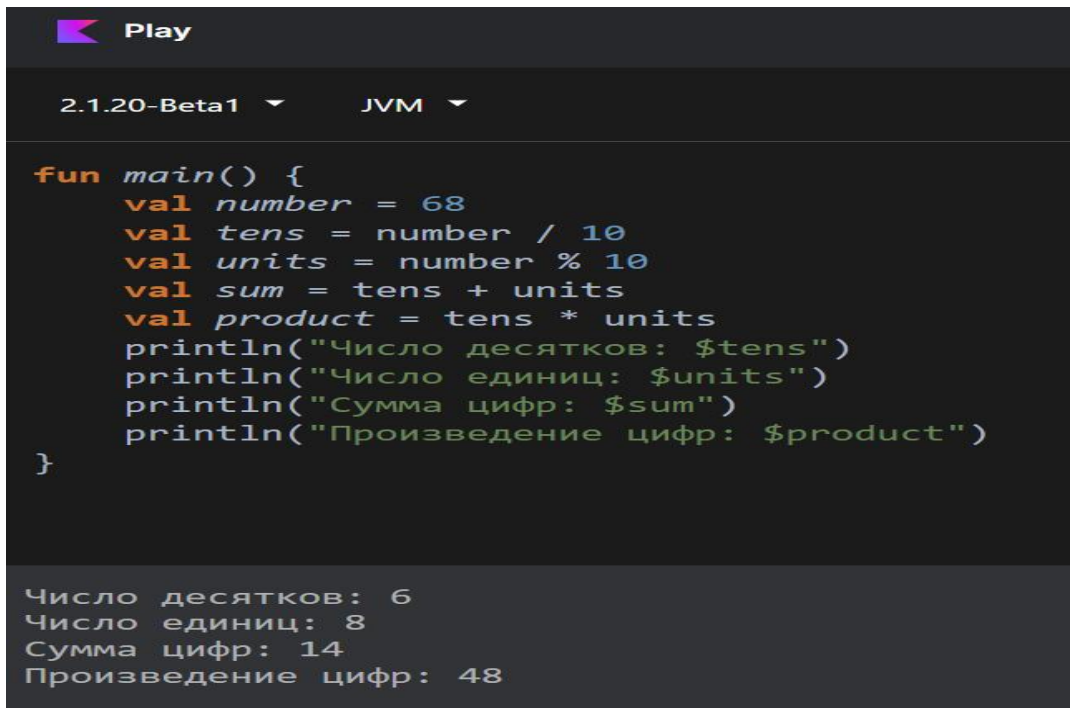


Практическая работа № 1

Выполнили: Андрухова и Загородняя

Операции с числами

```
1)fun main() {  
    val number = 68  
    val tens = number / 10  
    val units = number % 10  
    val sum = tens + units  
    val product = tens * units  
    println("Число десятков: $tens")  
    println("Число единиц: $units")  
    println("Сумма цифр: $sum")  
    println("Произведение цифр: $product")  
}
```



```
Play  
2.1.20-Beta1 JVM  
  
fun main() {  
    val number = 68  
    val tens = number / 10  
    val units = number % 10  
    val sum = tens + units  
    val product = tens * units  
    println("Число десятков: $tens")  
    println("Число единиц: $units")  
    println("Сумма цифр: $sum")  
    println("Произведение цифр: $product")  
}  
  
Число десятков: 6  
Число единиц: 8  
Сумма цифр: 14  
Произведение цифр: 48
```

```
2)fun main() {  
    val number = 453  
    val units = number % 10
```

```

val tens = (number / 10) % 10
val hundreds = number / 100
val sum = hundreds + tens + units
val product = hundreds * tens * units
println("Число единиц: $units")
println("Число десятков: $tens")
println("Сумма цифр: $sum")
println("Произведение цифр: $product")
}

```

The screenshot shows the Kotlin Playground interface. At the top, there's a 'Play' button and version information '2.1.20-Beta1' and 'JVM'. The code block contains the same Kotlin code as the first block. Below the code, the output is displayed:

```

Число единиц: 3
Число десятков: 5
Сумма цифр: 12
Произведение цифр: 60

