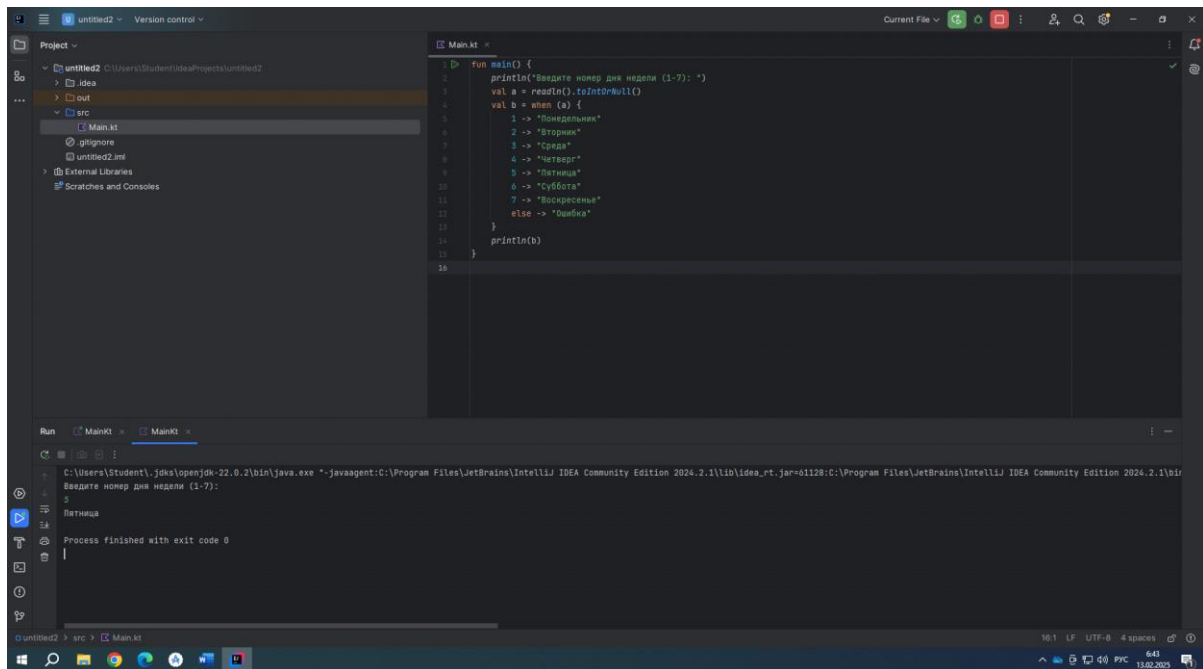


1)Определить день недели по номеру

```
fun main() {  
    println("Введите номер дня недели (1-7): ")  
    val a = readln().toIntOrNull()  
    val b = when (a) {  
        1 -> "Понедельник"  
        2 -> "Вторник"  
        3 -> "Среда"  
        4 -> "Четверг"  
        5 -> "Пятница"  
        6 -> "Суббота"  
        7 -> "Воскресенье"  
        else -> "Ошибка"  
    }  
    println(b)  
}
```



2) Определить тип треугольника по длинам сторон

```
fun main() {
```

```
    println("Введите длину первой стороны: ")
```

```
    val a = readln().toDoubleOrNull()
```

```
    println("Введите длину второй стороны: ")
```

```
    val b = readln().toDoubleOrNull()
```

```
    println("Введите длину третьей стороны: ")
```

```
    val c = readln().toDoubleOrNull()
```

```
    if (a == null || b == null || c == null || a <= 0 || b <= 0 || c <= 0) {
```

```
        println("Ошибка")
```

```
        return
```

```
    }
```

```
    if (a + b <= c || a + c <= b || b + c <= a) {
```

```
        println("Треугольник не существует.")
```

```
        return
```

```
}
```

```
val i = when {
```

```
    a == b && b == c -> "Равносторонний треугольник"
```

