Практическая работа № 7.1

Выполнили: Андрухова и Загородняя.

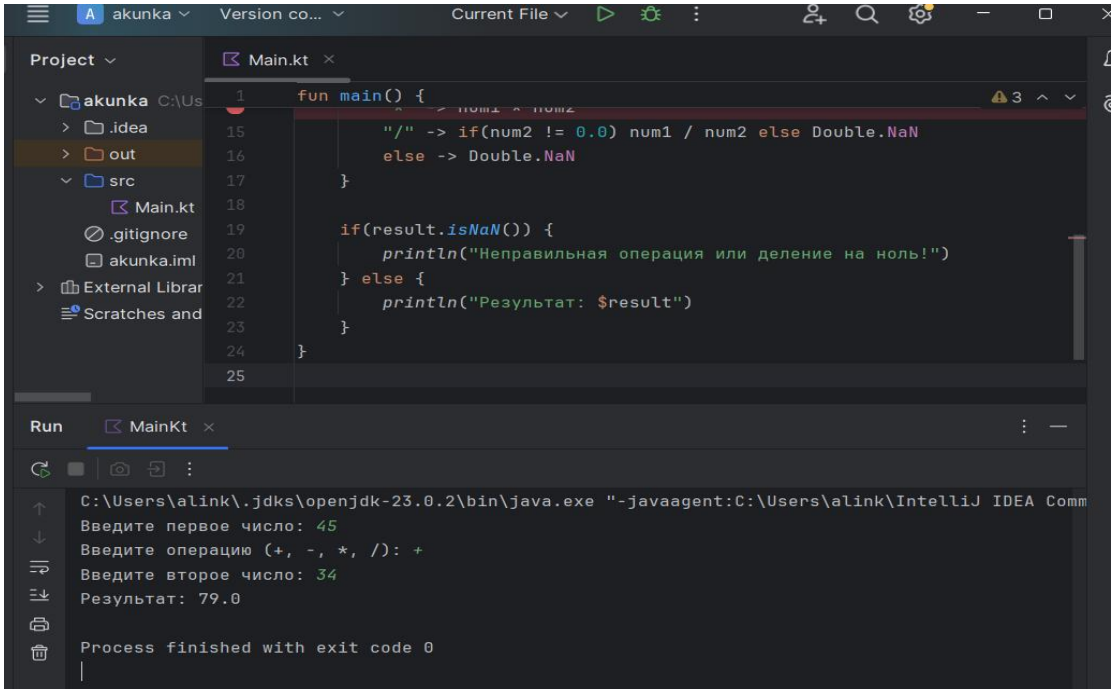
1.
```
fun main() {
    print("Введите первое число: ")
    val num1 = readLine()!!.toDouble()

    print("Введите операцию (+, -, *, /): ")
    val op = readLine()!!

    print("Введите второе число: ")
    val num2 = readLine()!!.toDouble()

    val result = when(op) {
        "+" -> num1 + num2
        "-" -> num1 - num2
        "*" -> num1 * num2
        "/" -> if(num2 != 0.0) num1 / num2 else Double.NaN
        else -> Double.NaN
    }

    if(result.isNaN()) {
        println("Неправильная операция или деление на ноль!")
    } else {
        println("Результат: $result")
    }
}
```



2.fun isPalindrome(word: String): Boolean {

return word.equals(word.reversed(), ignoreCase = true)

}

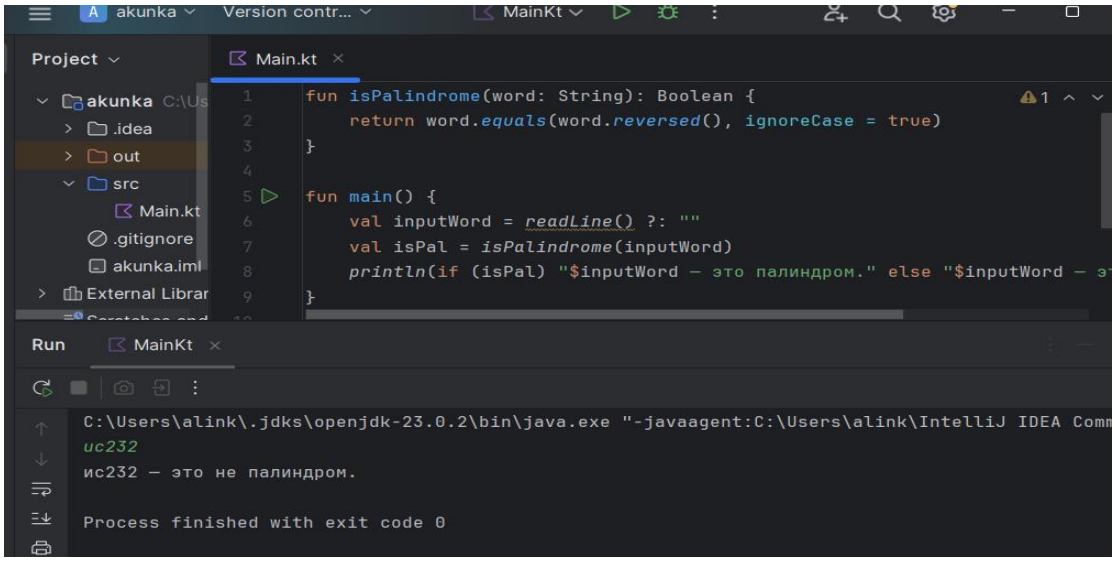
fun main() {

val inputWord = readLine() ?: ""

val isPal = isPalindrome(inputWord)

println(if (isPal) "$inputWord — это палиндром." else "$inputWord — это не палиндром.")

