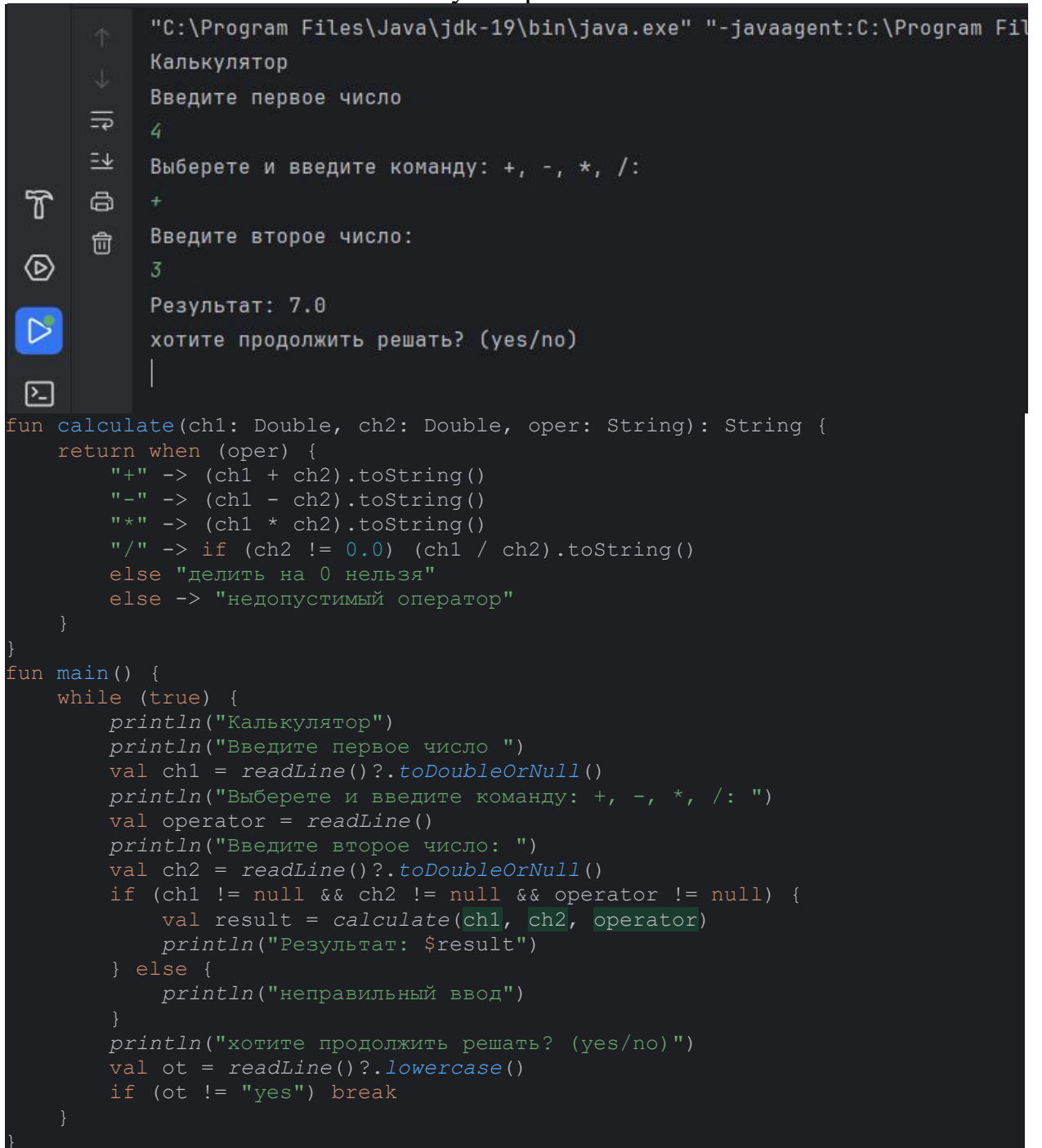


Практическая работа 7

1. Написать консольный калькулятор

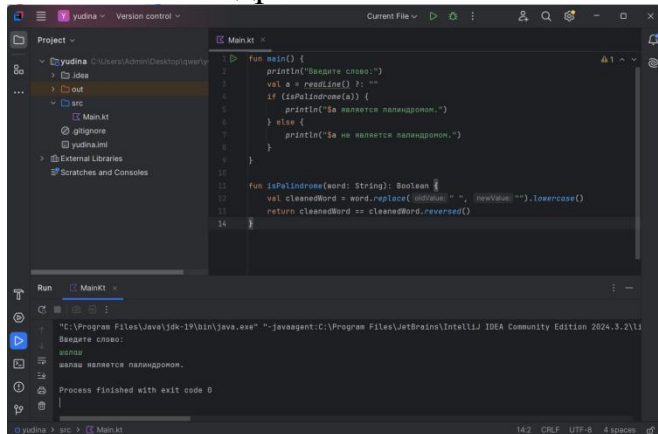


```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Fil
Калькулятор
Введите первое число
4
Выберете и введите команду: +, -, *, /:
+
Введите второе число:
3
Результат: 7.0
хотите продолжить решать? (yes/no)
|

fun calculate(ch1: Double, ch2: Double, oper: String): String {
    return when (oper) {
        "+" -> (ch1 + ch2).toString()
        "-" -> (ch1 - ch2).toString()
        "*" -> (ch1 * ch2).toString()
        "/" -> if (ch2 != 0.0) (ch1 / ch2).toString()
        else "делить на 0 нельзя"
        else -> "недопустимый оператор"
    }
}

fun main() {
    while (true) {
        println("Калькулятор")
        println("Введите первое число ")
        val ch1 = readLine()?.toDoubleOrNull()
        println("Выберете и введите команду: +, -, *, /: ")
        val operator = readLine()
        println("Введите второе число: ")
        val ch2 = readLine()?.toDoubleOrNull()
        if (ch1 != null && ch2 != null && operator != null) {
            val result = calculate(ch1, ch2, operator)
            println("Результат: $result")
        } else {
            println("неправильный ввод")
        }
        println("хотите продолжить решать? (yes/no)")
        val ot = readLine()?.lowercase()
        if (ot != "yes") break
    }
}
```

2. Найти палиндром слова



```
fun main() {
    println("Введите слово:")
    val a = readLine() ?: ""
    if (isPalindrome(a)) {
        println("$a является палиндромом.")
    } else {
        println("$a не является палиндромом.")
    }
}

fun isPalindrome(word: String): Boolean {
    val cleanedWord = word.replace(" ", "").lowercase()