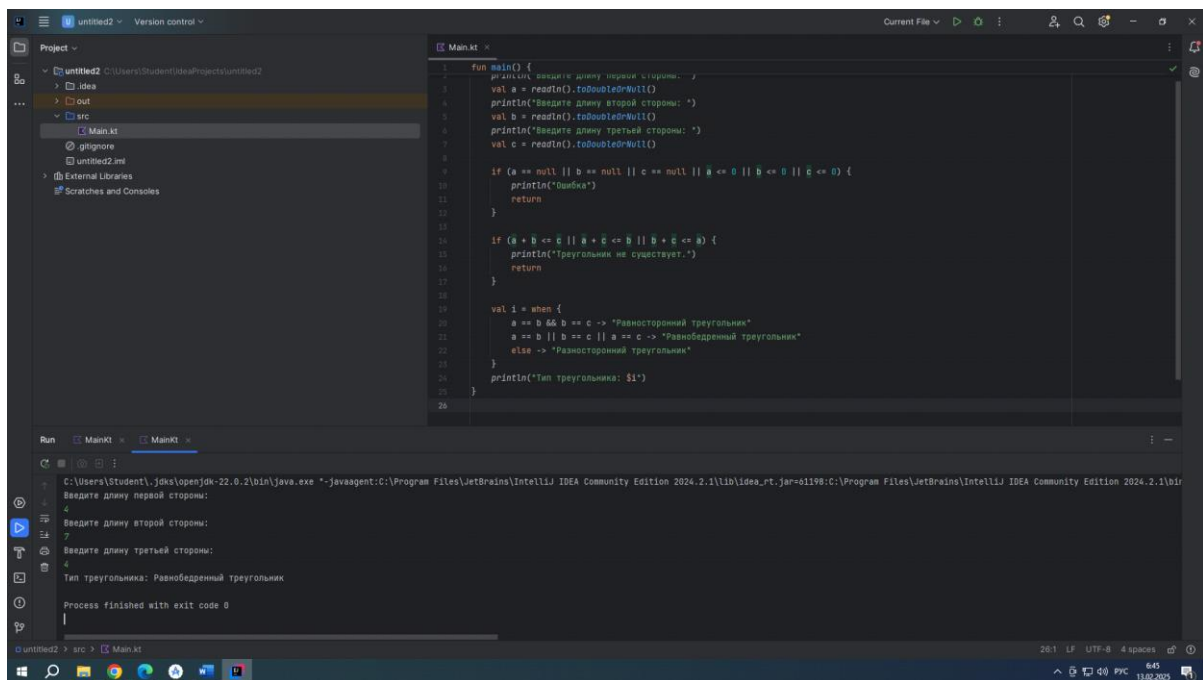
```
    a == b || b == c || a == c -> "Равнобедренный треугольник"
```

```
    else -> "Разносторонний треугольник"
```

```
}
```

```
println("Тип треугольника: $i")
```

```
}
```



3) Вывод оценок по числовым значениям

```
fun main() {
```

```
    println("Введите числовую оценку (0-100):")
```

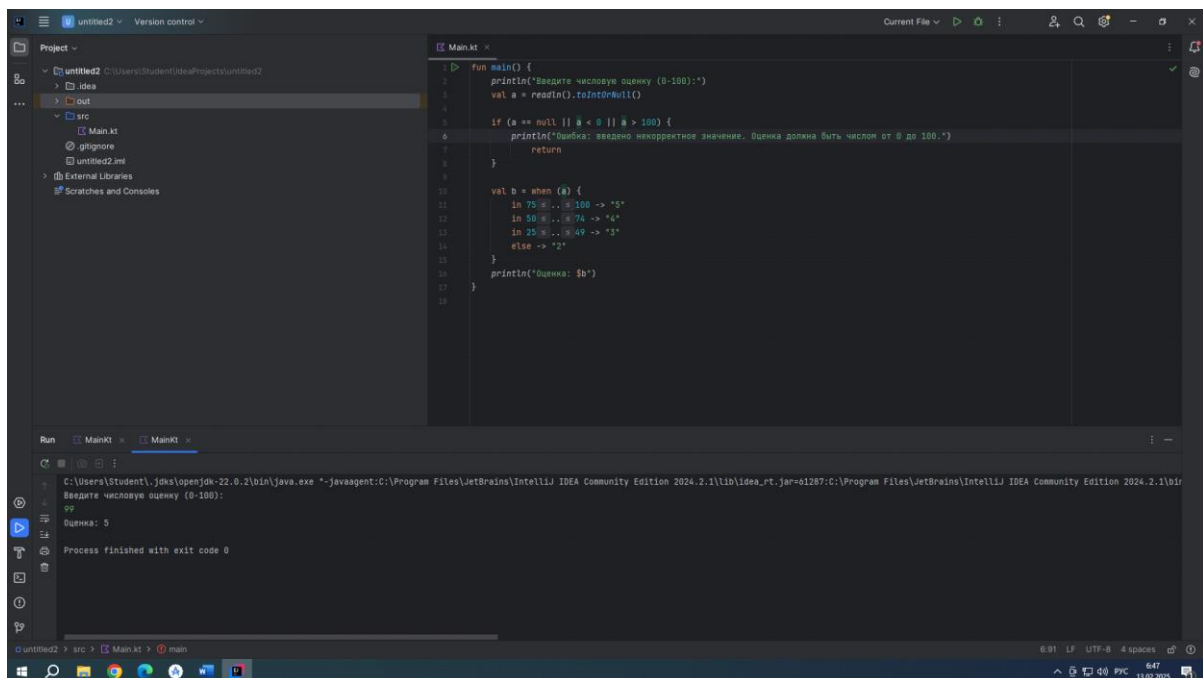
```
    val a = readln().toIntOrNull()
```

```

if (a == null || a < 0 || a > 100) {
    println("Ошибка: введено некорректное значение. Оценка должна быть
числом от 0 до
100.")
    return
}

val b = when (a) {
    in 75..100 -> "5"
    in 50..74 -> "4"
    in 25..49 -> "3"
    else -> "2"
}
println("Оценка: $b")
}