```

```

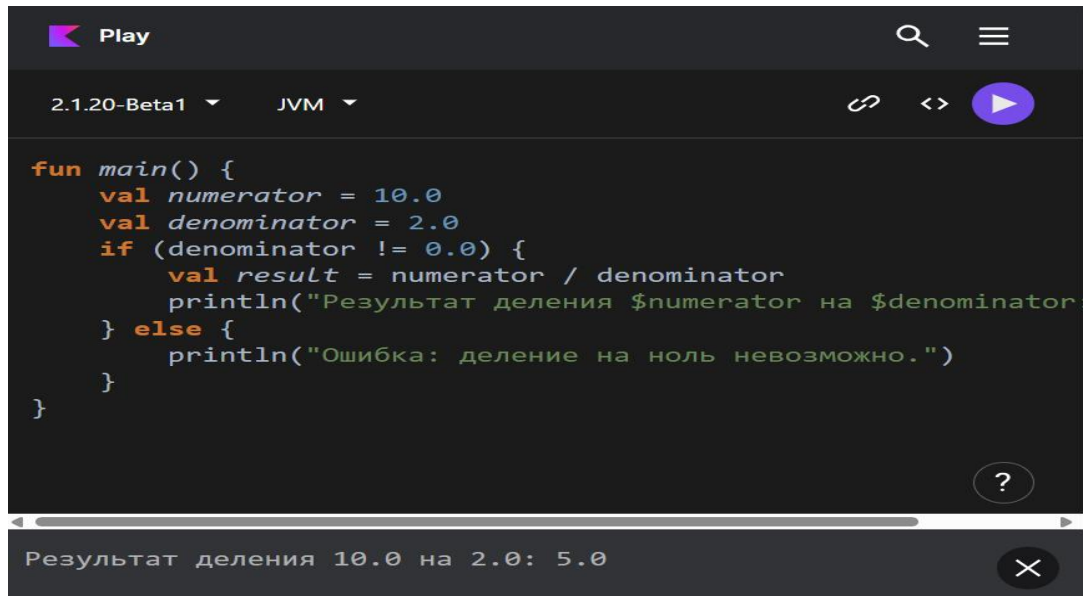
3) fun main() {
    val numerator = 10.0
    val denominator = 2.0
    if (denominator != 0.0) {
        val result = numerator / denominator
        println("Результат деления $numerator на $denominator: $result")
    } else {

```

```

        println("Ошибка: деление на ноль невозможно.")
    }
}

```



The screenshot shows the Kotlin Playground interface. At the top, it says "Play" with a search icon and a menu icon. Below that, the version "2.1.20-Beta1" and the target "JVM" are selected. The code editor contains the following Kotlin code:

```

fun main() {
    val numerator = 10.0
    val denominator = 2.0
    if (denominator != 0.0) {
        val result = numerator / denominator
        println("Результат деления $numerator на $denominator:")
    } else {
        println("Ошибка: деление на ноль невозможно.")
    }
}

```

At the bottom, the output console shows the result: "Результат деления 10.0 на 2.0: 5.0".

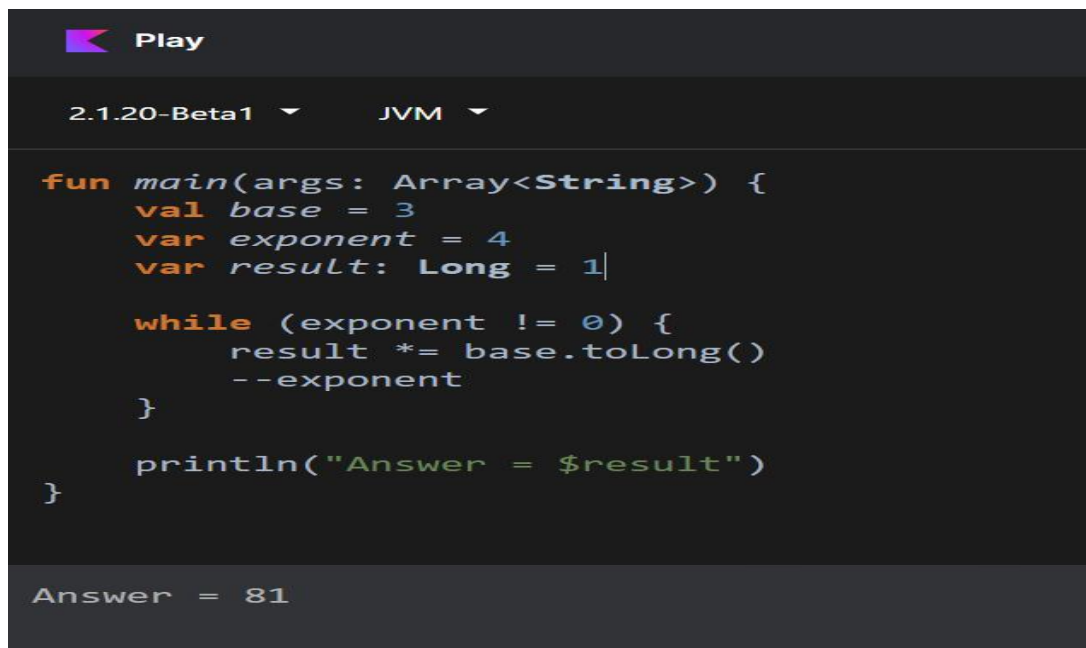
```

4) fun main(args: Array<String>) {
    val base = 3
    var exponent = 4
    var result: Long = 1

    while (exponent != 0) {
        result *= base.toLong()
        --exponent
    }

    println("Answer = $result")
}

