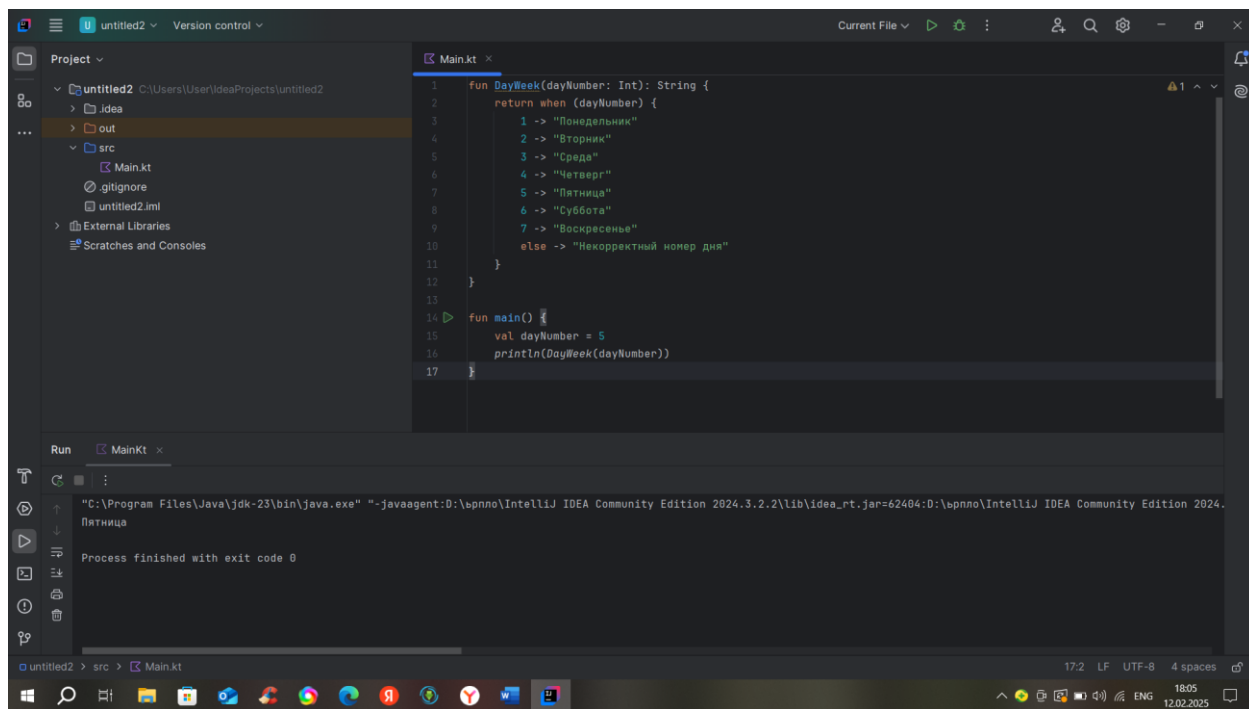


## Практическая №4

### 1. Определить день недели по номеру

```
fun DayWeek(dayNumber: Int): String {
    return when (dayNumber) {
        1 -> "Понедельник"
        2 -> "Вторник"
        3 -> "Среда"
        4 -> "Четверг"
        5 -> "Пятница"
        6 -> "Суббота"
        7 -> "Воскресенье"
        else -> "Некорректный номер дня"
    }
}

fun main() {
    val dayNumber = 5
    println(DayWeek(dayNumber))
}
```



