

Практическая работа 7

1. Написать консольный калькулятор

```
fun main() {
    var continueCalculating = true

    while (continueCalculating) {
        println("Введите выражение (например, 1 + 2, или 'exit' для выхода):")
        val input = readLine()?.lowercase() ?: ""

        if (input == "exit") {
            continueCalculating = false
            continue
        }

        try {
            val parts = input.split(" ")
            if (parts.size != 3) {
                println("Некорректный формат ввода.")
                continue
            }

            val num1 = parts[0].toDoubleOrNull() ?: throw IllegalArgumentException("Некорректное первое число")
            val operator = parts[1]
            val num2 = parts[2].toDoubleOrNull() ?: throw IllegalArgumentException("Некорректное второе число")

            val result = when (operator) {
                "+" -> num1 + num2
                "-" -> num1 - num2
                "*" -> num1 * num2
                "/" -> if (num2 != 0.0) num1 / num2 else throw ArithmeticException("Деление на ноль")
                else -> throw IllegalArgumentException("Неизвестная операция")
            }

            println("Результат: $result")

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

    println("Калькулятор завершил работу.")
}
```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\IntelliJ IDEA 2023.1.3\lib\idea
Введите выражение (например, 1 + 2, или 'exit' для выхода):
1 + 2
Результат: 3.0
Введите выражение (например, 1 + 2, или 'exit' для выхода):
9 - 7
Результат: 2.0
Введите выражение (например, 1 + 2, или 'exit' для выхода):
2 * 4
Результат: 8.0
Введите выражение (например, 1 + 2, или 'exit' для выхода):
6 / 2
Результат: 3.0
Введите выражение (например, 1 + 2, или 'exit' для выхода):
-2 + 1
Результат: -1.0
Введите выражение (например, 1 + 2, или 'exit' для выхода):
exit
Калькулятор завершил работу.

Process finished with exit code 0

```

2. Найти палиндром слова

```

fun isPalindrome(word: String): Boolean {
    val cleanWord = word.lowercase().replace("\\s".toRegex(), "")
    return cleanWord == cleanWord.reversed()
}

```

```

fun main() {
    println("Введите слово:")
    val word = readLine() ?: ""
    if (isPalindrome(word)) {
        println("$word - это палиндром.")
    } else {
        println("$word - это не палиндром.")
    }
}

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaa
Введите слово:
заказ
заказ - это палиндром.

Process finished with exit code 0

```

```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:
Введите слово:
жизнь
жизнь - это не палиндром.

Process finished with exit code 0
```

3. Напишите функцию, которая принимает количество побед, ничейных игр и поражений и возвращает количество очков, которая набрала команда.

Очки набираются так: • победы дают 3 очка; • ничьи дают 1 очко; • поражения дают 0 очков

```
fun calculatePoints(wins: Int, draws: Int, losses: Int): Int {
    return wins * 3 + draws * 1
}
```

```
fun main() {
    println("Введите количество побед:")
    val wins = readLine()?.toIntOrNull() ?: 0

    println("Введите количество ничьих:")
    val draws = readLine()?.toIntOrNull() ?: 0

    println("Введите количество поражений:")
    val losses = readLine()?.toIntOrNull() ?: 0

    val totalPoints = calculatePoints(wins, draws, losses)
    println("Всего очков: $totalPoints")
}
```

```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C
Введите количество побед:
4
Введите количество ничьих:
1
Введите количество поражений:
7
Всего очков: 13

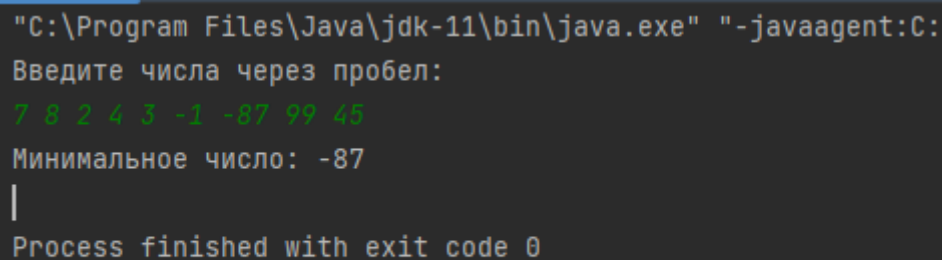
Process finished with exit code 0
```

Напишите программу, которая принимает на вход список чисел, и возвращает самое маленькое число из этого списка.