```



4)Определение времени суток

```

fun main() {

```

```
println("Введите текущий час (0-23):")
```

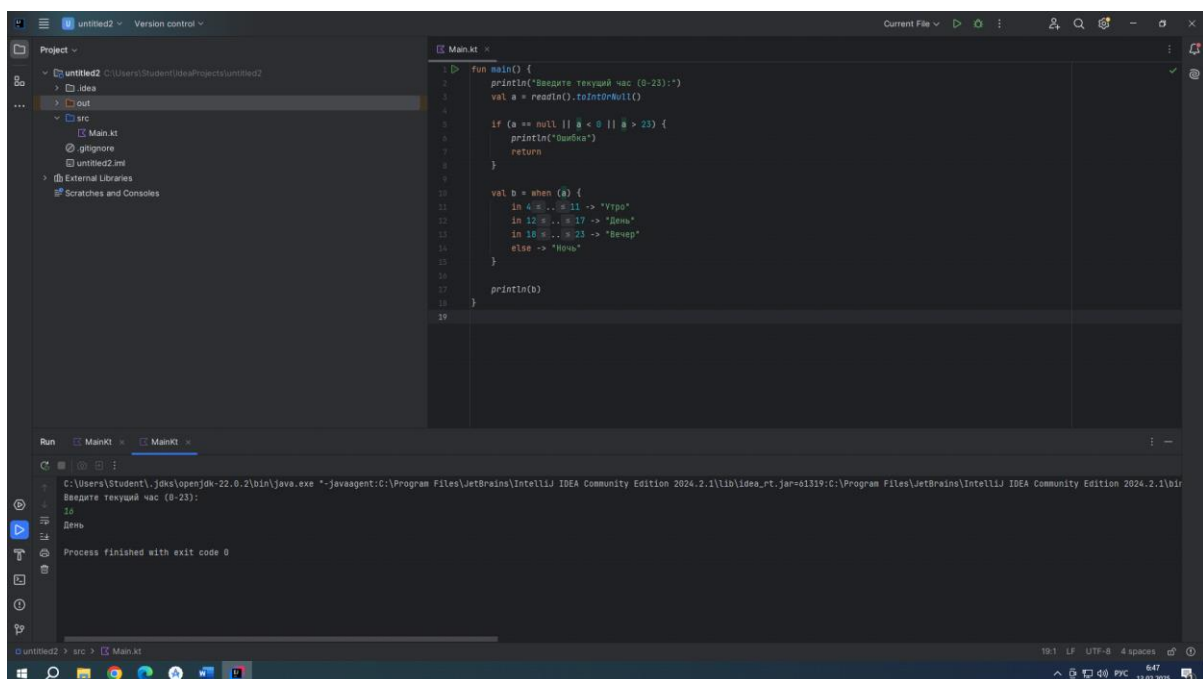
```
val a = readln().toIntOrNull()
```

```
if (a == null || a < 0 || a > 23) {  
    println("Ошибка")  
    return  
}
```

```
val b = when (a) {  
    in 4..11 -> "Утро"  
    in 12..17 -> "День"  
    in 18..23 -> "Вечер"  
    else -> "Ночь"  
}
```