### 2. Определить тип треугольника по длинам сторон

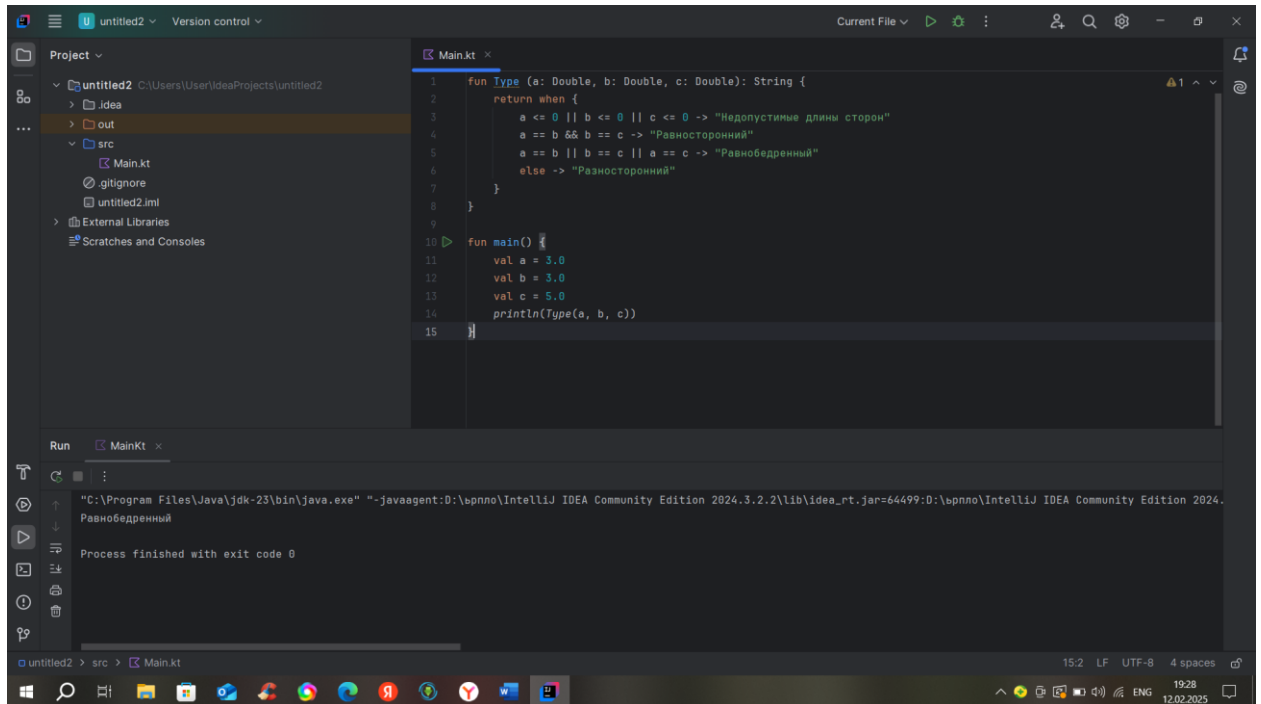
```
fun Type(a: Double, b: Double, c: Double): String {
    return when {
        a <= 0 || b <= 0 || c <= 0 -> "Недопустимые длины сторон"
        a == b && b == c -> "Равносторонний"
        a == b || b == c || a == c -> "Равнобедренный"
        else -> "Разносторонний"
    }
}

fun main() {
    val a = 3.0
    val b = 3.0
```

```

    val c = 5.0