```

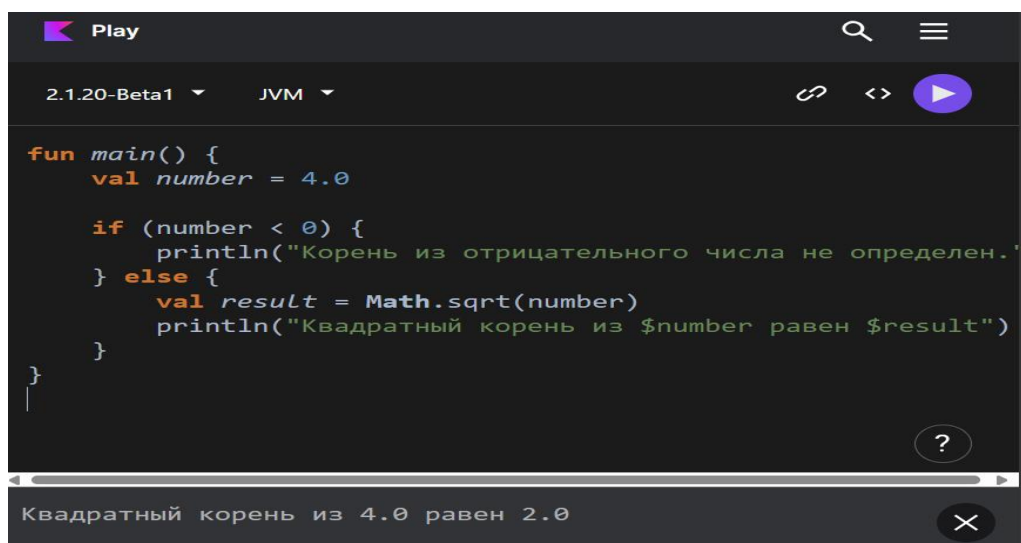


The screenshot shows the Kotlin Playground interface. At the top, there's a 'Play' button and version information '2.1.20-Beta1' and 'JVM'. The code area contains a `main` function that calculates 3 to the power of 4. The output area shows 'Answer = 81'.

```
fun main(args: Array<String>) {  
    val base = 3  
    var exponent = 4  
    var result: Long = 1  
  
    while (exponent != 0) {  
        result *= base.toLong()  
        --exponent  
    }  
  
    println("Answer = $result")  
}
```

Answer = 81

```
5)fun main() {  
    val number = 4.0  
  
    if (number < 0) {  
        println("Корень из отрицательного числа не определен.")  
    } else {  
        val result = Math.sqrt(number)  
        println("Квадратный корень из $number равен $result")  
    }  
}
```



