

Практическая работа № 4

1. 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"

}

}

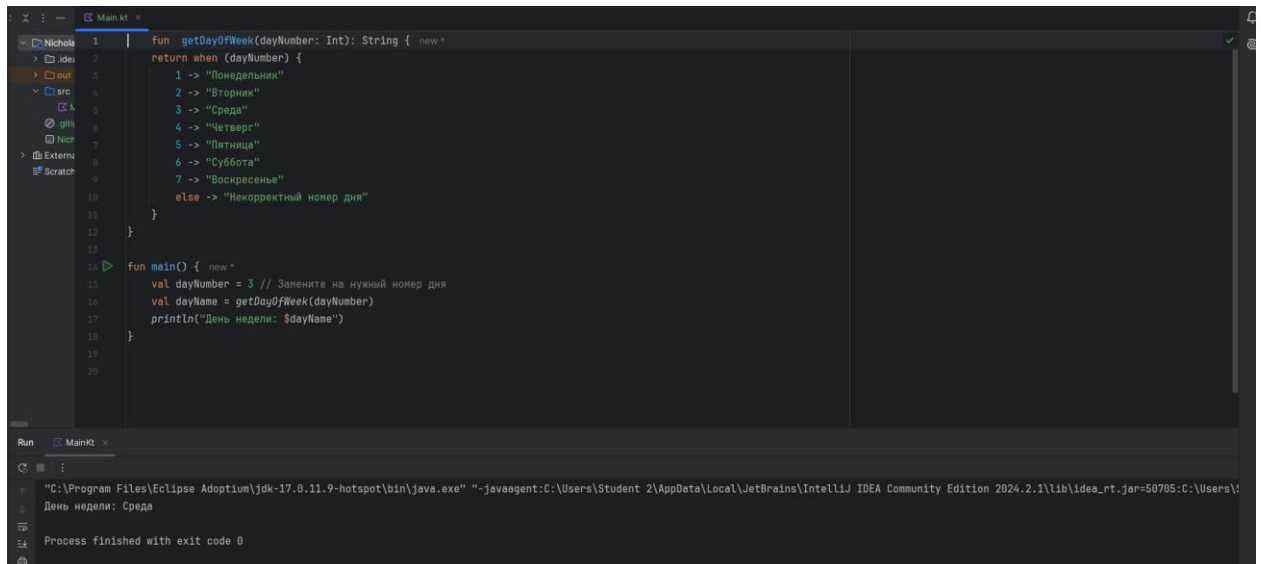
fun main() {

val dayNumber = 3 // Замените на нужный номер дня

val dayName = getDayOfWeek(dayNumber)

println("День недели: \$dayName")

}



2

fun determineTriangleType(a: Double, b: Double, c: Double): String {

// Проверка на существование треугольника

if (a <= 0 || b <= 0 || c <= 0 || a + b <= c || a + c <= b || b + c <= a) {

return "Треугольник не существует"

}

```

// Определение типа треугольника

return when {

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

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

    else -> "Разносторонний треугольник"

}

}

```

```

fun main() {

    println("Введите длины сторон треугольника:")

    val a = readLine()!!.toDouble()

    val b = readLine()!!.toDouble()

    val c = readLine()!!.toDouble()

    val result = determineTriangleType(a, b, c)

    println(result)

}

