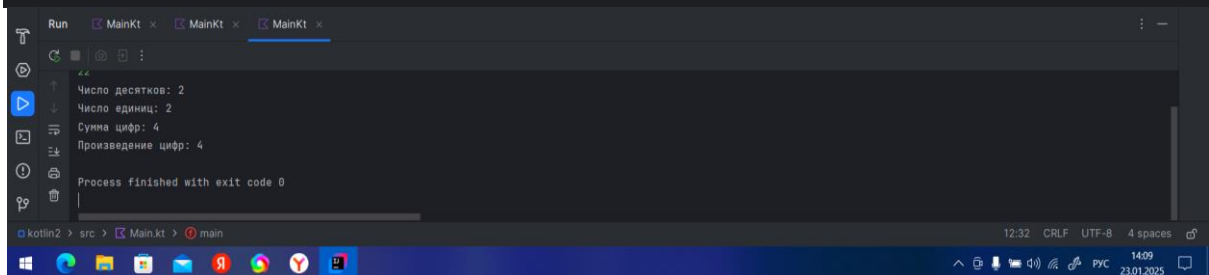


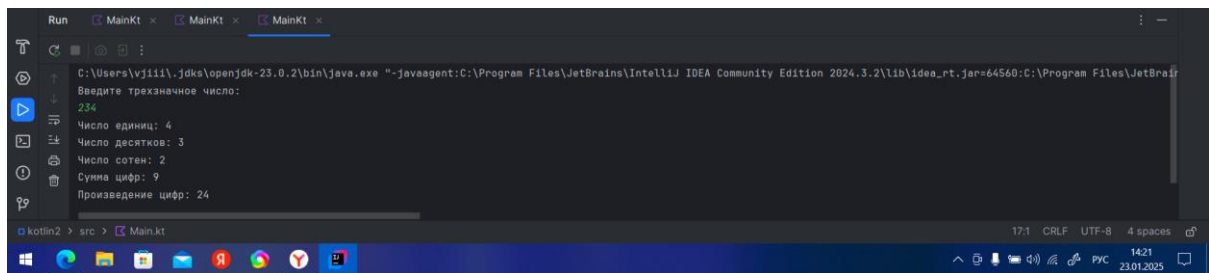
1. Дано двузначное число. Найти: а) число десятков в нем; б) число единиц в нем; в) сумму его цифр; г) произведение его цифр.

```
fun main() {  
    println("Введите двузначное число:")  
    val number = readLine()!!.toInt()  
  
    val tens = number / 10 // число десятков  
    val units = number % 10 // число единиц  
    val sum = tens + units // сумма цифр  
    val product = tens * units // произведение цифр  
  
    println("Число десятков: $tens")  
    println("Число единиц: $units")  
    println("Сумма цифр: $sum")  
    println("Произведение цифр: $product")  
}
```



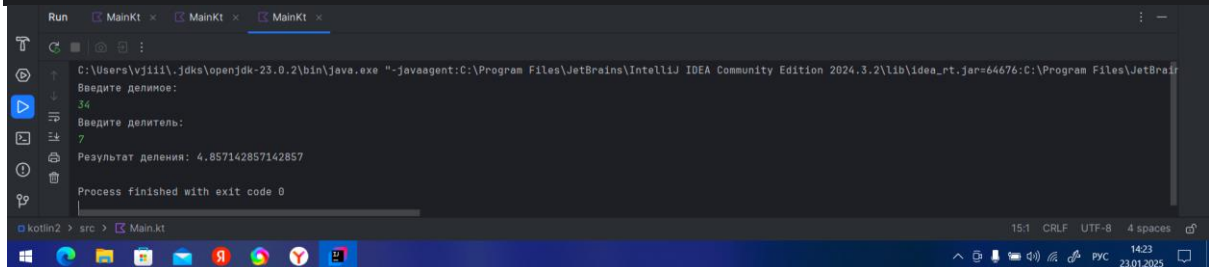
2. Дано трехзначное число. Найти: а) число единиц в нем; б) число десятков в нем; в) сумму его цифр; г) произведение его цифр

```
fun main() {  
    println("Введите трехзначное число:")  
    val number = readLine()!!.toInt()  
  
    val hundreds = number / 100 // число сотен  
    val tens = (number / 10) % 10 // число десятков  
    val units = number % 10 // число единиц  
    val sum = hundreds + tens + units // сумма цифр  
    val product = hundreds * tens * units // произведение цифр  
  
    println("Число единиц: $units")  
    println("Число десятков: $tens")  
    println("Число сотен: $hundreds")  
    println("Сумма цифр: $sum")  
    println("Произведение цифр: $product")  
}
```