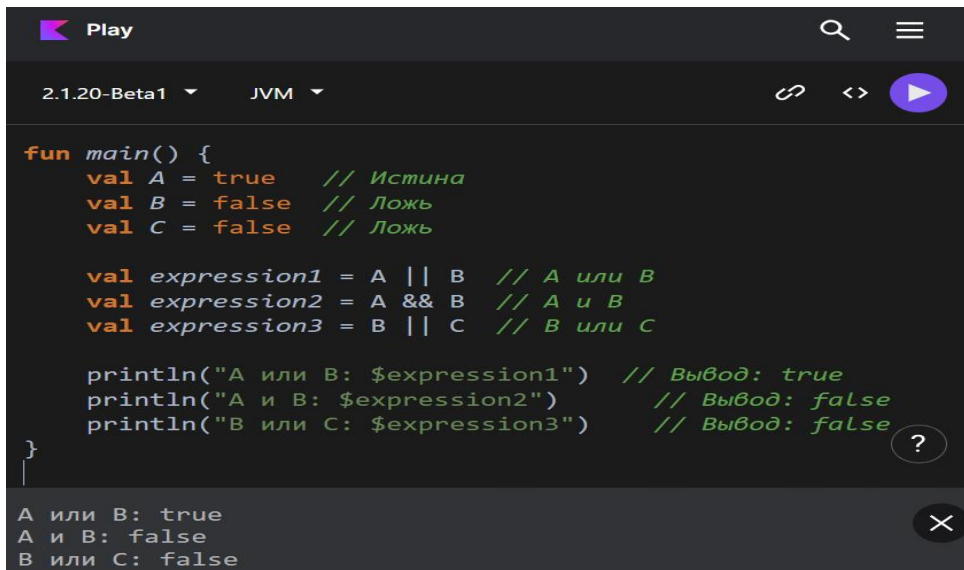
The screenshot shows the Kotlin Playground interface with a search icon and a menu icon in the top right. The code area contains a `main` function that checks if a number is negative and calculates its square root if it's positive. The output area shows 'Квадратный корень из 4.0 равен 2.0'.

```
fun main() {  
    val number = 4.0  
  
    if (number < 0) {  
        println("Корень из отрицательного числа не определен.")  
    } else {  
        val result = Math.sqrt(number)  
        println("Квадратный корень из $number равен $result")  
    }  
}
```

Квадратный корень из 4.0 равен 2.0

Вычисление логических выражений

```
1)fun main() {  
    val A = true // Истина  
    val B = false // Ложь  
    val C = false // Ложь  
    val expression1 = A || B // A или B  
    val expression2 = A && B // A и B  
    val expression3 = B || C // B или C  
    println("A или B: $expression1") // Вывод: true  
    println("A и B: $expression2") // Вывод: false  
    println("B или C: $expression3") // Вывод: false  
}
```



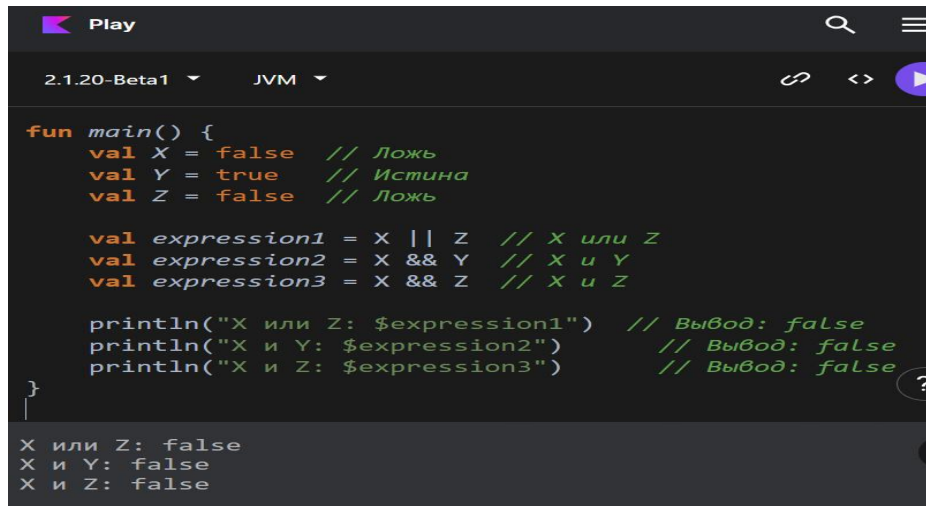
```
Play  
2.1.20-Beta1 JVM  
fun main() {  
    val A = true // Истина  
    val B = false // Ложь  
    val C = false // Ложь  
    val expression1 = A || B // A или B  
    val expression2 = A && B // A и B  
    val expression3 = B || C // B или C  
    println("A или B: $expression1") // Вывод: true  
    println("A и B: $expression2") // Вывод: false  
    println("B или C: $expression3") // Вывод: false  
}  
A или B: true  
A и B: false  
B или C: false
```

```
2)fun main() {  
    val X = false // Ложь  
    val Y = true // Истина  
    val Z = false // Ложь  
    val expression1 = X || Z // X или Z  
    val expression2 = X && Y // X и Y  
    val expression3 = X && Z // X и Z  
    println("X или Z: $expression1") // Вывод: false
```

```

println("X и Y: $expression2")    // Вывод: false
println("X и Z: $expression3")    // Вывод: false
}