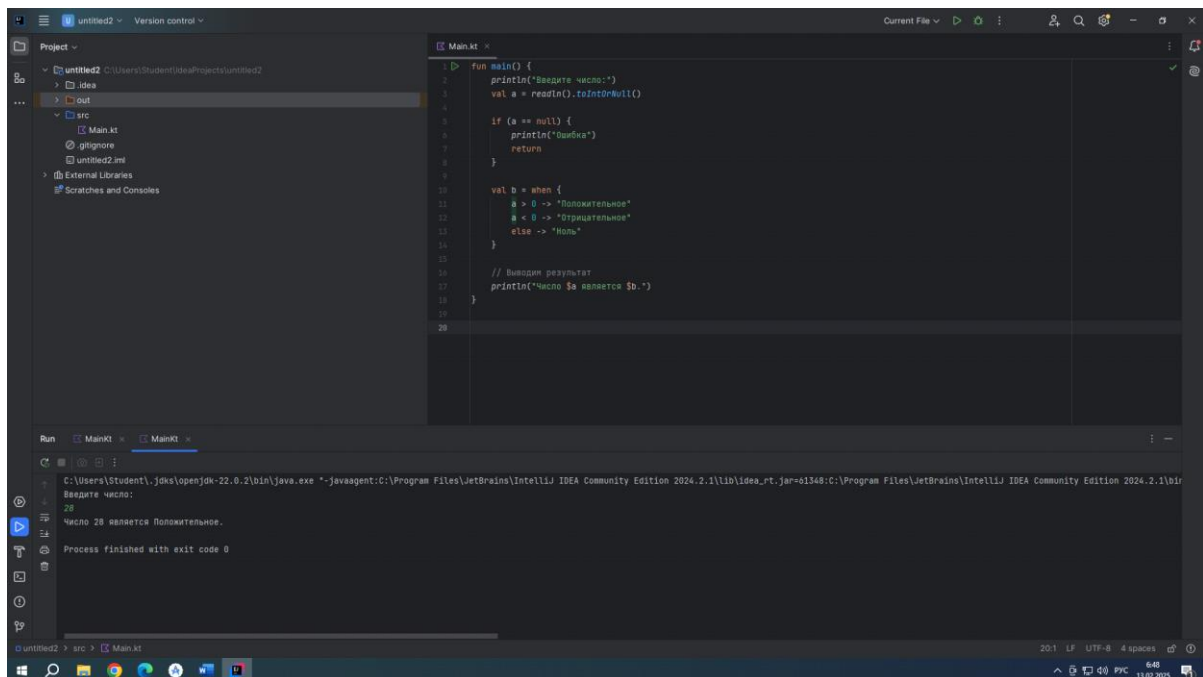
```
println(b)
```

```
}
```



5) Определить знак числа

```
fun main() {  
    println("Введите число:")  
    val a = readln().toIntOrNull()  
  
    if (a == null) {  
        println("Ошибка")  
        return  
    }  
  
    val b = when {  
        a > 0 -> "Положительное"  
        a < 0 -> "Отрицательное"  
        else -> "Ноль"  
    }  
    println("Число $a является $b.")  
}
```



6) Угадать число

```
import kotlin.random.Random
```

```
fun main() {
```

```
val a = Random.nextInt(1, 11)
```

```
var b = false
```

```
println("Угадайте число от 1 до 10:")
```

```
while (!b) {
```

```
val c = readLine()
```

```
if (c != null) {
```

```
val d = c.toIntOrNull()
```

```
if (d != null) {
```

when {

```
d < a -> println("Слишком мало")
```

```
d > a -> println("Слишком много")
```

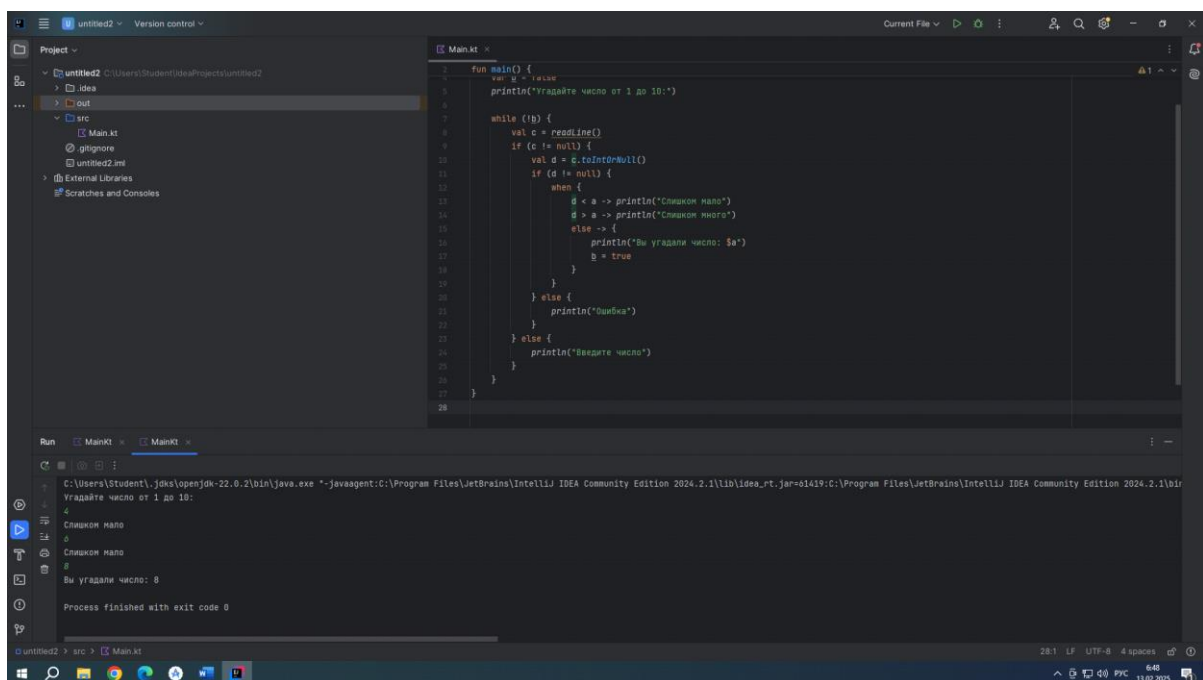
else -> {

```
println("Вы угадали число: $a")
```

```

        b = true
    }
}
} else {
    println("Ошибка")
}
} else {
    println("Введите число")
}
}
}