    println(Type(a, b, c))
}

```



### 3) Вывод оценок по числовым значениям

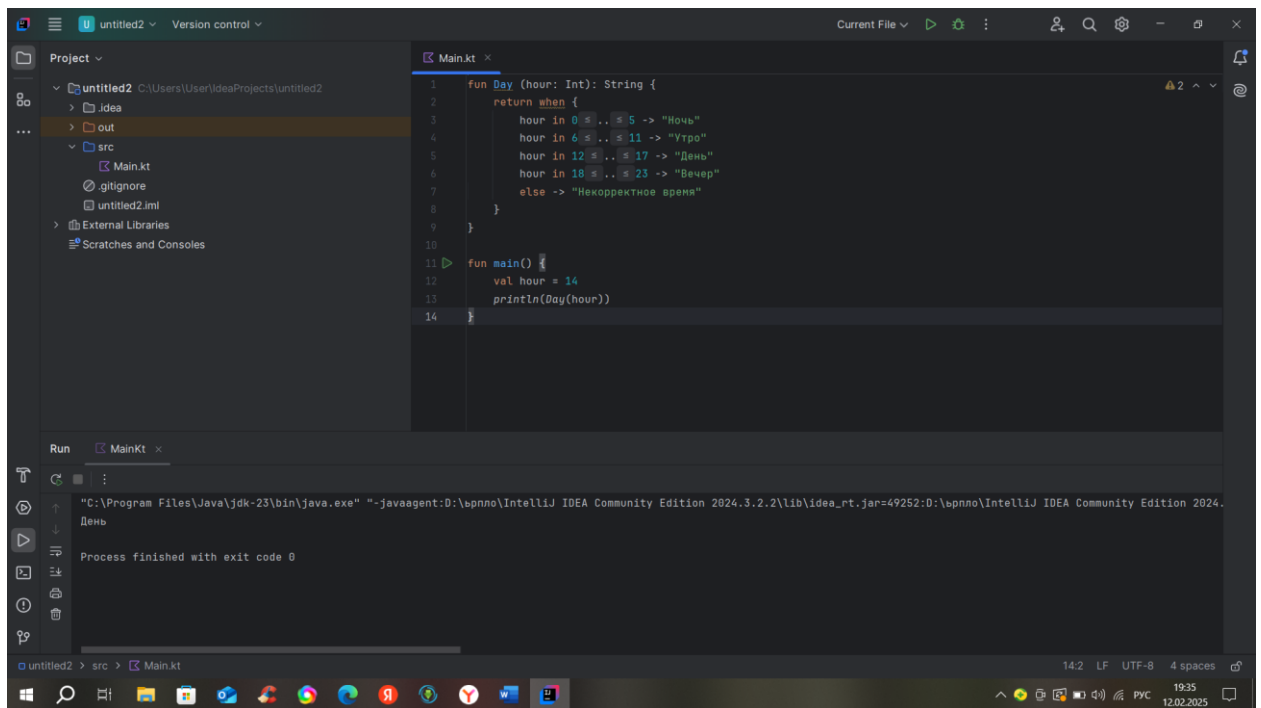
```

fun Grade(score: Int): String {
    return when (score) {
        in 90..100 -> "Оценка: A"
        in 80..89 -> "Оценка: B"
        in 70..79 -> "Оценка: C"
        in 60..69 -> "Оценка: D"
        in 0..59 -> "Оценка: F"
        else -> "Некорректное значение"
    }
}

fun main() {
    val score = 85
    println(Grade(score))
}

