

Мироненко Артём ИС231

Практическая работа 7

1. Написать консольный калькулятор

```
fun main() {
    println("Калькулятор")

    while (true) {
        try {
            print("Введите выражение (например, 5 + 7):\nДля выхода напишите слово 'выход'")
            val input = readLine()?.trim() ?: break

            if (input.lowercase() == "выход") break

            val parts = input.split(" ")
            if (parts.size != 3) {
                println("Неверный формат. Используйте: число операция число")
                continue
            }

            val num1 = parts[0].toDouble()
            val operator = parts[1]
            val num2 = parts[2].toDouble()

            val result = calculate(num1, operator, num2)
            println("Результат: $result")

        } catch (e: NumberFormatException) {
            println("Ошибка: введите корректные числа")
        } catch (e: Exception) {
            println("Произошла ошибка: ${e.message}")
        }
    }
}

fun calculate(num1: Double, operator: String, num2: Double): Double {
    return when (operator) {
        "+" -> num1 + num2
        "-" -> num1 - num2
        "*" -> num1 * num2
        "/" -> {
            if (num2 == 0.0) throw ArithmeticException("Деление на ноль!")
            num1 / num2
        }
        else -> throw IllegalArgumentException("Неподдерживаемая операция: $operator")
    }
}
```

```
}
```

```
"C:\Users\User\AppData\Local\Programs\Eclipse Adopter\bin>
Калькулятор
Введите выражение (например, 5 + 7):
Для выхода напишите слово 'выход' 5 + 5
Результат: 10.0
Введите выражение (например, 5 + 7):
Для выхода напишите слово 'выход' 4 - 2
Результат: 2.0
Введите выражение (например, 5 + 7):
Для выхода напишите слово 'выход' 6 * 6
Результат: 36.0
Введите выражение (например, 5 + 7):
Для выхода напишите слово 'выход' 12 / 2
Произошла ошибка: Неподдерживаемая операция: |
Введите выражение (например, 5 + 7):
Для выхода напишите слово 'выход' 12 / 2
Результат: 6.0
Введите выражение (например, 5 + 7):
Для выхода напишите слово 'выход' выход

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:\Users\User\AppData\Local\Programs  
Введите слово:  
азмеремза  
азмеремза - это палиндром.  
  
Process finished with exit code 0
```

```
"C:\Users\User\AppData\Local\Programs  
Введите слово:  
азмер  
азмер - 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:\Users\User\AppData\Local\Programs  
Введите количество побед:  
3  
Введите количество ничьих:  
5  
Введите количество поражений:  
2  
Всего очков: 14  
  
Process finished with exit code 0
```

4. 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:\Users\User\AppData\Local\Programs
Введите числа через пробел:
4 65 12 7 8 34 5 89 0 -43
Минимальное число: -43

Process finished with exit code 0
```

5. 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
    val result = areEqual(num1, num2)
    if (areEqual(num1, num2)) {
        println("Числа равны. ($result)")
    } else {
        println("Числа не равны. ($result)")
    }
}
```

```

Введите первое число:
3
Введите второе число:
4
Числа не равны. (false)

```

```

Введите первое число:
3
Введите второе число:
3
Числа равны. (true)

```

```

Введите первое число:
5
Введите второе число:
-5
Числа не равны. (false)

```

6. Карточная игра 21

```

data class Card(val rank: String, val suit: String, val value: Int)

class Deck {
    private val cards = mutableListOf<Card>()
    private val suits = listOf("♠", "♥", "♦", "♣")
    private val ranks = listOf("2", "3", "4", "5", "6", "7", "8",
    "9", "10", "J", "Q", "K", "A")

    init {
        for (suit in suits) {
            for (rank in ranks) {
                val value = when (rank) {
                    "A" -> 11
                    in listOf("K", "Q", "J") -> 10
                    else -> rank.toInt()
                }
                cards.add(Card(rank, suit, value))
            }
        }
        cards.shuffle()
    }

    fun drawCard(): Card = cards.removeFirst()
}

class Player(val name: String) {
    val hand = mutableListOf<Card>()

    fun getScore(): Int {
        var score = 0
        var aces = 0

        for (card in hand) {
            if (card.rank == "A") aces++
            score += card.value
        }

        while (score > 21 && aces > 0) {
            score -= 10
            aces--
        }

        return score
    }
}