```
fun findMin(numbers: List<Int>): Int {
    return numbers.minOrNull() ?: throw IllegalArgumentException("Список пуст")
}
```

```
}
```

```
fun main() {  
    println("Введите числа через пробел:")  
    val input = readLine() ?: ""  
    val numbers = input.split(" ").mapNotNull { it.toIntOrNull() }  
    if(numbers.isEmpty()){  
        println("Список пуст")  
    } else {  
        val min = findMin(numbers)  
        println("Минимальное число: $min")  
    }  
}
```



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:  
Введите числа через пробел:  
7 8 2 4 3 -1 -87 99 45  
Минимальное число: -87  
|  
Process finished with exit code 0
```

Создайте программу, которая в качестве параметров принимает два числа и возвращает True, если эти числа равны, и False в противном случае.

```
fun areEqual(num1: Int, num2: Int): Boolean {  
    return num1 == num2  
}
```

```
fun main() {  
    println("Введите первое число:")  
    val num1 = readLine()?.toIntOrNull() ?: 0  
  
    println("Введите второе число:")  
    val num2 = readLine()?.toIntOrNull() ?: 0
```

```

if (areEqual(num1, num2)) {
    println("Числа равны.")
} else {
    println("Числа не равны.")
}
}

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-j
Введите первое число:
8
Введите второе число:
1
Числа не равны.

Process finished with exit code 0
|

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-jav
Введите первое число:
7
Введите второе число:
7
Числа равны.

Process finished with exit code 0
|

```

4. Карточная игра 21

```
import kotlin.random.Random
```

```

fun main() {
    val deck = generateDeck()
    var playerHand = mutableListOf<Int>()
    var dealerHand = mutableListOf<Int>()

```

```

    playerHand.add(deck.removeAt(0))
    dealerHand.add(deck.removeAt(0))
    playerHand.add(deck.removeAt(0))
    dealerHand.add(deck.removeAt(0))

```

```

    println("Ваши карты: ${playerHand.joinToString(" ")}, сумма: ${sumHand(playerHand)}")
    println("Карта дилера: ${dealerHand[0]}, сумма: ?")

```

```

while (sumHand(playerHand) < 21) {
    println("Взять еще карту? (yes/no)")
    val answer = readLine()?.lowercase() ?: ""
    if (answer == "yes") {
        playerHand.add(deck.removeAt(0))
        println("Ваши карты: ${playerHand.joinToString(" ")}, сумма: ${sumHand(playerHand)}")
    } else {
        break
    }
}

if (sumHand(playerHand) > 21) {
    println("Перебор! Вы проиграли.")
} else {
    //Логика дилера (упрощенная)
    while (sumHand(dealerHand) < 17) {
        dealerHand.add(deck.removeAt(0))
    }

    println("Карты дилера: ${dealerHand.joinToString(" ")}, сумма: ${sumHand(dealerHand)}")

    if (sumHand(dealerHand) > 21) {
        println("Дилер перебрал! Вы выиграли.")
    } else if (sumHand(playerHand) > sumHand(dealerHand)) {
        println("Вы выиграли!")
    } else if (sumHand(playerHand) == sumHand(dealerHand)) {
        println("Ничья!")
    } else {
        println("Вы проиграли.")
    }
}

fun generateDeck(): MutableList<Int> {
    val deck = mutableListOf<Int>()
    for (i in 1..13) {
        deck.add(i)
        deck.add(i)
        deck.add(i)
        deck.add(i)
    }
    deck.shuffle()
    return deck
}

fun sumHand(hand: List<Int>): Int {
    var sum = 0
    var aces = 0

```

```

for (card in hand) {
    sum += when (card) {
        1 -> 11
        11, 12, 13 -> 10
        else -> card
    }
    if (card == 1) aces++
}
while (sum > 21 && aces > 0) {
    sum -= 10
    aces--
}
return sum
}

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C
Ваши карты: 11 9, сумма: 19
Карта дилера: 5, сумма: ?
Взять еще карту? (yes/no)
no
Карты дилера: 5 8 9, сумма: 22
Дилер перебрал! Вы выиграли.

Process finished with exit code 0

```

```

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:
Ваши карты: 2 3, сумма: 5
Карта дилера: 4, сумма: ?
Взять еще карту? (yes/no)
yes
Ваши карты: 2 3 7, сумма: 12
Взять еще карту? (yes/no)
yes
Ваши карты: 2 3 7 9, сумма: 21
Карты дилера: 4 2 9 3, сумма: 18
Вы выиграли!

Process finished with exit code 0
|

```

```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\In
Ваши карты: 5 6, сумма: 11
Карта дилера: 6, сумма: ?
Взять еще карту? (yes/no)
yes
Ваши карты: 5 6 7, сумма: 18
Взять еще карту? (yes/no)
yes
Ваши карты: 5 6 7 10, сумма: 28
Перебор! Вы проиграли.

Process finished with exit code 0
|
```