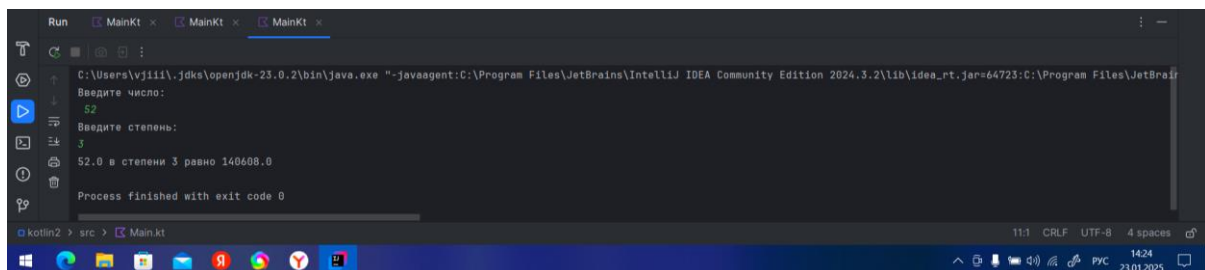
3. Напишите программу, которая делит одно число на другое.

```
fun main() {  
    println("Введите делимое:")  
    val dividend = readLine()!!.toDouble()  
  
    println("Введите делитель:")  
    val divisor = readLine()!!.toDouble()  
  
    if (divisor != 0.0) {  
        val result = dividend / divisor  
        println("Результат деления: $result")  
    } else {  
        println("Деление на ноль невозможно.")  
    }  
}
```



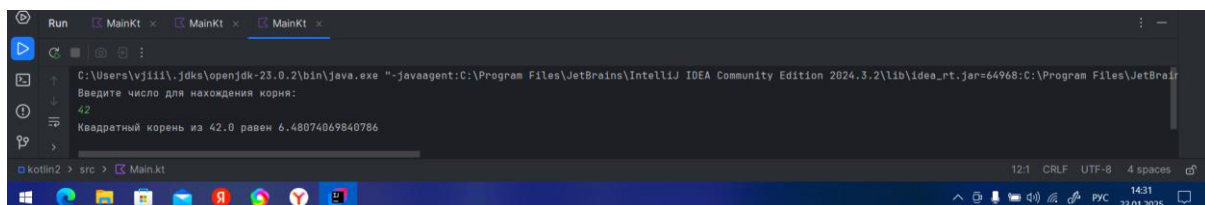
4. Разработайте программу, которая возведет число в определенную степень.

```
fun main() {  
    println("Введите число:")  
    val base = readLine()!!.toDouble()  
  
    println("Введите степень:")  
    val exponent = readLine()!!.toInt()  
  
    val result = Math.pow(base, exponent.toDouble())  
    println("$base в степени $exponent равно $result")  
}
```



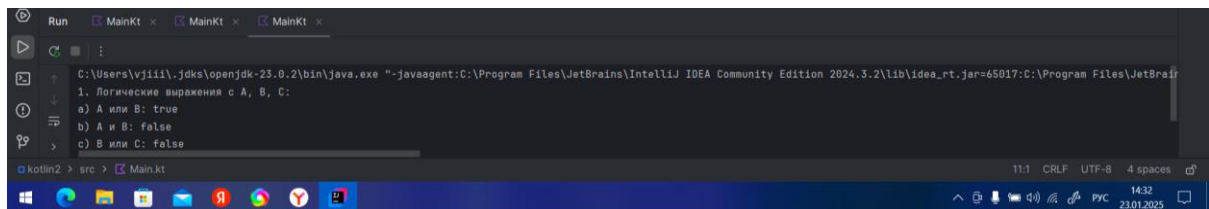
5. Разработайте программу, которая найдёт корень числа.

```
fun main() {  
    println("Введите число для нахождения корня:")  
    val number = readLine()!!.toDouble()  
  
    if (number >= 0) {  
        val root = Math.sqrt(number)  
        println("Квадратный корень из $number равен $root")  
    } else {  
        println("Корень из отрицательного числа не существует.")  
    }  
}
```