}



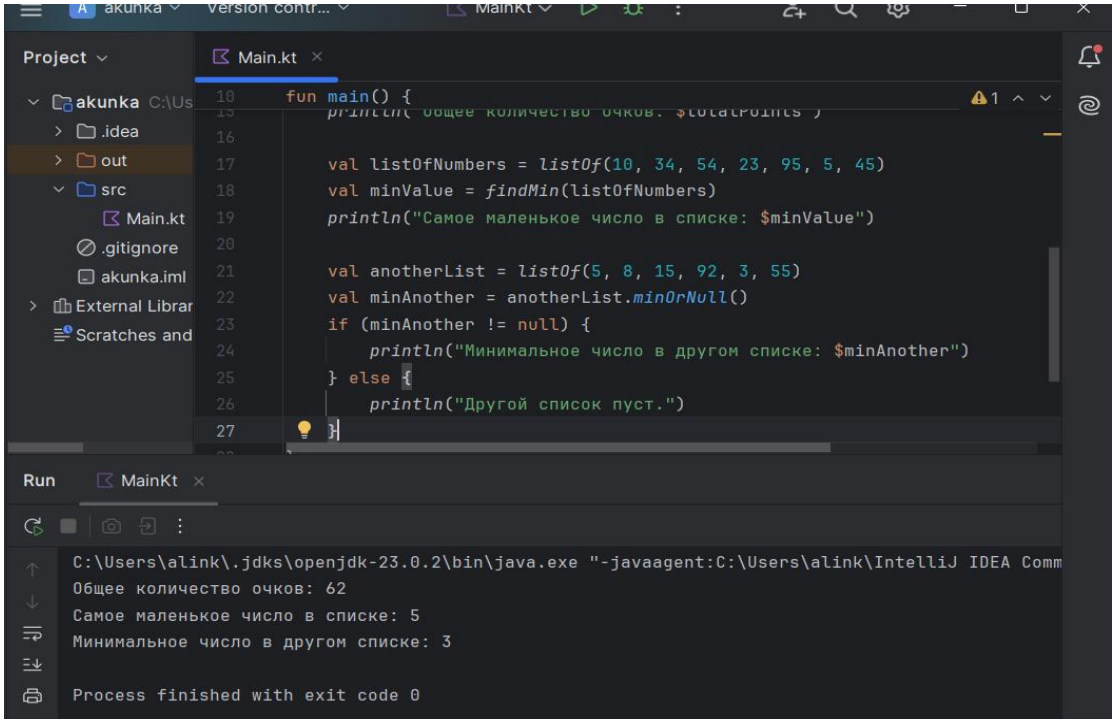
```
3.fun calculatePoints(wins: Int, draws: Int, losses: Int): Int {
    return wins * 3 + draws * 1 + losses * 0
}

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

fun main() {
    val wins = 18
    val draws = 8
    val losses = 6
    val totalPoints = calculatePoints(wins, draws, losses)
    println("Общее количество очков: $totalPoints")

    val listOfNumbers = listOf(10, 34, 54, 23, 95, 5, 45)
    val minValue = findMin(listOfNumbers)
    println("Самое маленькое число в списке: $minValue")

    val anotherList = listOf(5, 8, 15, 92, 3, 55)
    val minAnother = anotherList.minOrNull()
    if (minAnother != null) {
        println("Минимальное число в другом списке: $minAnother")
    } else {
        println("Другой список пуст.")
    }
}
```



4.import kotlin.random.Random

```
fun main() {
    val deck = createDeck()
    val playerHand = mutableListOf<Int>()
    val dealerHand = mutableListOf<Int>()

    dealCards(deck, playerHand, 2)
    dealCards(deck, dealerHand, 2)

    println("Твоя рука: $playerHand (сумма:
${getHandSum(playerHand)})")
    println("Рука дилера: $dealerHand (сумма:
${getHandSum(dealerHand)})")

    while (getHandSum(playerHand) <= 21 && getUserChoice()) {
        dealCard(deck, playerHand)
```

```
            println("Твоя новая рука: $playerHand (сумма:
${getHandSum(playerHand)})")
        }

        if (getHandSum(playerHand) > 21) {
            println("Перебор! Ты проиграл.")
            return
        }

        while (getHandSum(dealerHand) < 17) {
            dealCard(deck, dealerHand)
            println("Новая рука дилера: $dealerHand (сумма:
${getHandSum(dealerHand)})")
        }

        if (getHandSum(dealerHand) > 21) {
            println("Дилер перебрал! Ты выиграл.")
        } else if (getHandSum(dealerHand) >= getHandSum(playerHand)) {
            println("Дилер победил!")
        } else {
            println("Ты победил!")
        }
    }

    fun createDeck(): MutableList<Int> {
        val deck = mutableListOf<Int>()
        for (i in 1..13) {
            deck.add(i)
            deck.add(i)
```