```



```

Play
2.1.20-Beta1 JVM
fun main() {
    val X = false    // Ложь
    val Y = true     // Истина
    val Z = false    // Ложь

    val expression1 = X || Z    // X или Z
    val expression2 = X && Y    // X и Y
    val expression3 = X && Z    // X и Z

    println("X или Z: $expression1")    // Вывод: false
    println("X и Y: $expression2")      // Вывод: false
    println("X и Z: $expression3")      // Вывод: false
}
X или Z: false
X и Y: false
X и Z: false

```

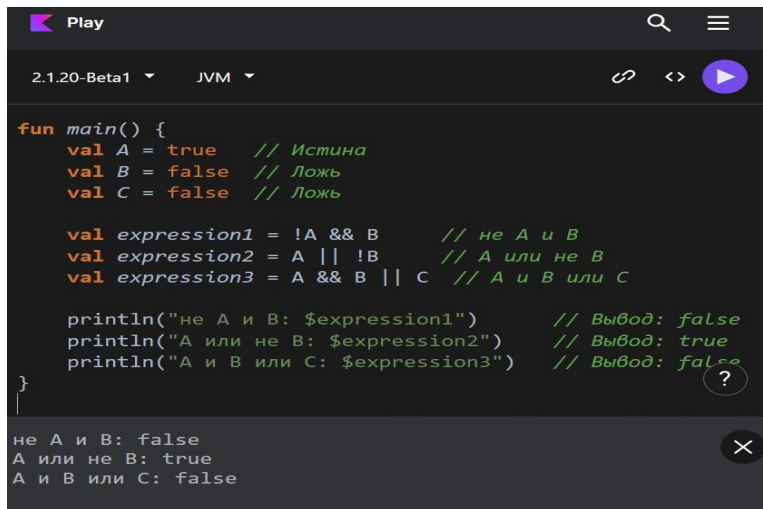
```

3)fun main() {
    val A = true    // Истина
    val B = false   // Ложь
    val C = false   // Ложь

    val expression1 = !A && B    // не A и B
    val expression2 = A || !B    // A или не B
    val expression3 = A && B || C // A и B или C

    println("не A и B: $expression1")    // Вывод: false
    println("A или не B: $expression2")    // Вывод: true
    println("A и B или C: $expression3")    // Вывод: false
}

