строку:")
val s = scanner.nextLine()
println("Результат: ${toUpperCase(s)}")
}
```

24

```
scanner.nextLine()
println("Введите первую строку:")
val s1 = scanner.nextLine()
println("Введите вторую строку:")
val s2 = scanner.nextLine()
println("Результат: ${concatenate(s1, s2)}")
}
```

25

```
println("Введите размер массива:")
val size = scanner.nextInt()
val arr = List(size) { Random.nextInt(1, 101) }
println("Массив: $arr\nПоследний элемент:
${lastElement(arr)}")
}
```

26

```
println("Введите размер массива:")
val size = scanner.nextInt()
val arr = List(size) { Random.nextInt(1, 101) }
```

```
println("Массив: $arr")

println("Введите искомый элемент:")

val element = scanner.nextInt()

println("Результат: ${containsElement(arr, element)}")

}
```

27

```
println("Введите N:")

val n = scanner.nextInt()

println("Массив: ${createSequence(n)}")

}
```

28

```
println("Введите размер массива:")

val size = scanner.nextInt()

val arr = List(size) { Random.nextInt(1, 101) }

println("Массив: $arr")

val (min, max) = findMinMax(arr)

println("Min: $min, Max: $max")

}
```

29

```
println("Введите N:")

val n = scanner.nextInt()

println("Сумма: ${sumFrom1ToN(n)}")

}
```

30

```
println("Введите температуру в °C:")

val c = scanner.nextDouble()

println("Результат: ${celsiusToFahrenheit(c)} °F")
```

```
}
```

```
31
```

```
    scanner.nextLine()

    println("Введите строку:")

    val s = scanner.nextLine()

    println("Результат: ${reverseString(s)}")
```

```
}
```

```
32
```

```
    println("Введите размер массива:")

    val size = scanner.nextInt()

    val arr = List(size) { Random.nextInt(1, 101) }

    println("Массив: $arr")

    println("Введите индекс:")

    val index = scanner.nextInt()

    println("Элемент: ${elementAtIndex(arr, index) ?: "Не
```

```
найден"}")
```

```
}
```

```
33
```

```
    scanner.nextLine()

    println("Введите строку:")

    val s = scanner.nextLine()

    println("Результат: ${removeSpaces(s)}")
```

```
}
```

```
34
```

```
    println("Введите N:")

    val n = scanner.nextInt()

    println("Сумма: ${sumNaturalNumbers(n)}")
```



```
}
```

35

```
scanner.nextLine()

println("Введите основную строку:")

val s1 = scanner.nextLine()

println("Введите подстроку:")

val s2 = scanner.nextLine()

println("Результат: ${containsSubstring(s1, s2)}")

}
```

36

```
println("Введите число:")

val n = scanner.nextInt()

multiplicationTable(n)

}
```

37

```
scanner.nextLine()

println("Введите строку:")

val s = scanner.nextLine()

println("Длина: ${getStringLength(s)}")

}
```

38

```
println("Введите размер массива:")

val size = scanner.nextInt()

val arr = List(size) { Random.nextInt(1, 101) }

println("Исходный массив: $arr")

println("Перевернутый: ${reverseArray(arr)}")

}
```

39

```
println("Введите размер массива:")

val size = scanner.nextInt()

val arr = List(size) { Random.nextInt(1, 101) }

println("Оригинал: $arr\nКопия: ${copyArray(arr)}")
}
```

40

```
scanner.nextLine()

println("Введите строку:")

val s = scanner.nextLine()

println("Количество гласных: ${countVowels(s)}")
}
```

41

```
println("Введите размер массива:")

val size = scanner.nextInt()

val arr = List(size) { Random.nextInt(1, 101) }

println("Массив: $arr")

println("Введите искомый элемент:")

val target = scanner.nextInt()

println("Индекс: ${findFirstIndex(arr, target)}")
}
```

42

```
println("Введите размер массива:")

val size = scanner.nextInt()

val arr = List(size) { Random.nextInt(1, 101) }

println("Массив: $arr")

println("Введите искомый элемент:")
```

```
    val target = scanner.nextInt()

    println("Последний индекс: ${findLastIndex(arr, target)}")
}
```