    return cleanedWord == cleanedWord.reversed()
}
```

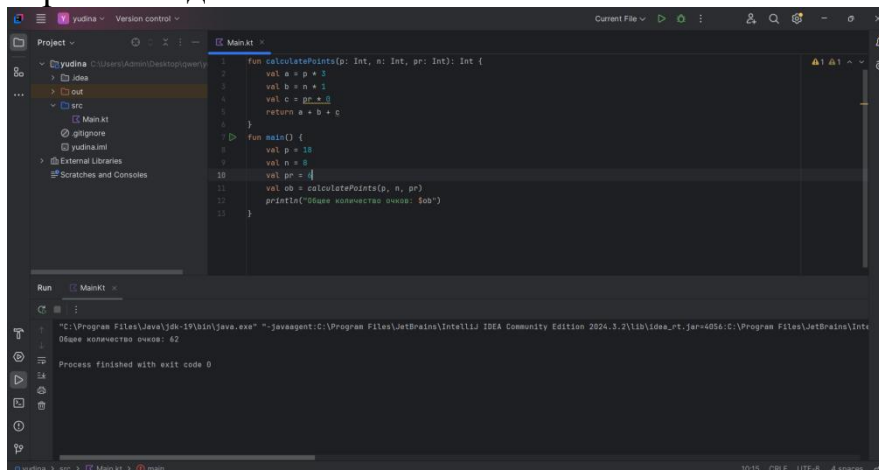
3. Напишите функцию, которая принимает количество побед, ничейных игр и поражений и возвращает количество очков, которые набрала команда.

Очки набираются так:

победы дают 3 очка;

ничьи дают 1 очко;

поражения дают 0 очков



```
fun calculatePoints(p: Int, n: Int, pr: Int): Int {
    val a = p * 3
    val b = n * 1
    val c = pr * 0
    return a + b + c
}

fun main() {
    val p = 18
    val n = 8
```

```

val pr = 6
val ob = calculatePoints(p, n, pr)
println("Общее количество очков: $ob")
}

```

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

The screenshot shows the IntelliJ IDEA IDE with a Kotlin file named `Main.kt`. The code defines a `main` function that prompts the user to enter a list of numbers separated by commas. It then splits the input, filters out null values, and finds the minimum number using the `findSmallestNumber` function. If the list is empty or contains invalid data, it prints an error message. The `findSmallestNumber` function uses `minOrNull` to find the minimum and throws an `IllegalArgumentException` if the list is empty.

```

1 fun main() {
2     println("Введите список чисел через запятую:")
3     val input = readLine() ?: ""
4     val numbers = input.split(",").map { it.trim().toIntOrNull() }.filterNotNull()
5     if (numbers.isNotEmpty()) {
6         val smallestNumber = findSmallestNumber(numbers)
7         println("Самое маленькое число в списке: $smallestNumber")
8     } else {
9         println("Список чисел пуст или введены некорректные данные")
10    }
11}
12
13 fun findSmallestNumber(numbers: List<Int>): Int {
14     return numbers.minOrNull() ?: throw IllegalArgumentException("Список не должен быть пустым")
15}