```



7) Определение длины строки

```

fun main() {
    println("Введите строку:")
    val a = readLn()
}

```

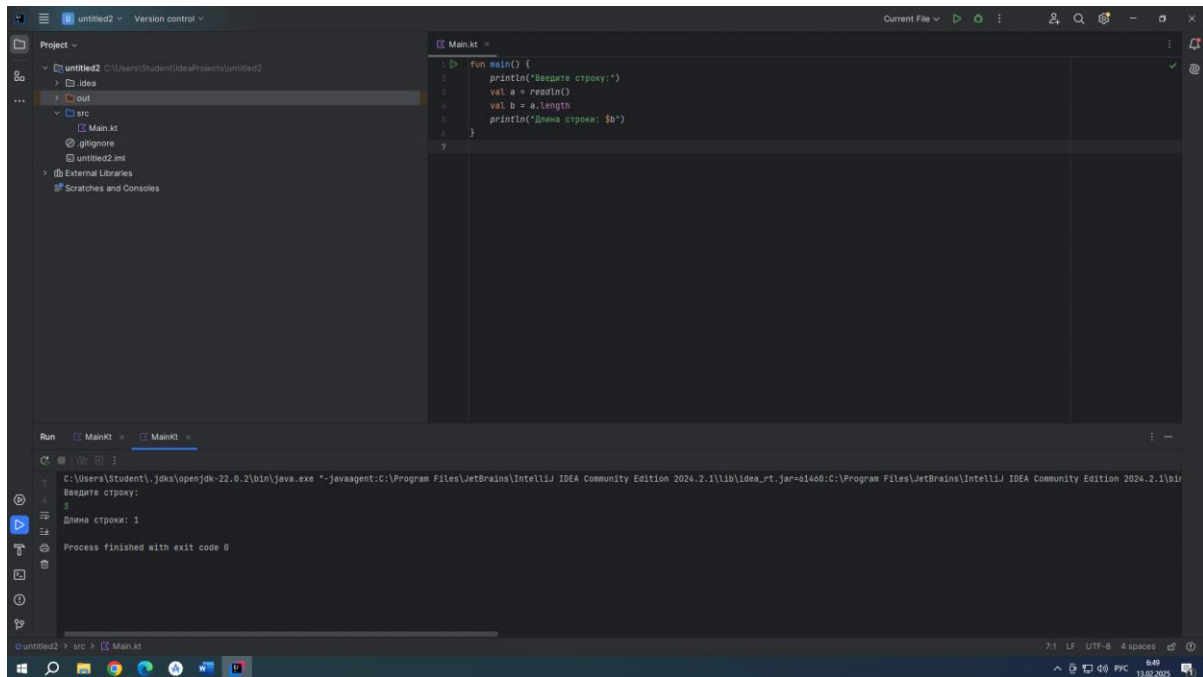


```

val b = a.length

println("Длина строки: $b")
}

```



8) Определить время приготовления по типу пищи

```

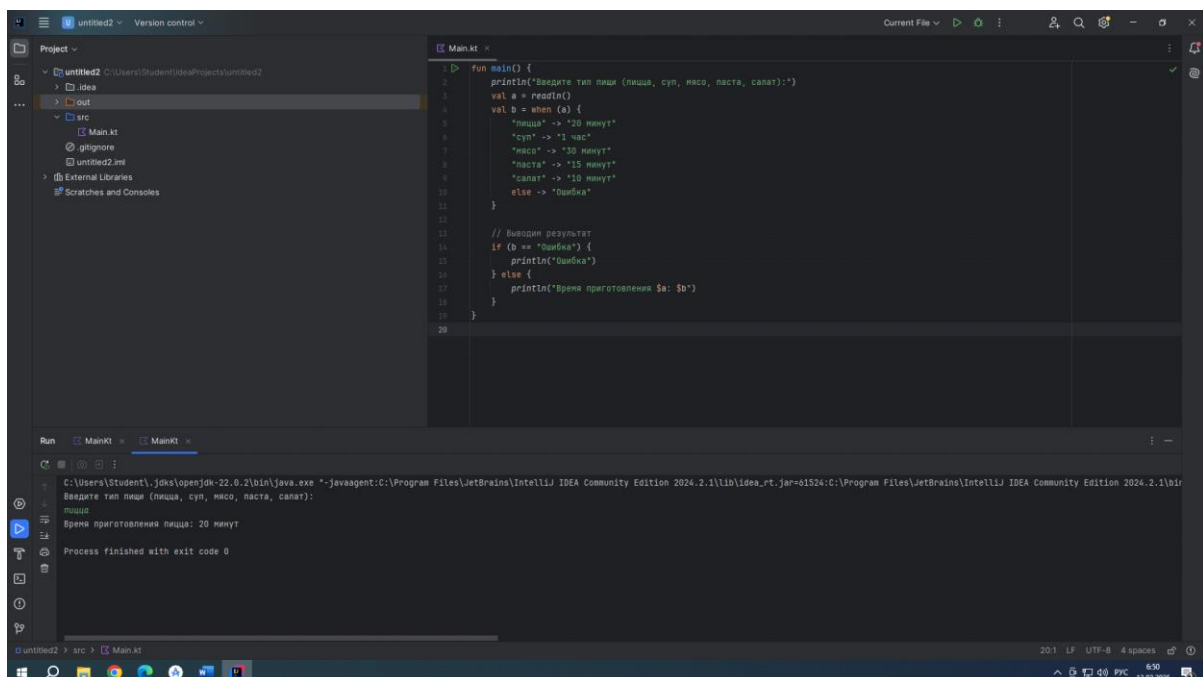
fun main() {
    println("Введите тип пищи (пицца, суп, мясо, паста, салат):")
    val a = readln()
    val b = when (a) {
        "пицца" -> "20 минут"
        "суп" -> "1 час"
        "мясо" -> "30 минут"
        "паста" -> "15 минут"
        "салат" -> "10 минут"
        else -> "Ошибка"
    }
}

