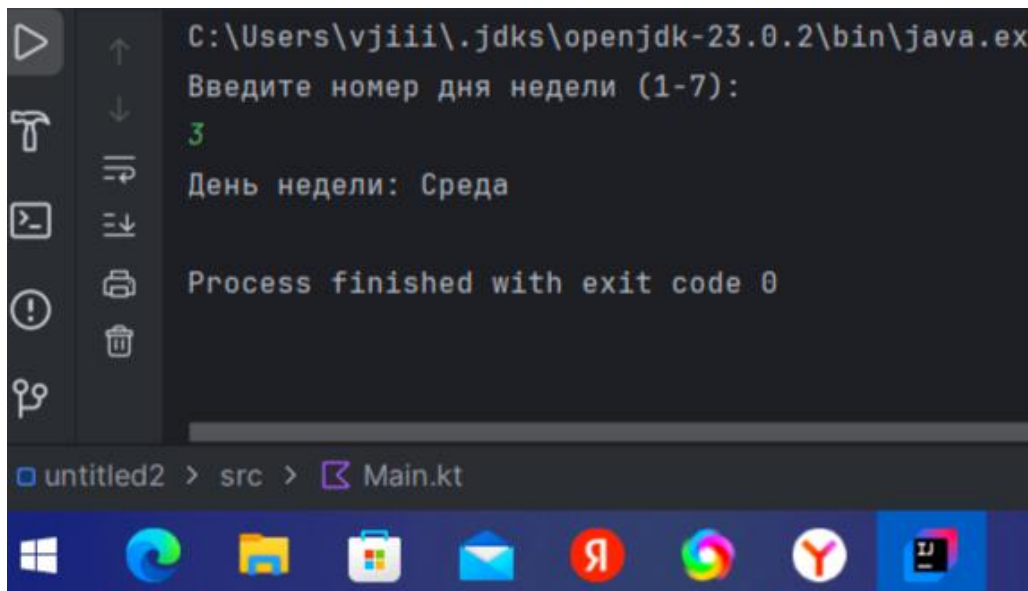


1. Определить день недели по номеру.

```
fun main() {  
    println("Введите номер дня недели (1-7):")  
    val dayNumber = readLine()?.toIntOrNull()  
    if (dayNumber != null) {  
        println("День недели: ${getDayOfWeek(dayNumber)}")  
    }  
}  
  
fun getDayOfWeek(dayNumber: Int): String {  
    return when (dayNumber) {  
        1 -> "Понедельник"  
        2 -> "Вторник"  
        3 -> "Среда"  
        4 -> "Четверг"  
        5 -> "Пятница"  
        6 -> "Суббота"  
        7 -> "Воскресенье"  
        else -> "Некорректный номер дня"  
    }  
}
```



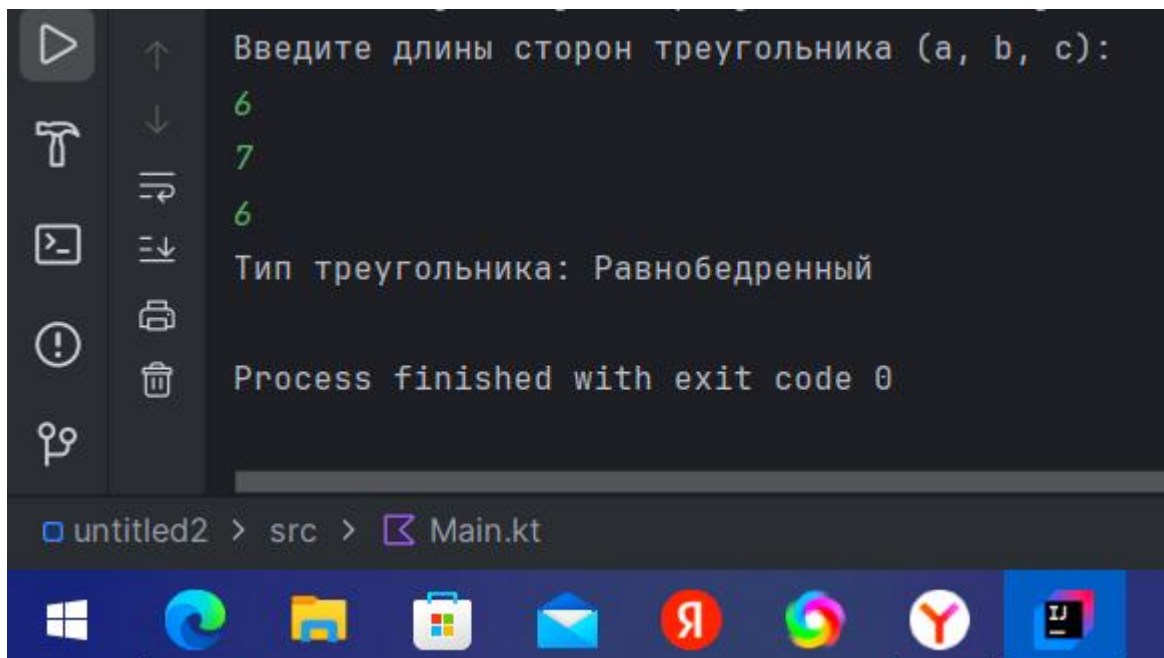
```
C:\Users\vjiii\jdk\openjdk-23.0.2\bin\java.exe  
Введите номер дня недели (1-7):  
3  
День недели: Среда  
  
Process finished with exit code 0
```

