

Практическая работа 2

Операции с числами

```
1) fun main() {  
    println("Введите число")  
    val a = readln()!!.toInt()  
    val b = a / 10  
    val c = a % 10  
    val d = b + c  
    val f = b * c  
    println("Число десятков: $b")  
    println("Число единиц: $c")  
    println("Сумма цифр: $d")  
    println("Произведение цифр: $f")  
}
```

Введите число

52

Число десятков: 5

Число единиц: 2

Сумма цифр: 7

Произведение цифр: 10

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

Введите число

345

Число сотен: 3

Число десятков: 4

Число единиц: 5

Сумма цифр: 12

Произведение цифр: 60

```
3) fun main() {  
    println("Введите число")  
    val a = readln()!!.toInt()  
    println("Введите число")  
    val b = readln()!!.toInt()  
    if (b != 0) {  
        val result = a / b  
        println("Результат деления $a на $b: $result")  
    } else {  
        println("Деление на ноль невозможно.")  
    }  
}
```

```
Введите число
4
Введите число
2
Результат деления 4 на 2: 2
```

```
4) fun main() {
    val base = 3.0
    val exponent = 4
    val result = Math.pow(base, exponent.toDouble())
    println("$base в степени $exponent равно $result")
}
3.0 в степени 4 равно 81.0
```

```
5) fun main() {
    val number = 42.0
    if (number >= 0) {
        val root = Math.sqrt(number)
        println("Корень числа $number равен $root")
    } else {
        println("Корень отрицательного числа не существует.")
    }
}
Корень числа 42.0 равен 6.48074069840786
```

Вычисление логических выражений

```
1) fun main() {
    val A = true
    val B = false
    val C = false
    println("A или B: ${A || B}")
    println("A и B: ${A && B}")
    println("B или C: ${B || C}")
}
```

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

```
2) fun main() {
    val X = false
    val Y = true
    val Z = false
    println("X или Z: ${X || Z}")
    println("X и Y: ${X && Y}")
    println("X и Z: ${X && Z}")
}
X или Z: false
X и Y: false
X и Z: false
```

```
3) fun main() {
    val A = true
    val B = false
    val C = false
    println("не A и B: ${!A && B}")
    println("A или не B: ${A || !B}")
}
```

```
println("A и B или C: ${A && B || C}")
}
```

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

```
4) fun main() {
    val X = true
    val Y = true
    val Z = false
    println("не X и Y: ${!X && Y}")
    println("X или не Y: ${X || !Y}")
    println("X или Y и Z: ${X || (Y && Z)}")
}
```

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

```
5) fun main() {
    val X = false
    val Y = false
    val Z = true
    println("X или Y и не Z: ${X || (Y && !Z)}")
    println("X и не Y или Z: ${X && !Y || Z}")
    println("не X и не Y: ${!X && !Y}")
    println("X и (не Y или Z): ${X && (!Y || Z)}")
    println("не (X и Z) или Y: ${!(X && Z) || Y}")
    println("X или (не (Y или Z)): ${X || !(Y || Z)}")
}
```

```
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
```

```
6) fun main() {
    val A = true
    val B = false
    val C = false
    println("A или не (A и B) или C: ${A || !(A && B) || C}")
    println("не A или A и (B или C): ${!A || (A && (B || C))}")
    println("(A или B и не C) и C: ${ (A || (B && !C)) && C}")
}
```

```
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
```

```
7) fun main() {
    val A = true
    val B = false
    val C = false
    val resultA = A || !(A && B) || C
    val resultB = !A || (A && (B || C))
    val resultC = (A || (B && !C)) && C
    println("Результат a)  $A \vee \neg(A \wedge B) \vee C$ : $resultA")
    println("Результат b)  $\neg A \vee (A \wedge (B \vee C))$ : $resultB")
}
```

```
println("Результат c)  $(A \vee (B \wedge \neg C)) \wedge C$ : $resultC")
}
```

Результат a) $A \vee \neg(A \wedge B) \vee C$: true
Результат b) $\neg A \vee (A \wedge (B \vee C))$: false
Результат c) $(A \vee (B \wedge \neg C)) \wedge C$: false