```

The screenshot shows an IDE with a Kotlin file named 'Main.kt'. The code defines a function `determineTriangleType` that takes three double values `a`, `b`, and `c` and returns a string representing the triangle type. It uses a `when` expression to handle different cases: equilateral, isosceles, and scalene. The `main` function prompts the user to enter the side lengths, reads them, and prints the result. The Run window at the bottom shows the execution output: 'Введите длины сторон треугольника:' followed by the input '4 4 3' and the output 'Разносторонний треугольник'.

```

1 fun determineTriangleType(a: Double, b: Double, c: Double): String {
2     // Проверка на существование треугольника
3     if (a <= 0 || b <= 0 || c <= 0 || a + b <= c || a + c <= b || b + c <= a) {
4         return "Треугольник не существует"
5     }
6
7     // Определение типа треугольника
8     return when {
9         a == b && b == c -> "Равносторонний треугольник"
10        a == b || b == c || a == c -> "Равнобедренный треугольник"
11        else -> "Разносторонний треугольник"
12    }
13 }
14
15 fun main() {
16     println("Введите длины сторон треугольника:")
17     val a = readLine()!!.toDouble()
18     val b = readLine()!!.toDouble()
19     val c = readLine()!!.toDouble()
20
21     val result = determineTriangleType(a, b, c)
22     println(result)
23 }

```

Run: Main.kt

```

"C:\Program Files\Eclipse Adoptium\jdk-17.0.11.9-hotspot\bin\java.exe" "-javaagent:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=50813:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -Didea.config.path=C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\config -Didea.system.path=C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\system -Didea.version=2024.2.1
Введите длины сторон треугольника:
4
4
3
Разносторонний треугольник
Process finished with exit code 0

```

```

3 fun getGrade(score: Int): String {

    return when {

        score in 90..100 -> "Отлично"

        score in 75..89 -> "Хорошо"

        score in 60..74 -> "Удовлетворительно"

        score in 0..59 -> "Неудовлетворительно"

        else -> "Недопустимое значение"

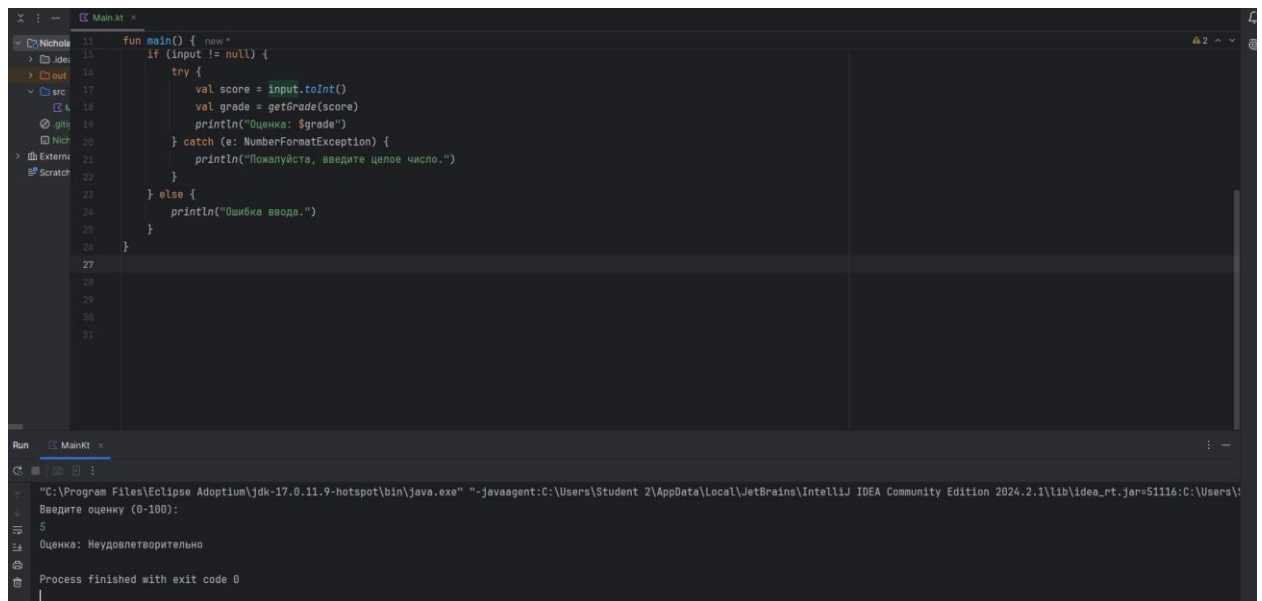
    }

}