untitled2 > src > Main.kt

2. Определить тип треугольника по длинам сторон.

```
fun main() {
    println("Введите длины сторон треугольника (a, b, c):")
    val a = readLine()?.toDoubleOrNull()
    val b = readLine()?.toDoubleOrNull()
    val c = readLine()?.toDoubleOrNull()
    if (a != null && b != null && c != null) {
        println("Тип треугольника: ${triangleType(a, b, c)}")
    }
}

fun triangleType(a: Double, b: Double, c: Double): String {
    return when {
        a <= 0 || b <= 0 || c <= 0 -> "Некорректные длины сторон"
        a == b && b == c -> "Равносторонний"
        a == b || b == c || a == c -> "Равнобедренный"
        else -> "Разносторонний"
    }
}
```

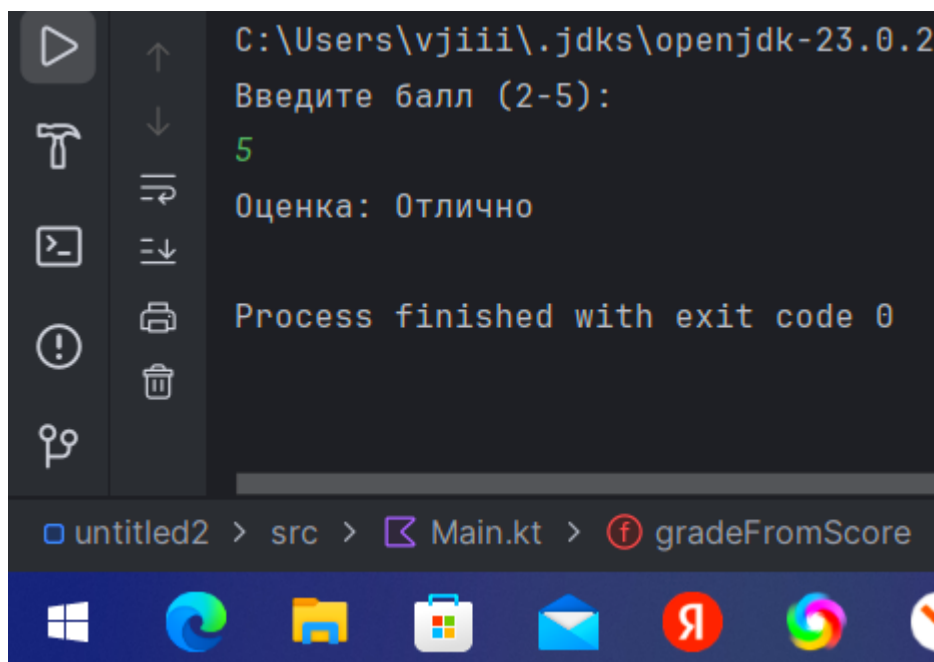


```
Введите длины сторон треугольника (a, b, c):
6
7
6
Тип треугольника: Равнобедренный
Process finished with exit code 0
```