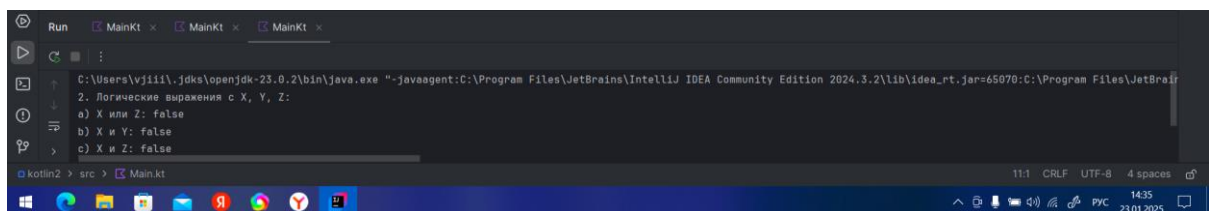
1. Вычислить значение логического выражения при следующих значениях логических величин A, B и C: A = Истина, B = Ложь, C = Ложь: а) A или B; б) A и B; в) B или C.

```
fun main() {  
    val A = true  
    val B = false  
    val C = false  
  
    println("1. Логические выражения с A, B, C:")  
    println("а) A или B: ${A || B}")  
    println("б) A и B: ${A && B}")  
    println("в) B или C: ${B || C}")  
}
```



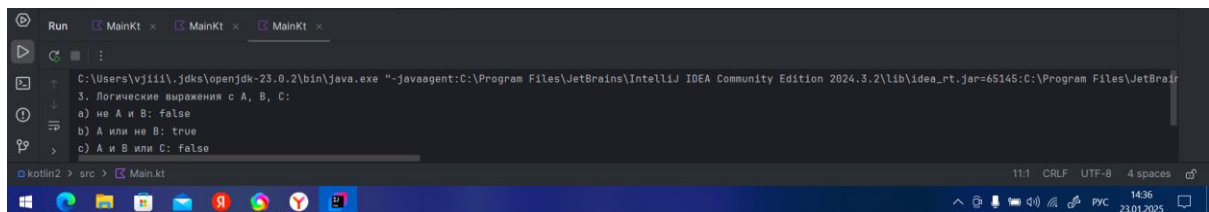
2. Вычислить значение логического выражения при следующих значениях логических величин X, Y и Z: X = Ложь, Y = Истина, Z = Ложь: а) X или Z; б) X и Y; в) X и Z

```
fun main() {  
    val X = false  
    val Y = true  
    val Z = false  
  
    println("2. Логические выражения с X, Y, Z:")  
    println("а) X или Z: ${X || Z}")  
    println("б) X и Y: ${X && Y}")  
    println("в) X и Z: ${X && Z}")  
}
```



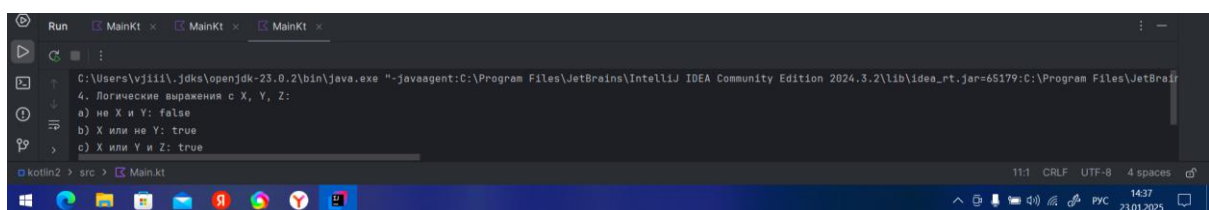
3. Вычислить значение логического выражения при следующих значениях логических величин A, B и C: A = Истина, B = Ложь, C = Ложь: а) не A и B; б) A или не B; в) A и B или C.

```
fun main() {  
    val A = true  
    val B = false  
    val C = false  
  
    println("3. Логические выражения с A, B, C:")  
    println("а) не A и B: ${!A && B}")  
    println("б) A или не B: ${A || !B}")  
    println("в) A и B или C: ${A && B || C}")  
}
```