```

```
}
```

```
fun main() {  
  
    println("Введите оценку (0-100):")  
  
    val input = readLine()  
  
    if (input != null) {  
  
        try {  
  
            val score = input.toInt()  
  
            val grade = getGrade(score)  
  
            println("Оценка: $grade")  
  
        } catch (e: NumberFormatException) {  
  
            println("Пожалуйста, введите целое число.")  
  
        }  
  
    } else {  
  
        println("Ошибка ввода.")  
  
    }  
  
}
```



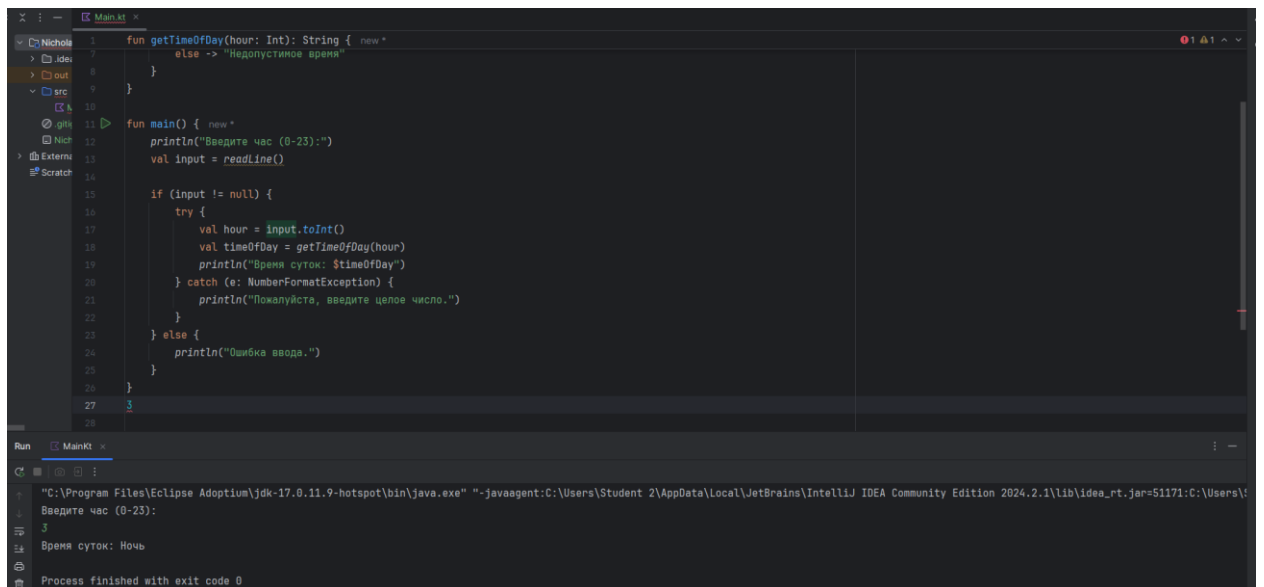
The screenshot shows an IDE window with a Kotlin file named 'Main.kt'. The code is identical to the one provided in the previous block. Below the editor, the 'Run' tab is active, showing the execution output. The output indicates that the program was executed successfully, with the user input '5' resulting in the output 'Оценка: Неудовлетворительно'. The process finished with exit code 0.

4

```
fun getTimeOfDay(hour: Int): String {  
  
    return when (hour) {  
  
        in 0..5 -> "Ночь"  
  
        in 6..11 -> "Утро"  
  
        in 12..17 -> "День"  
  
        in 18..23 -> "Вечер"  
  
        else -> "Недопустимое время"  
  
    }  
  
}
```

```
}  
}
```

```
fun main() {  
  
    println("Введите час (0-23):")  
  
    val input = readLine()  
  
    if (input != null) {  
  
        try {  
  
            val hour = input.toInt()  
  
            val timeOfDay = getTimeOfDay(hour)  
  
            println("Время суток: $timeOfDay")  
  
        } catch (e: NumberFormatException) {  
  
            println("Пожалуйста, введите целое число.")  
  
        }  
  
    } else {  
  
        println("Ошибка ввода.")  
  
    }  
  
}
```