untitled2 > src > Main.kt

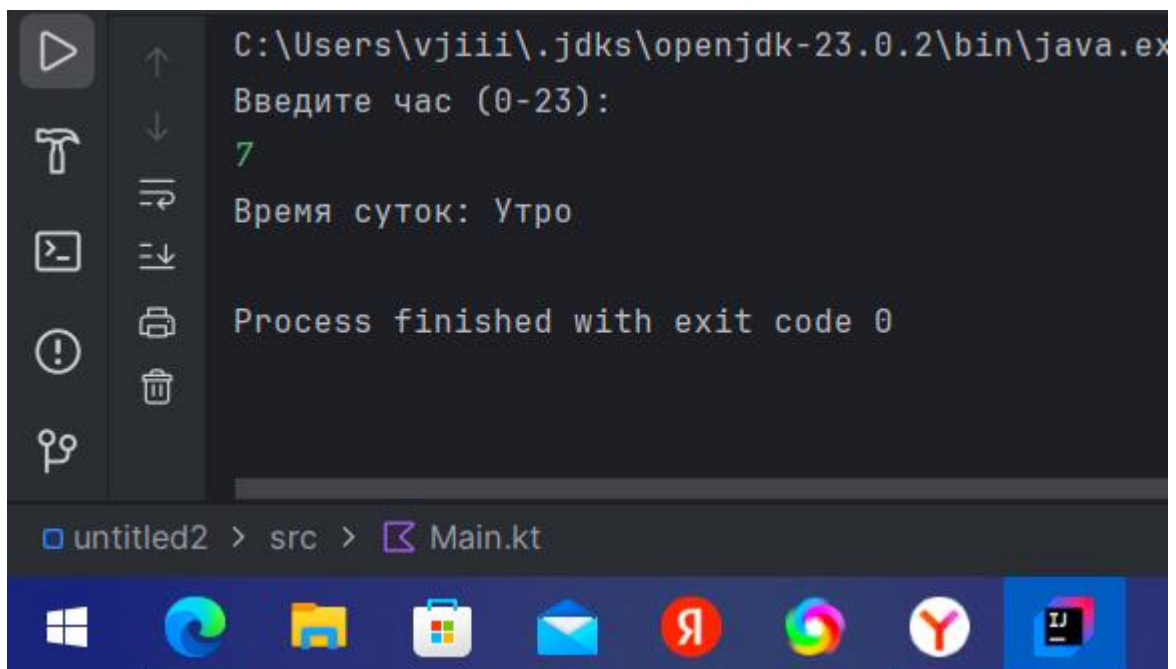
### 3. Вывод оценок по числовым значениям.

```
fun main() {  
    println("Введите балл (2-5):")  
    val score = readLine()?.toIntOrNull()  
    if (score != null) {  
        println("Оценка: ${gradeFromScore(score)}")  
    }  
}  
  
fun gradeFromScore(score: Int): String {  
    return when (score) {  
        in 5..5 -> "Отлично"  
        in 4..4 -> "Хорошо"  
        in 3..3 -> "Удовлетворительно"  
        in 2..2 -> "Неудовлетворительно"  
        else -> "Некорректный балл"  
    }  
}
```



#### 4. Определение времени суток.

```
fun main() {  
    println("Введите час (0-23):")  
    val hour = readLine()?.toIntOrNull()  
    if (hour != null) {  
        println("Время суток: ${timeOfDay(hour)}")  
    }  
}  
  
fun timeOfDay(hour: Int): String {  
    return when (hour) {  
        in 0..5 -> "Ночь"  
        in 6..11 -> "Утро"  
        in 12..17 -> "День"  
        in 18..23 -> "Вечер"  
        else -> "Некорректный час"  
    }  
}
```



```
C:\Users\vjiii\jdk\openjdk-23.0.2\bin\java.exe  
Введите час (0-23):  
7  
Время суток: Утро  
  
Process finished with exit code 0  
  
untitled2 > src > Main.kt
```

## 5. Определить знак числа

```
fun main() {  
    println("Введите число:")  
    val number = readLine()?.toIntOrNull()  
  
    when {  
        number == null -> println("Ошибка: введите корректное  
число.")  
        number > 0 -> println("Число положительное.")  
        number < 0 -> println("Число отрицательное.")  
        else -> println("Число равно нулю.")  
    }  
}
```

Введите число:

-4

Число отрицательное.