```

```

    if (b == "Ошибка") {
        println("Ошибка")
    } else {
        println("Время приготовления $a: $b")
    }
}

```



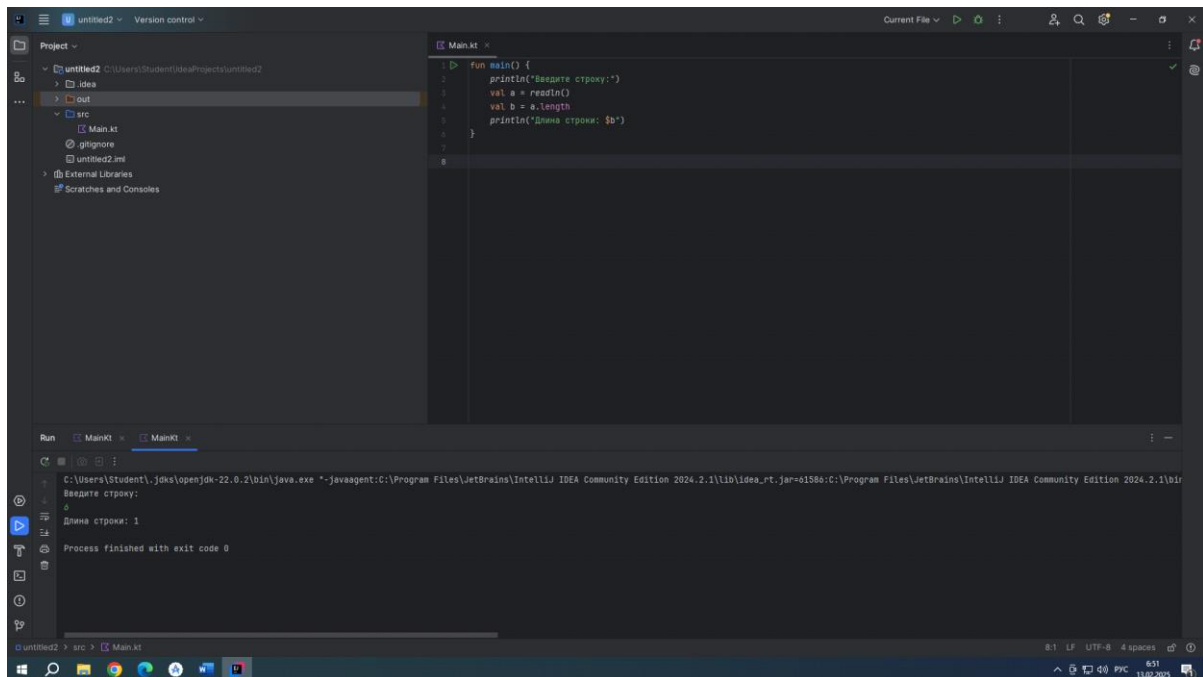
9) Определение длины строки

```

fun main() {
    println("Введите строку:")
    val a = readln()
    val b = a.length
    println("Длина строки: $b")
}