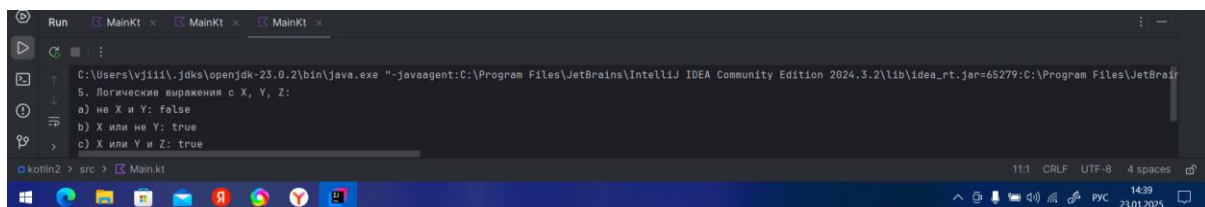
4. Вычислить значение логического выражения при следующих значениях логических величин X, Y и Z: X = Истина, Y = Истина, Z = Ложь: а) не X и Y; б) X или не Y; в) X или Y и Z

```
fun main() {  
    val X = true  
    val Y = true  
    val Z = false  
  
    println("4. Логические выражения с X, Y, Z:")  
    println("а) не X и Y: ${!X && Y}")  
    println("б) X или не Y: ${X || !Y}")  
    println("в) X или Y и Z: ${X || (Y && Z)}")  
}
```



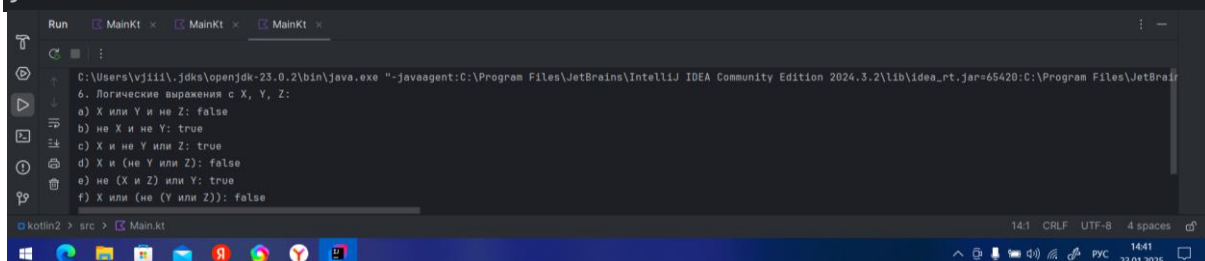
5. Вычислить значение логического выражения при следующих значениях логических величин X, Y и Z: X = Истина, Y = Истина, Z = Ложь: а) не X и Y; б) X или не Y; в) X или Y и Z

```
fun main() {  
    val X = true  
    val Y = true  
    val Z = false  
  
    println("5. Логические выражения с X, Y, Z:")  
    println("а) не X и Y: ${!X && Y}")  
    println("б) X или не Y: ${X || !Y}")  
    println("в) X или Y и Z: ${X || (Y && Z)}")  
}
```



6. Вычислить значение логического выражения при следующих значениях логических величин X, Y и Z: X = Ложь, Y = Ложь, Z = Истина: а) X или Y и не Z; г) X и не Y или Z; б) не X и не Y; д) X и (не Y или Z); в) не (X и Z) или Y; е) X или (не (Y или Z)).

```
fun main() {  
    val X = false  
    val Y = false  
    val Z = true  
  
    println("6. Логические выражения с X, Y, Z:")  
    println("а) X или Y и не Z: ${X || (Y && !Z)}")  
    println("б) не X и не Y: ${!X && !Y}")  
    println("в) X и не Y или Z: ${X && !Y || Z}")  
    println("г) X и (не Y или Z): ${X && (!Y || Z)}")  
    println("д) не (X и Z) или Y: ${!(X && Z) || Y}")  
    println("е) X или (не (Y или Z)): ${X || !(Y || Z)}")  
}
```



7. Вычислить значение логического выражения при следующих значениях логических величин A, B и C: A = Истина, B = Ложь, C = Ложь: а) A или не (A и B) или C; б) не A или A и (B или C); в) (A или B и не C) и C.

```
fun main() {  
    val A = true  
    val B = false  
    val C = false  
  
    println("7. Логические выражения с A, B, C:")  
    println("а) A или не (A и B) или C: ${A || !(A && B) || C}")  
    println("б) не A или A и (B или C): ${!A || (A && (B || C))}")  
    println("в) (A или B и не C) и C: ${(A || (B && !C)) && C}")  
}
```