## 6. Угадай число

```
fun main() {  
    val secretNumber = (1..100).random()  
    var guess: Int? = null  
  
    println("Угадайте число от 1 до 100:")  
  
    while (guess != secretNumber) {  
        guess = readLine()?.toIntOrNull()  
        when {  
            guess == null -> println("Ошибка: введите корректное  
число.")  
            guess < secretNumber -> println("Слишком низко!")  
            guess > secretNumber -> println("Слишком высоко!")  
            else -> println("Поздравляем! Вы угадали число!")  
        }  
    }  
}
```

```

    }
}

Угадайте число от 1 до 100:
42
Слишком низко!
55
Слишком низко!
69
Слишком низко!
75
Поздравляем! Вы угадали число!

```

## 7. Определение длины строки

```

fun main() {
    println("Введите строку:")
    val input = readLine()

    when {
        input.isNullOrEmpty() -> println("Строка пуста.")
        else -> println("Длина строки: ${input.length}")
    }
}

Введите строку:
42 ,hfne[f
Длина строки: 10

```

## 8. Определить время приготовления по типу пищи

```

import java.util.*

fun main() {
    println("Введите тип пищи (мясо, рыба, овощи):")
    val foodType = readLine()

    val cookingTime = when
(foodType?.lowercase(Locale.getDefault())) {

```

```

        "мясо" -> "1-2 часа"
        "рыба" -> "20-30 минут"
        "овощи" -> "10-20 минут"
        else -> "Неизвестный тип пищи."
    }

    println("Время приготовления: $cookingTime")
}

```

```

Введите тип пищи (мясо, рыба, овощи):
мясо
Время приготовления: 1-2 часа

Process finished with exit code 0

```

## 9. Определение длины строки

Уже было....

10.Способы оплаты: Напишите программу, которая принимает способ оплаты (наличные, кредитная карта, PayPal) и выдает соответствующее сообщение.

```

import java.util.*
fun main() {
    println("Введите способ оплаты (наличные, кредитная карта, PayPal):")
    val paymentMethod = readLine()
    when (paymentMethod?.lowercase(Locale.getDefault())) {
        "наличные" -> println("Вы выбрали оплату наличными.")
        "кредитная карта" -> println("Вы выбрали оплату кредитной картой.")
        "paypal" -> println("Вы выбрали оплату через PayPal.")
        else -> println("Неизвестный способ оплаты.")
    }
}

```

```

Введите способ оплаты (наличные, кредитная карта, PayPal):
наличные
Вы выбрали оплату наличными.

```

11.Группа крови: вводится группа крови (А, В, АВ, О) и выводится, какие типы крови можно применять для переливания.

```

fun main() {
    println("Введите группу крови (A, B, AB, O):")
    val bloodType = readLine()
    val compatibleTypes = when (bloodType) {
        "A" -> "A, AB"
        "B" -> "B, AB"
        "AB" -> "AB"
        "O" -> "A, B, AB, O"
        else -> "Неизвестная группа крови."
    }
    println("Совместимые группы крови: $compatibleTypes")
}

```

```

Введите группу крови (A, B, AB, O):
A
Совместимые группы крови: A, AB

```

12.Национальности: Напишите программу, которая по странам (США, Россия, Япония и т.д.) выводит информацию о соответствующей национальности.

```

fun main() {
    println("Введите страну (США, Россия, Япония):")
    val country = readLine()
    val nationality = when (country) {
        "США" -> "Американец"
        "Россия" -> "Русский"
        "Япония" -> "Японец"
        else -> "Неизвестная страна."
    }
    println("Национальность: $nationality")
}

```

```

Введите страну (США, Россия, Япония):
Россия
Национальность: Русский

```

13.Коды ошибок: Программа принимает код ошибки (100, 200, 300) и выводит сообщение об ошибке (например, "Ошибка сети", "Ошибка сервера" и т.д.).



```
fun main() {  
    println("Введите код ошибки (100, 200, 300):")  
    val errorCode = readLine()?.toIntOrNull()  
    val errorMessage = when (errorCode) {  
        100 -> "Ошибка сети."  
        200 -> "Ошибка сервера."  
        300 -> "Ошибка клиента."  
        else -> "Неизвестный код ошибки."  
    }  
    println(errorMessage)  
}
```

```
Введите код ошибки (100, 200, 300):  
100  
Ошибка сети.
```