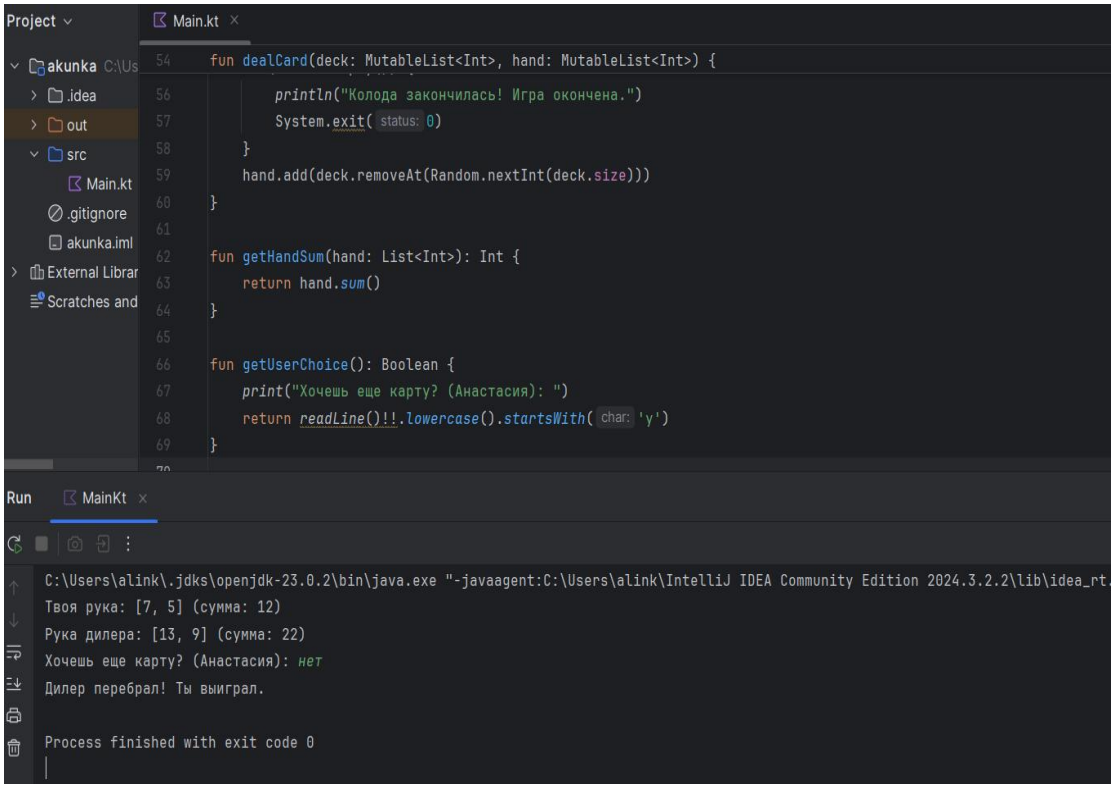
```
            deck.add(i)
            deck.add(i)
        }
        deck.shuffle()
        return deck
    }

    fun dealCards(deck: MutableList<Int>, hand: MutableList<Int>, count:
    Int) {
        repeat(count) { dealCard(deck, hand) }
    }

    fun dealCard(deck: MutableList<Int>, hand: MutableList<Int>) {
        if (deck.isEmpty()) {
            println("Колода закончилась! Игра окончена.")
            System.exit(0)
        }
        hand.add(deck.removeAt(Random.nextInt(deck.size)))
    }

    fun getHandSum(hand: List<Int>): Int {
        return hand.sum()
    }

    fun getUserChoice(): Boolean {
        print("Хочешь еще карту? (Анастасия): ")
        return readLine()!!.lowercase().startsWith('y')
    }
```


Project
akunka C:\Us
.idea
out
src
Main.kt
.gitignore
akunka.iml
External Librar
Scratches and
Main.kt
54 fun dealCard(deck: MutableList<Int>, hand: MutableList<Int>) {
println("Колода закончилась! Игра окончена.")
System.exit(status: 0)
}
hand.add(deck.removeAt(Random.nextInt(deck.size)))
}
fun getHandSum(hand: List<Int>): Int {
return hand.sum()
}
fun getUserChoice(): Boolean {
print("Хочешь еще карту? (Анастасия): ")
return readLine()!!.lowercase().startsWith(char: 'y')
}
Run
MainKt
C:\Users\alink\.jdks\openjdk-23.0.2\bin\java.exe "-javaagent:C:\Users\alink\IntelliJ IDEA Community Edition 2024.3.2.2\lib\idea_rt.
Твоя рука: [7, 5] (сумма: 12)
Рука дилера: [13, 9] (сумма: 22)
Хочешь еще карту? (Анастасия): нет
Дилер перебрал! Ты выиграл.
Process finished with exit code 0