```

The Run window shows the execution output:

```

C:\Program Files\Java\jdk-19\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.2\l1
Введите список чисел через запятую:
2, 16, 1, 8, -5
Самое маленькое число в списке: -5
Process finished with exit code 0

```

The bottom part of the image shows the same code in a lighter theme for reference:

```

fun main() {
    println("Введите список чисел через запятую:")
    val input = readLine() ?: ""
    val numbers = input.split(",").map
{ it.trim().toIntOrNull() }.filterNotNull()
    if (numbers.isNotEmpty()) {
        val smallestNumber = findSmallestNumber(numbers)
        println("Самое маленькое число в списке: $smallestNumber")
    } else {
        println("Список чисел пуст или введены некорректные данные")
    }
}

fun findSmallestNumber(numbers: List<Int>): Int {
    return numbers.minOrNull() ?: throw IllegalArgumentException("Список
не должен быть пустым")
}

```

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

```
fun main() {  
    println("введите первое число:")  
    val a = readLine()!!.toInt()  
    println("введите второе число:")  
    val b = readLine()!!.toInt()  
    if (a==b)  
        println("true")  
    else  
        println("false")  
}
```

Run MainKt

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.2\li  
введите первое число:  
1  
введите второе число:  
2  
false  
Process finished with exit code 0
```

yudina > src > Main.kt > main 9:25 CRLF UTF-8 4 spaces

```
fun main() {  
    println("введите первое число:")  
    val a = readLine()!!.toInt()  
    println("введите второе число:")  
    val b = readLine()!!.toInt()  
    if (a==b)  
        println("true")  
    else  
        println("false")  
}
```

6. Карточная игра 21

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.2\li  
Добро пожаловать в игру 21!  
Ваши карты: [A, 3] (сумма:14)  
Карты дилера: [K, ?]  
Хотите взять карту? (да/нет)  
да  
Ваши карты: [A, 3, 9] (сумма:13)  
Карты дилера: [K, ?]  
Хотите взять карту? (да/нет)  
да  
Вы перебрали! Дилер выиграл!  
Process finished with exit code 0
```

yudina > src > Main.kt > playGame 42:52 CRLF UTF-8 4 spaces

```
import kotlin.random.Random  
val cardValues = mapOf(  
    "2" to 2, "3" to 3, "4" to 4, "5" to 5, "6" to 6,  
    "7" to 7, "8" to 8, "9" to 9, "10" to 10,  
    "J" to 10, "Q" to 10, "K" to 10, "A" to 11  
)  
fun main() {  
    println("Добро пожаловать в игру 21!")  
    playGame()  
}
```

```

}
fun playGame() {
    val playerCards = mutableListOf<String>()
    val dealerCards = mutableListOf<String>()
    playerCards.add(drawCard())
    playerCards.add(drawCard())
    dealerCards.add(drawCard())
    dealerCards.add(drawCard())
    var playerBusted = false
    while (true) {
        println("Ваши карты: $playerCards
(сумма:${calculateScore(playerCards)})")
        println("Карты дилера: [${dealerCards[0]}, ?]")
        if (calculateScore(playerCards) == 21) {
            println("Вы набрали 21! Вы выиграли!")
            return
        }
        println("Хотите взять карту? (да/нет)")
        val input = readLine()
        if (input.equals("да", ignoreCase = true)) {
            playerCards.add(drawCard())
            if (calculateScore(playerCards) > 21) {
                playerBusted = true
                break
            }
        } else {
            break
        }
    }
    if (!playerBusted) {
        while (calculateScore(dealerCards) < 17) {
            dealerCards.add(drawCard())
        }
        println("Карты дилера: $dealerCards
(сумма:${calculateScore(dealerCards)})")
        val playerScore = calculateScore(playerCards)
        val dealerScore = calculateScore(dealerCards)
        when {
            dealerScore > 21 -> println("Дилер перебрал! Вы выиграли!")
            playerScore > dealerScore -> println("Вы выиграли!")
            playerScore < dealerScore -> println("Дилер выиграл!")
            else -> println("Ничья!")
        }
    } else {
        println("Вы перебрали! Дилер выиграл!")
    }
}
}

fun drawCard(): String {
    val cards = cardValues.keys.toList()
    return cards[Random.nextInt(cards.size)]
}

fun calculateScore(cards: List<String>): Int {
    var score = cards.sumOf { cardValues[it] ?: 0 }
    var acesCount = cards.count { it == "A" }
    while (score > 21 && acesCount > 0) {
        score -= 10
        acesCount--
    }
    return score
}
}

```