```

}

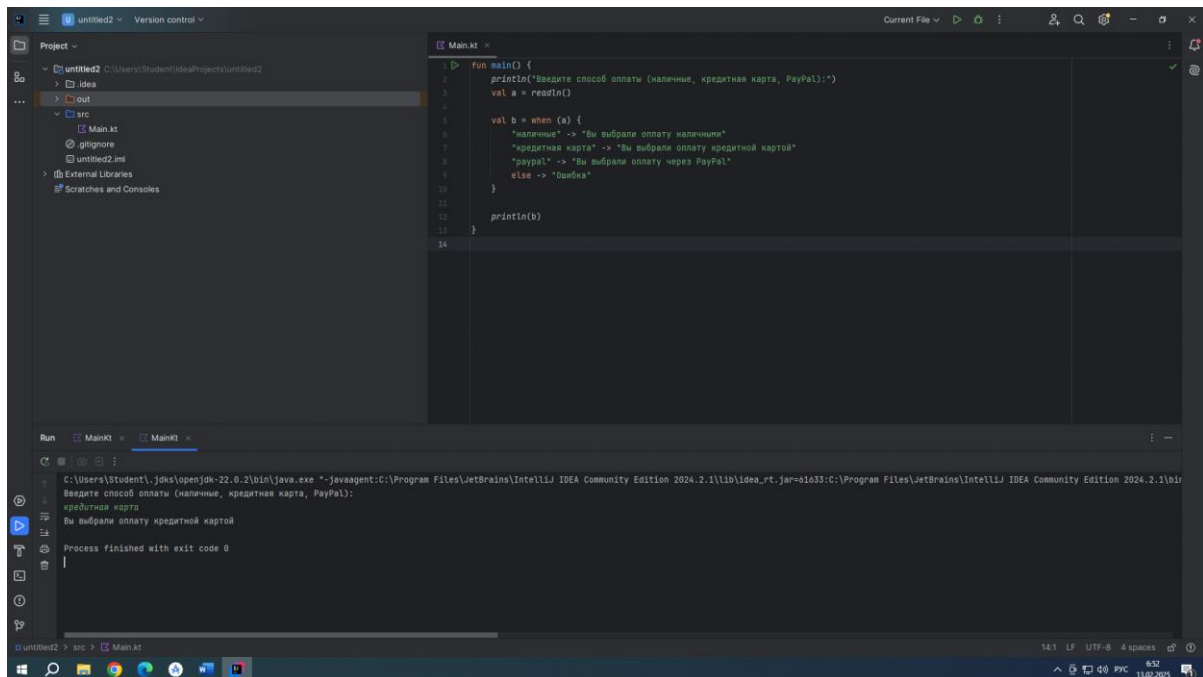


10) Способы оплаты: Напишите программу, которая принимает способ оплаты

(кредитная карта наличные, , PayPal) и выдает соответствующее сообщение.

```
fun main() {  
    println("Введите способ оплаты (наличные, кредитная карта, PayPal):")  
    val a = readln()  
  
    val b = when (a) {  
        "наличные" -> "Вы выбрали оплату наличными"  
        "кредитная карта" -> "Вы выбрали оплату кредитной картой"  
        "paypal" -> "Вы выбрали оплату через PayPal"  
        else -> "Ошибка"  
    }  
  
    println(b)
```

}



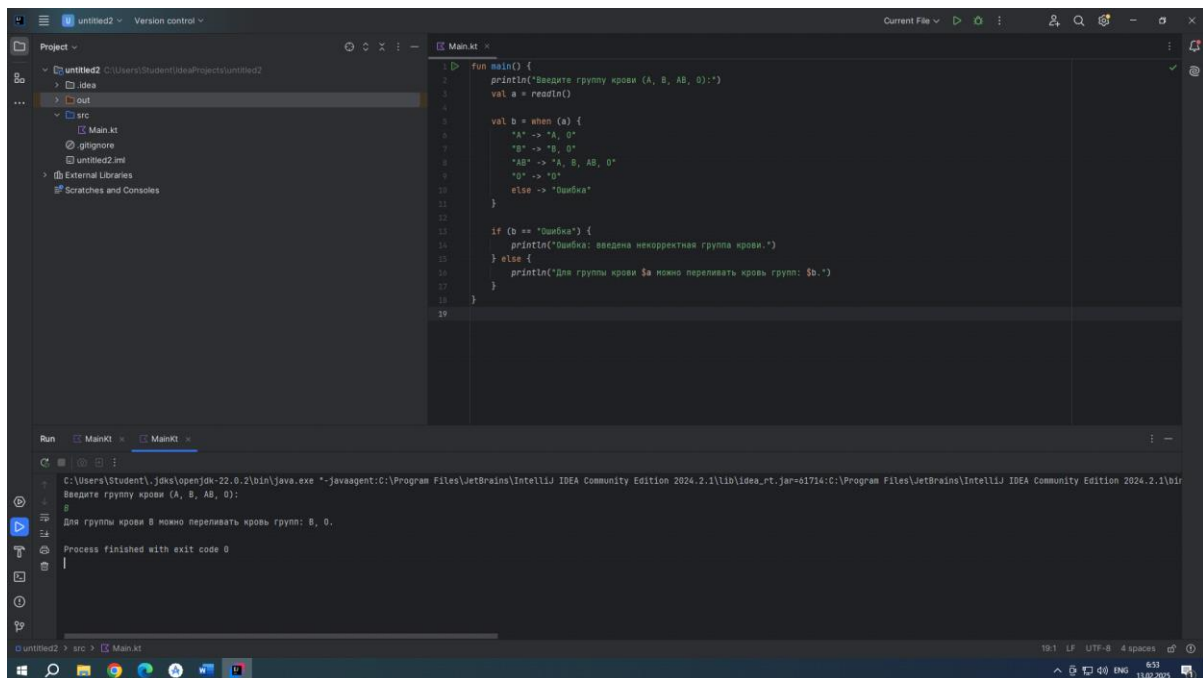
11) Группа крови: вводится группа крови (A, B, AB, O) и выводится, какие типы крови можно применять для переливания.

```
fun main() {  
    println("Введите группу крови (A, B, AB, O):")  
    val a = readln()  
  
    val b = when (a) {  
        "A" -> "A, O"  
        "B" -> "B, O"  
        "AB" -> "A, B, AB, O"  
        "O" -> "O"  
        else -> "Ошибка"  
    }  
}
```

```

if (b == "Ошибка") {
    println("Ошибка: введена некорректная группа крови.")
} else {
    println("Для группы крови $a можно переливать кровь групп: $b.")
}
}