```

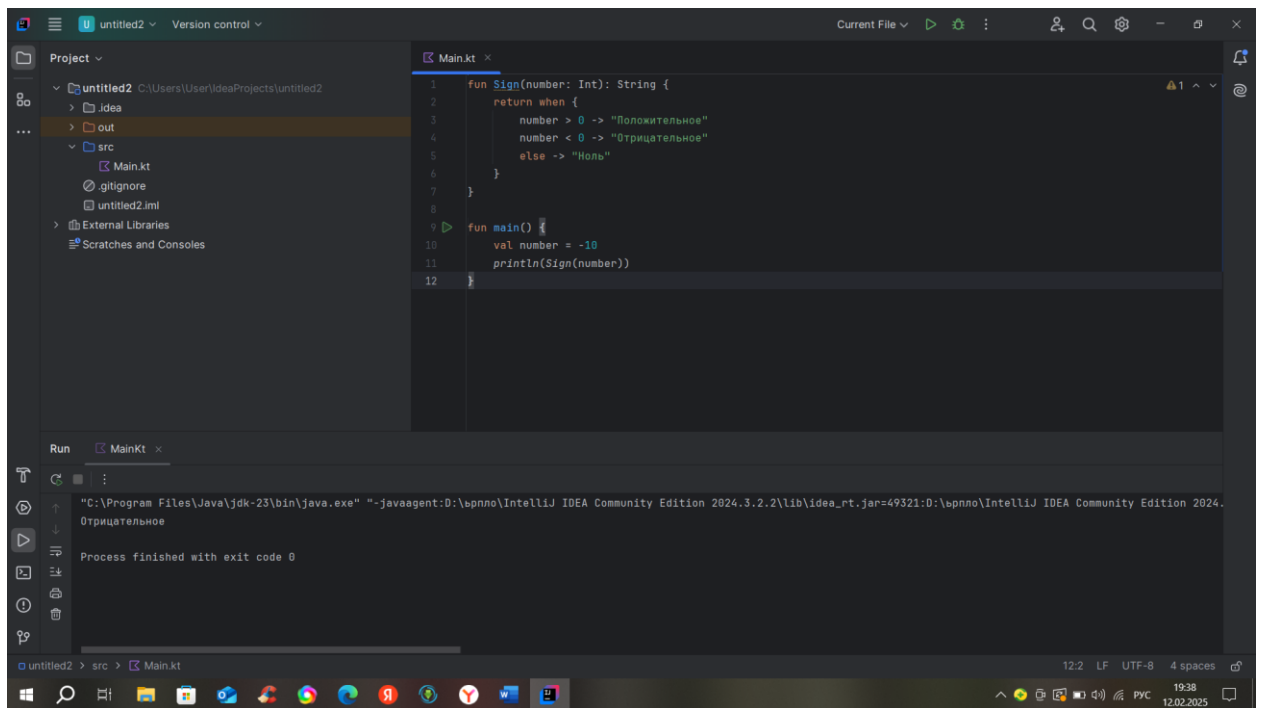




#### 4. Определить знак числа

```
fun Sign(number: Int): String {
    return when {
        number > 0 -> "Положительное"
        number < 0 -> "Отрицательное"
        else -> "Ноль"
    }
}

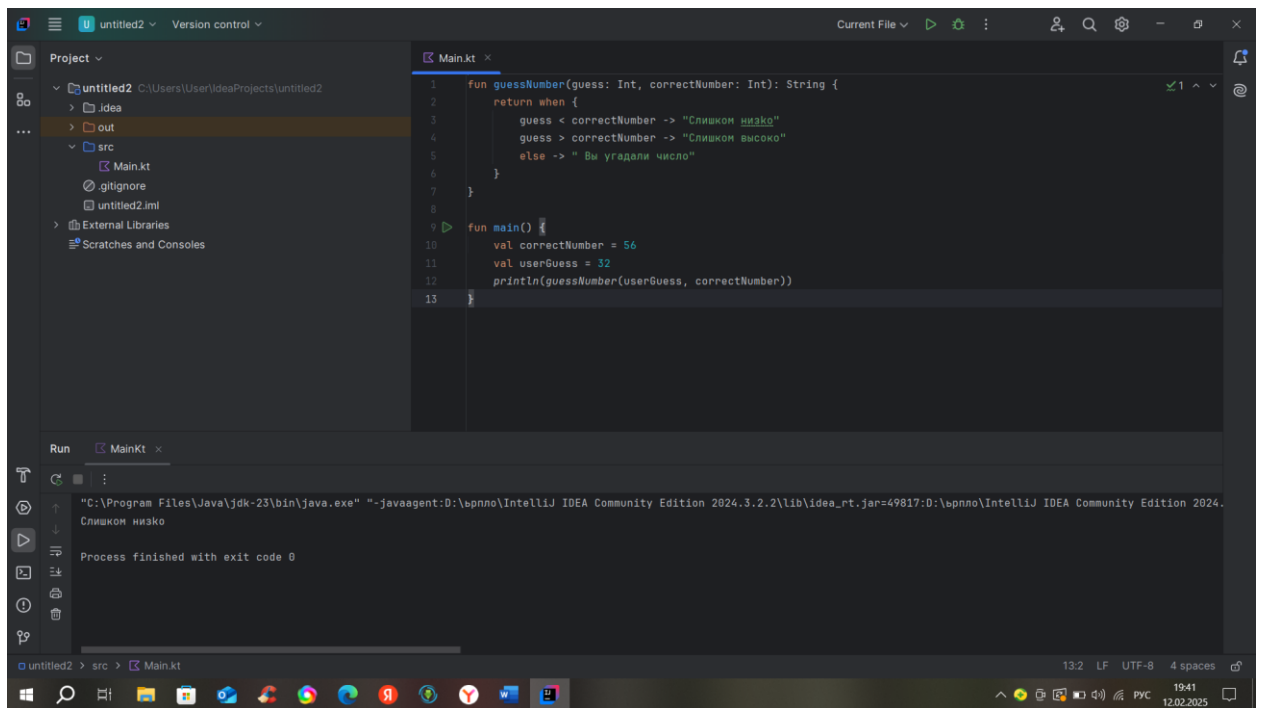
fun main() {
    val number = -10
    println(Sign(number))
}
```