```

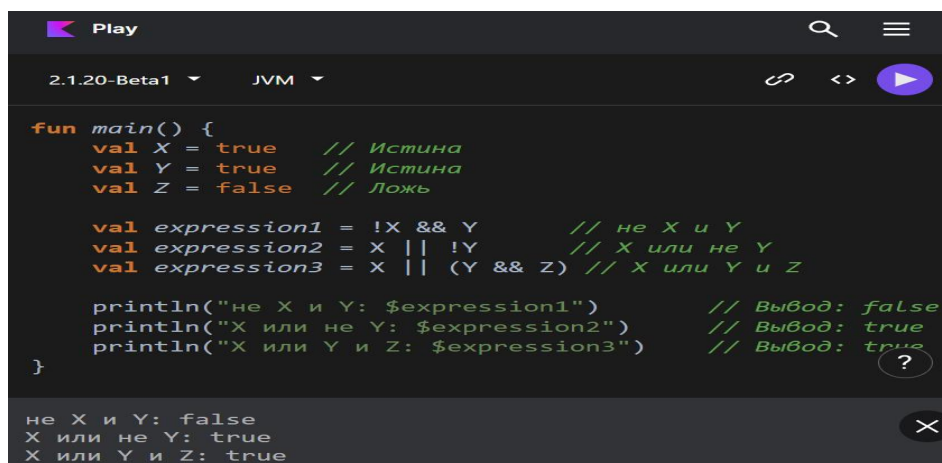


The screenshot shows the Kotlin Playground interface. At the top, there's a header with the 'Play' logo, a search icon, and a menu icon. Below the header, the version '2.1.20-Beta1' and the target 'JVM' are displayed. The main area contains a Kotlin code snippet for a `main()` function. The code defines three boolean variables: `A = true` (Истина), `B = false` (Ложь), and `C = false` (Ложь). It then calculates three expressions: `expression1 = !A && B` (не A и B), `expression2 = A || !B` (A или не B), and `expression3 = A && B || C` (A и B или C). The code prints the results of these expressions. The output at the bottom shows: 'не A и B: false', 'A или не B: true', and 'A и B или C: false'. A question mark icon is visible next to the third output line.

```
fun main() {  
    val A = true    // Истина  
    val B = false   // Ложь  
    val C = false   // Ложь  
  
    val expression1 = !A && B    // не A и B  
    val expression2 = A || !B   // A или не B  
    val expression3 = A && B || C // A и B или C  
  
    println("не A и B: $expression1")    // Вывод: false  
    println("A или не B: $expression2")  // Вывод: true  
    println("A и B или C: $expression3") // Вывод: false  
}
```

не A и B: false
A или не B: true
A и B или C: false

```
4)fun main() {  
    val X = true // Истина  
    val Y = true // Истина  
    val Z = false // Ложь  
    val expression1 = !X && Y    // не X и Y  
    val expression2 = X || !Y    // X или не Y  
    val expression3 = X || (Y && Z) // X или Y и Z  
    println("не X и Y: $expression1")    // Вывод: false  
    println("X или не Y: $expression2")    // Вывод: true  
    println("X или Y и Z: $expression3")    // Вывод: true  
}
```



The screenshot shows the Kotlin Playground interface. At the top, there's a header with the 'Play' logo, a search icon, and a menu icon. Below the header, the version '2.1.20-Beta1' and the target 'JVM' are displayed. The main area contains a Kotlin code snippet for a `main()` function. The code defines three boolean variables: `X = true` (Истина), `Y = true` (Истина), and `Z = false` (Ложь). It then calculates three expressions: `expression1 = !X && Y` (не X и Y), `expression2 = X || !Y` (X или не Y), and `expression3 = X || (Y && Z)` (X или Y и Z). The code prints the results of these expressions. The output at the bottom shows: 'не X и Y: false', 'X или не Y: true', and 'X или Y и Z: true'. A question mark icon is visible next to the third output line.

```
fun main() {  
    val X = true    // Истина  
    val Y = true    // Истина  
    val Z = false   // Ложь  
  
    val expression1 = !X && Y    // не X и Y  
    val expression2 = X || !Y    // X или не Y  
    val expression3 = X || (Y && Z) // X или Y и Z  
  
    println("не X и Y: $expression1")    // Вывод: false  
    println("X или не Y: $expression2")    // Вывод: true  
    println("X или Y и Z: $expression3")    // Вывод: true  
}
```

не X и Y: false
X или не Y: true
X или Y и Z: true

```
5)fun main() {
```

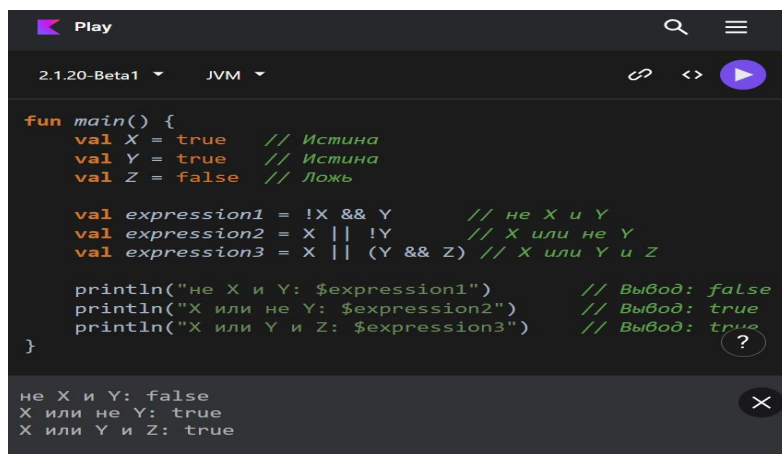
```

val X = true // Истина
val Y = true // Истина
val Z = false // Ложь

val expression1 = !X && Y // не X и Y
val expression2 = X || !Y // X или не Y
val expression3 = X || (Y && Z) // X или Y и Z

println("не X и Y: $expression1") // Вывод: false
println("X или не Y: $expression2") // Вывод: true
println("X или Y и Z: $expression3") // Вывод: true
}