```



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

```

fun main() {
    // Запрашиваем у пользователя ввод страны
    println("Введите страну (США, Россия, Япония, Германия, Франция):")
    val a = readln()

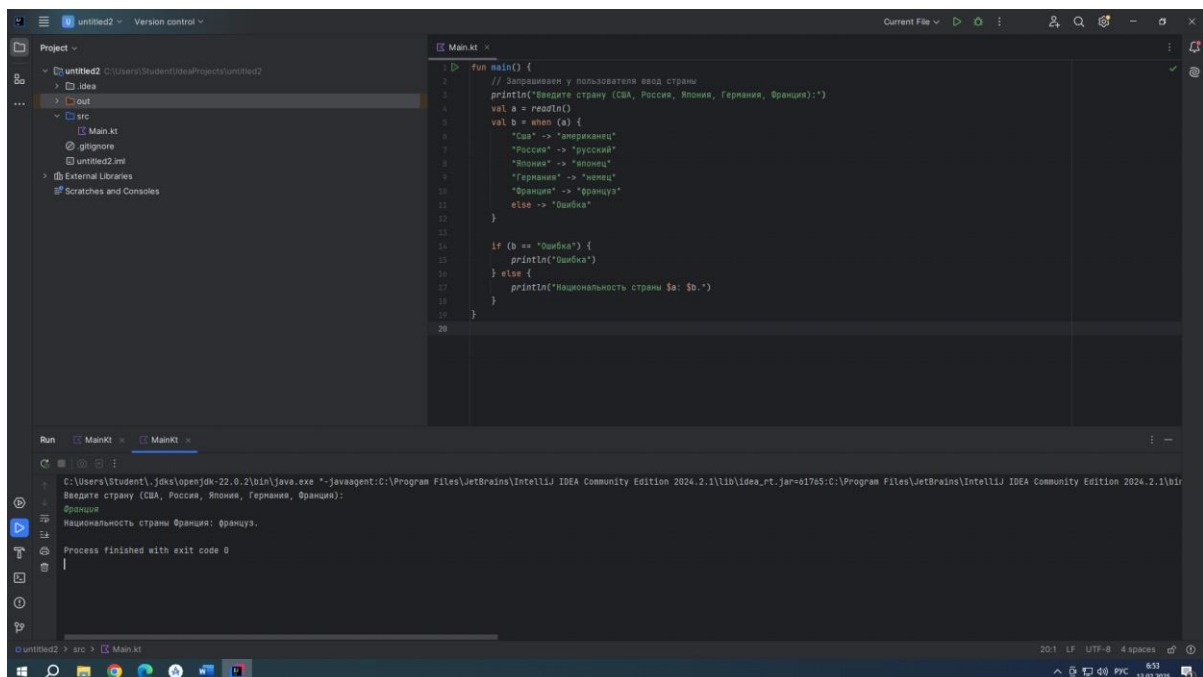
```

```

val b = when (a) {
    "Сша" -> "американец"
    "Россия" -> "русский"
    "Япония" -> "японец"
    "Германия" -> "немец"
    "Франция" -> "француз"
    else -> "Ошибка"
}

if (b == "Ошибка") {
    println("Ошибка")
} else {
    println("Национальность страны $a: $b.")
}
}

```



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 a = readln().toIntOrNull()  
    if (a == null) {  
        println("Ошибка")  
        return  
    }  
  
    val b = when (a) {  
        100 -> "Ошибка сети"  
        200 -> "Ошибка сервера"  
        300 -> "Ошибка базы данных"  
        else -> "Неизвестный код ошибки"  
    }  
    println(b)  
}
```