## 5. Угадай число

```
fun guessNumber(guess: Int, correctNumber: Int): String {
    return when {
        guess < correctNumber -> "Слишком низко"
        guess > correctNumber -> "Слишком высоко"
        else -> "Вы угадали число"
    }
}

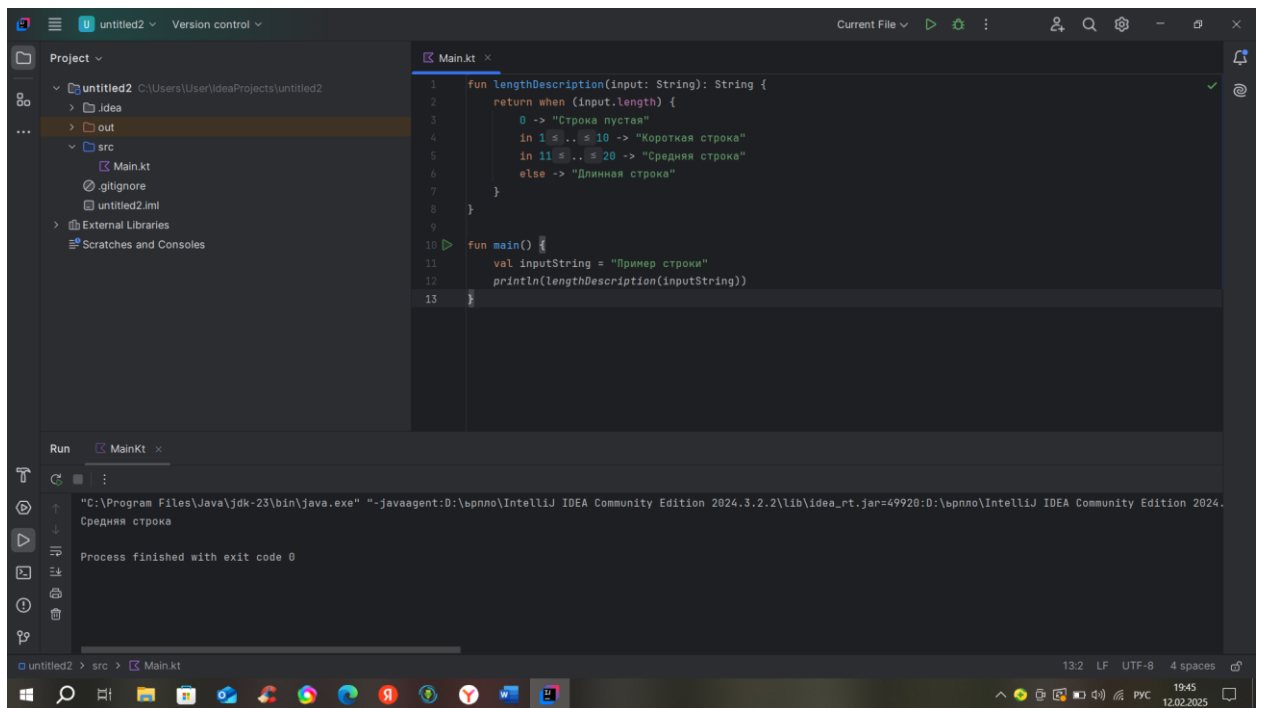
fun main() {
    val correctNumber = 56
    val userGuess = 32
    println(guessNumber(userGuess, correctNumber))
}
```