```



```

6) fun main() {
    val X = false // Ложь
    val Y = false // Ложь
    val Z = true // Истина

    val expressionA = X || (Y && !Z) // X или Y и не Z
    val expressionG = (X && !Y) || Z // X и не Y или Z
    val expressionB = !X && !Y // не X и не Y
    val expressionD = X && (!Y || Z) // X и (не Y или Z)
    val expressionV = !(X && Z) || Y // не (X и Z) или Y
    val expressionE = X || !(Y || Z) // X или (не (Y или Z))
}

```



```

println("X или Y и не Z: $expressionA")    // Вывод: false
println("X и не Y или Z: $expressionG")    // Вывод: true
println("не X и не Y: $expressionB")       // Вывод: true
println("X и (не Y или Z): $expressionD")  // Вывод: false
println("не (X и Z) или Y: $expressionV")  // Вывод: true
println("X или (не (Y или Z)): $expressionE") // Вывод: false
}

```

The screenshot shows the Kotlin IDE interface. At the top, the Kotlin logo is visible. Below it, the version is 2.1.20-Beta1 and the target is JVM. A text box for 'Program arguments' is present. The main editor contains the following code:

```

val expressionA = X || (Y && !Z)           // X или Y и не Z
val expressionG = (X && !Y) || Z           // X и не Y или Z
val expressionB = !X && !Y                 // не X и не Y
val expressionD = X && (!Y || Z)           // X и (не Y или Z)
val expressionV = !(X && Z) || Y           // не (X и Z) или Y
val expressionE = X || !(Y || Z)          // X или (не (Y или Z))

println("X или Y и не Z: $expressionA")    // Вывод: false
println("X и не Y или Z: $expressionG")    // Вывод: true
println("не X и не Y: $expressionB")       // Вывод: true
println("X и (не Y или Z): $expressionD")  // Вывод: false
println("не (X и Z) или Y: $expressionV")  // Вывод: true
println("X или (не (Y или Z)): $expressionE") // Вывод: false
}

```

Below the code, the output of the program is displayed:

```

X или Y и не Z: false
X и не Y или Z: true
не X и не Y: true
X и (не Y или Z): false
не (X и Z) или Y: true
X или (не (Y или Z)): false

```

```

7) fun main() {
    val A = true // Истина
    val B = false // Ложь
    val C = false // Ложь

    val expressionA = A || !(A && B) || C    // A или не (A и B) или C
    val expressionB = !A || (A && (B || C))  // не A или A и (B или C)
    val expressionV = (A || (B && !C)) && C  // (A или B и не C) и C

    println("A или не (A и B) или C: $expressionA") // Вывод: true
    println("не A или A и (B или C): $expressionB") // Вывод: false
    println("(A или B и не C) и C: $expressionV")   // Вывод: false
}

```

```
fun main() {  
    val A = true    // Истина  
    val B = false   // Ложь  
    val C = false   // Ложь  
  
    val expressionA = A || !(A && B) || C           // A или не (A и B) или C  
    val expressionB = !A || (A && (B || C))         // не A или A и (B или C)  
    val expressionV = (A || (B && !C)) && C         // (A или B и не C) и C  
  
    println("A или не (A и B) или C: $expressionA") // Вывод: true  
    println("не A или A и (B или C): $expressionB") // Вывод: false  
    println("(A или B и не C) и C: $expressionV")   // Вывод: false  
}
```

```
A или не (A и B) или C: true  
не A или A и (B или C): false  
(A или B и не C) и C: false
```