```

```

    fun showHand(hideFirst: Boolean = false): String {
        if (hideFirst && hand.size > 0) {
            return "XX " + hand.subList(1,
hand.size).joinToString(" ") { "${it.rank}${it.suit}" }
        }
        return hand.joinToString(" ") { "${it.rank}${it.suit}" }
    }
}

fun main() {
    println("Добро пожаловать в игру!")

    while (true) {
        val deck = Deck()
        val player = Player("Игрок")
        val dealer = Player("Дилер")

        // Начальная раздача
        player.hand.add(deck.drawCard())
        dealer.hand.add(deck.drawCard())
        player.hand.add(deck.drawCard())
        dealer.hand.add(deck.drawCard())

        // Ход игрока
        while (true) {
            println("\nВаши карты: ${player.showHand()}")
            println("${player.getScore()} очков")
            println("Карты дилера: ${dealer.showHand(true)}")

            if (player.getScore() == 21) {
                println("Black Jack! Вы выиграли!")
                break
            } else if (player.getScore() > 21) {
                println("Перебор! Вы проиграли!")
                break
            }
        }

        print("Взять карту? (да/нет) : ")
        val answer = readLine()?.lowercase() ?: "нет"

        if (answer == "да") {
            player.hand.add(deck.drawCard())
        } else {
            break
        }
    }

    // Ход дилера
    if (player.getScore() <= 21) {
        println("\nХод дилера:")
        while (dealer.getScore() < 17) {
            dealer.hand.add(deck.drawCard())
        }
    }
}

```

```

        println("Дилер берет карту: ${dealer.showHand() }")
    }

    println("\nВаши карты: ${player.showHand() }
(${player.getScore() } очков")
        println("Карты дилера: ${dealer.showHand() }
(${dealer.getScore() } очков")

    when {
                dealer.getScore() > 21 -> println("Дилер перебрал!
Вы выиграли!")
                dealer.getScore() > player.getScore() ->
println("Дилер выиграл!")
                dealer.getScore() < player.getScore() ->
println("Вы выиграли!")
                else -> println("Ничья!")
    }
}

print("\nСыграть еще раз? (да/нет) : ")
if (readLine()?.lowercase() != "да") break
}

println("Спасибо за игру!")
}

```

Ваши карты: 9♦ 8♥ (17 очков)
 Карты дилера: XX 3♥
 Взять карту? (да/нет): нет

 Ход дилера:
 Дилер берет карту: 10♦ 3♥ 7♦

 Ваши карты: 9♦ 8♥ (17 очков)
 Карты дилера: 10♦ 3♥ 7♦ (20 очков)
 Дилер выиграл!

Ваши карты: 4♥ 10♣ (14 очков)
 Карты дилера: XX 2♥
 Взять карту? (да/нет): да

 Ваши карты: 4♥ 10♣ 5♠ (19 очков)
 Карты дилера: XX 2♥
 Взять карту? (да/нет): нет

 Ход дилера:
 Дилер берет карту: A♣ 2♥ 5♦

 Ваши карты: 4♥ 10♣ 5♠ (19 очков)
 Карты дилера: A♣ 2♥ 5♦ (18 очков)
 Вы выиграли!

Ваши карты: 2♦ 9♦ (11 очков)

Карты дилера: XX J♣

Взять карту? (да/нет): да

Ваши карты: 2♦ 9♦ 6♦ (17 очков)

Карты дилера: XX J♣

Взять карту? (да/нет): нет

Ваши карты: 3♣ J♣ (13 очков)

Карты дилера: XX 8♦

Взять карту? (да/нет): да

Ваши карты: 3♣ J♣ 10♥ (23 очков)

Карты дилера: XX 8♦

Перебор! Вы проиграли!

Ход дилера:

Дилер берет карту: 2♣ J♣ Q♣

Ваши карты: 2♦ 9♦ 6♦ (17 очков)

Карты дилера: 2♣ J♣ Q♣ (22 очков)

Дилер перебрал! Вы выиграли!

Добро пожаловать в игру!

Ваши карты: 4♥ 4♣ (8 очков)

Карты дилера: XX 8♠

Взять карту? (да/нет): да

Ваши карты: 4♥ 4♣ 9♥ (17 очков)

Карты дилера: XX 8♠

Взять карту? (да/нет): нет

Ход дилера:

Ваши карты: 4♥ 4♣ 9♥ (17 очков)

Карты дилера: 9♣ 8♣ (17 очков)

Ничья!

Сыграть еще раз? (да/нет): |