## 6. Определение длины строки

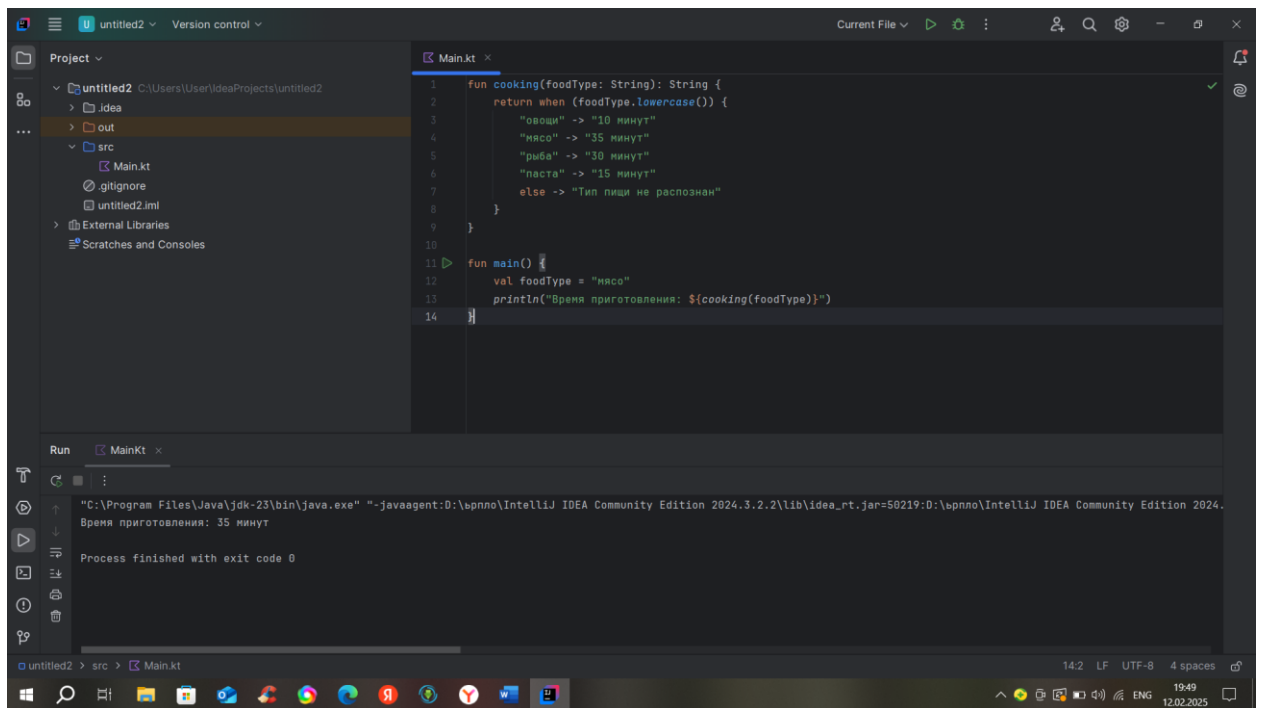
```
fun lengthDescription(input: String): String {
    return when (input.length) {
        0 -> "Строка пустая"
        in 1..10 -> "Короткая строка"
        in 11..20 -> "Средняя строка"
        else -> "Длинная строка"
    }
}

fun main() {
    val inputString = "Пример строки"
    println(lengthDescription(inputString))
}
```



## 7. Определить время приготовления по типу пищи

```
8. fun cooking(foodType: String): String {  
    return when (foodType.lowercase()) {  
        "овощи" -> "10 минут"  
        "мясо" -> "35 минут"  
        "рыба" -> "30 минут"  
        "паста" -> "15 минут"  
        else -> "Тип пищи не распознан"  
    }  
}  
  
fun main() {  
    val foodType = "мясо"  
    println("Время приготовления: ${cooking(foodType)}")  
}
```

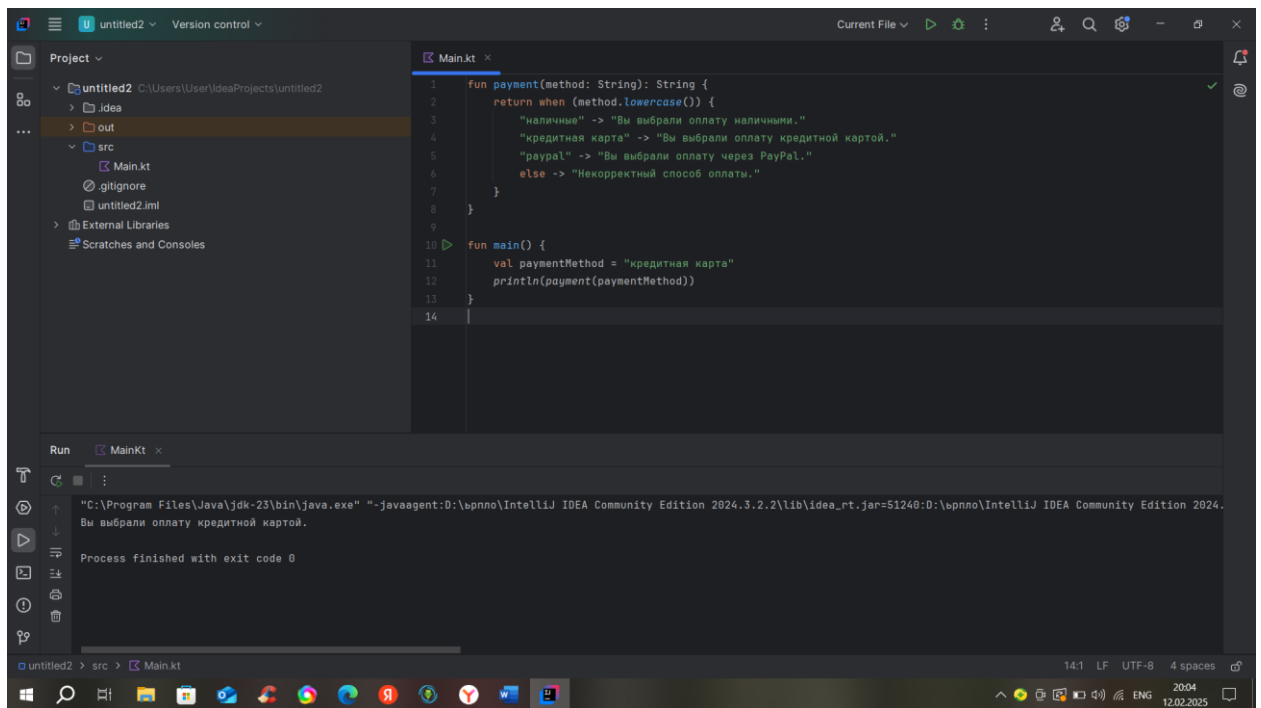


10.Способы оплаты: Напишите программу, которая принимает способ оплаты (наличные, кредитная карта, PayPal) и выдает соответствующее сообщение.

```
fun payment(method: String): String {
    return when (method.lowercase()) {
        "наличные" -> "Вы выбрали оплату наличными."
        "кредитная карта" -> "Вы выбрали оплату кредитной картой."
        "paypal" -> "Вы выбрали оплату через PayPal."
        else -> "Некорректный способ оплаты."
    }
}

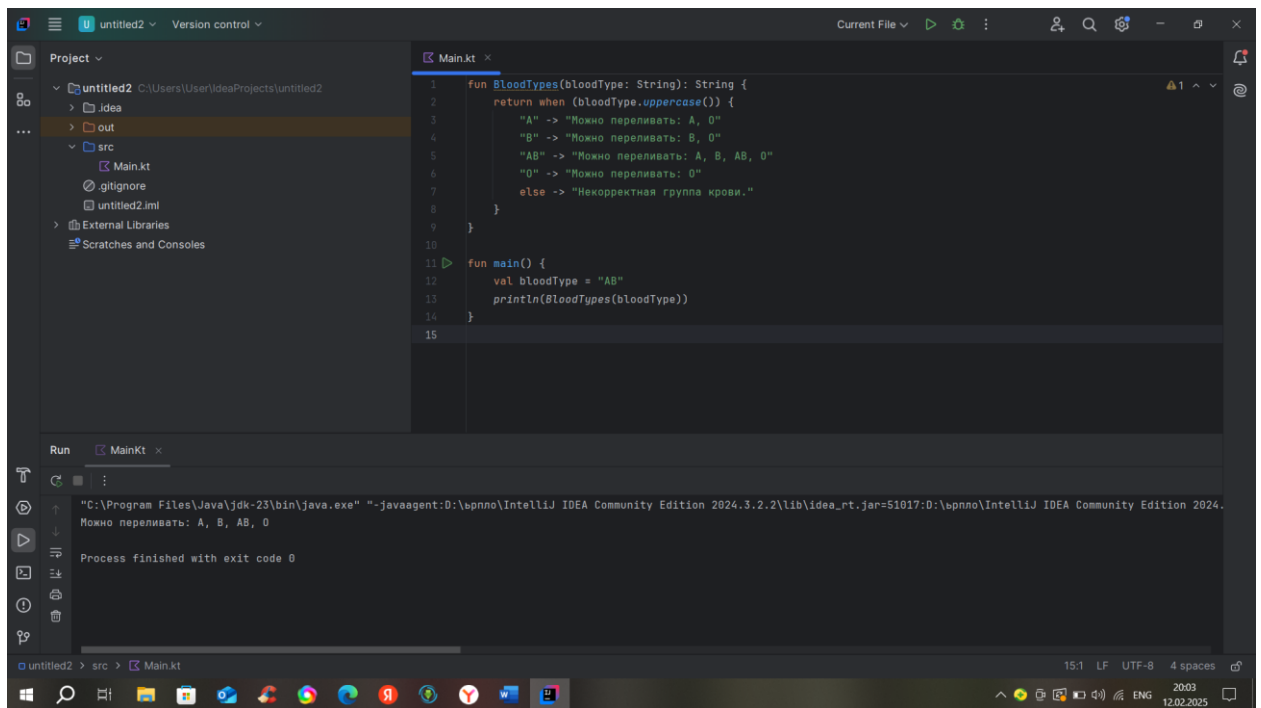
fun main() {
    val paymentMethod = "кредитная карта"
    println(payment(paymentMethod))
}
```





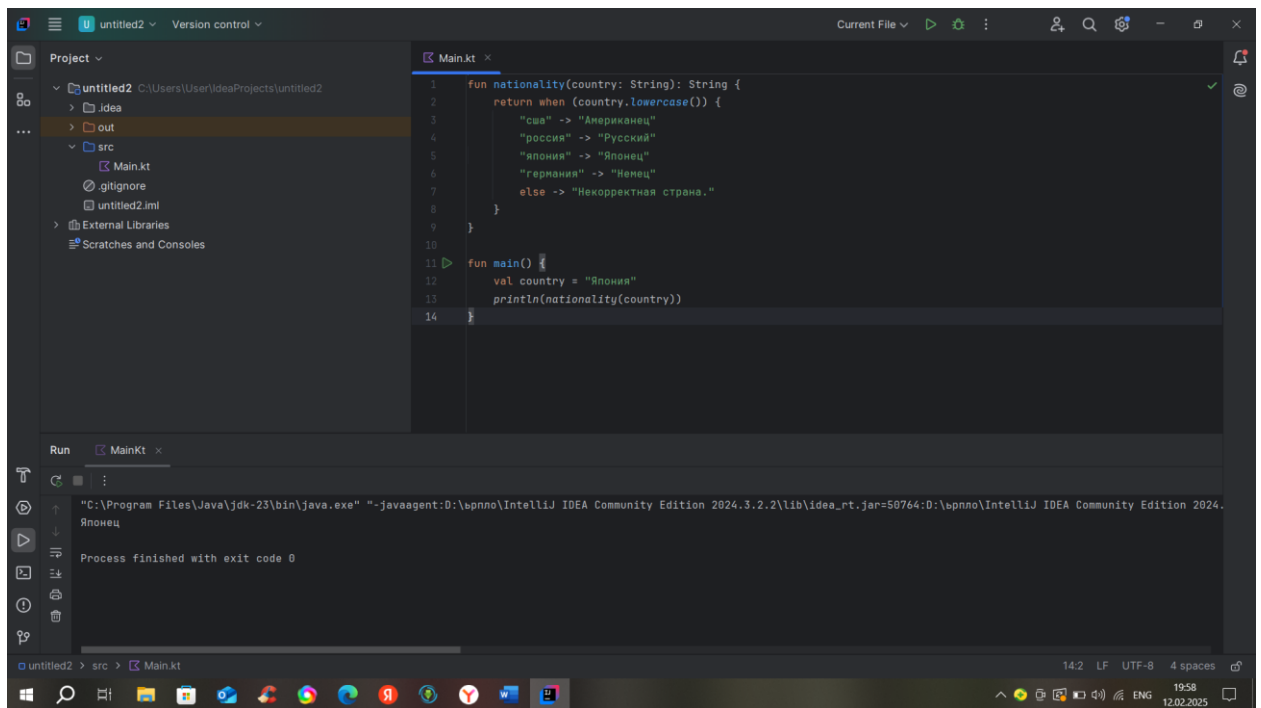
11.Группа крови: вводится группа крови (A, B, AB, O) и выводится, какие типы крови можно применять для переливания.

```
fun BloodTypes(bloodType: String): String {  
    return when (bloodType.uppercase()) {  
        "A" -> "Можно переливать: A, O"  
        "B" -> "Можно переливать: B, O"  
        "AB" -> "Можно переливать: A, B, AB, O"  
        "O" -> "Можно переливать: O"  
        else -> "Некорректная группа крови."  
    }  
}  
  
fun main() {  
    val bloodType = "AB"  
    println(BloodTypes(bloodType))  
}
```



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

```
fun nationality(country: String): String {  
    return when (country.lowercase()) {  
        "сша" -> "Американец"  
        "россия" -> "Русский"  
        "япония" -> "Японец"  
        "германия" -> "Немец"  
        else -> "Некорректная страна."  
    }  
}  
  
fun main() {  
    val country = "Япония"  
    println(nationality(country))  
}
```



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

```
fun errorMessage(code: Int): String {
    return when (code) {
        100 -> "Ошибка сети"
        200 -> "Ошибка сервера"
        300 -> "Ошибка доступа"
        else -> "Неизвестный код ошибки."
    }
}

fun main() {
    val errorCode = 200
    println(errorMessage(errorCode))
}
```