The screenshot shows an IDE window with a Kotlin file named 'Main.kt'. The code defines a function `getTimeOfDay` and a `main` function. The `main` function prompts the user to enter a time in hours (0-23), reads the input, and uses `getTimeOfDay` to determine the time of day. It includes error handling for non-integer inputs. The IDE's Run window at the bottom shows the execution output: 'Введите час (0-23):', '3', and 'Время суток: Ночь'. The process finished with exit code 0.

```
1 fun getTimeOfDay(hour: Int): String {  
2     else -> "Недопустимое время"  
3 }  
4  
5 fun main() {  
6     println("Введите час (0-23):")  
7     val input = readLine()  
8  
9     if (input != null) {  
10  
11         try {  
12  
13             val hour = input.toInt()  
14  
15             val timeOfDay = getTimeOfDay(hour)  
16  
17             println("Время суток: $timeOfDay")  
18  
19         } catch (e: NumberFormatException) {  
20  
21             println("Пожалуйста, введите целое число.")  
22  
23         }  
24  
25     } else {  
26  
27         println("Ошибка ввода.")  
28  
29     }  
30 }  
31
```

Run Main.kt

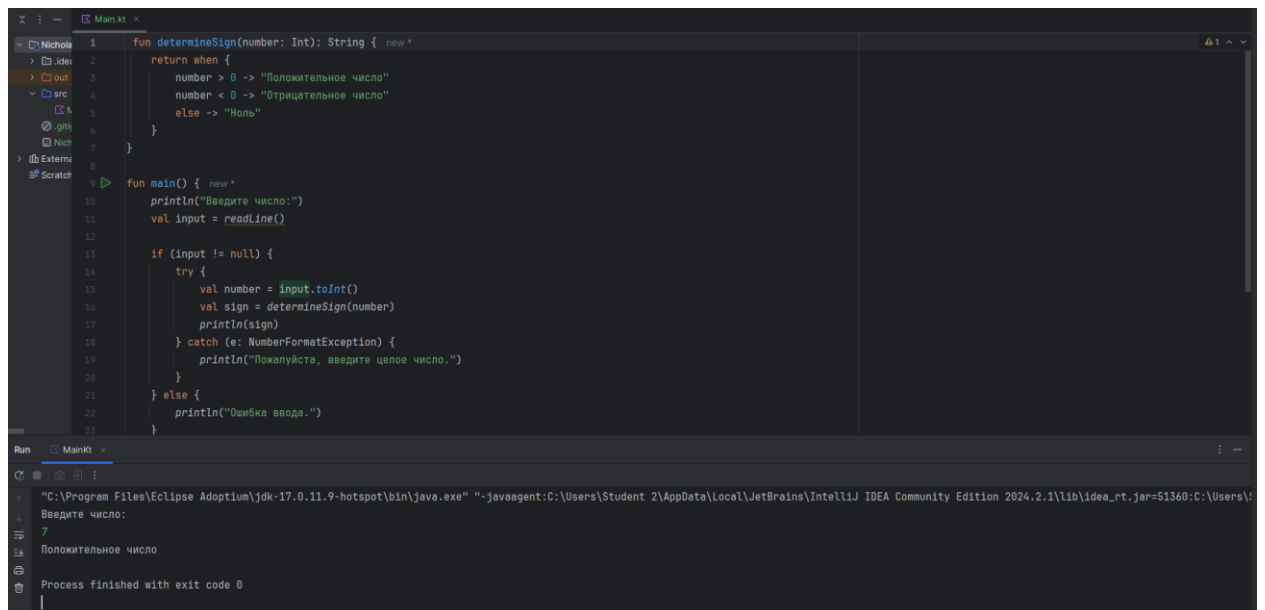
"C:\Program Files\Eclipse Adoptium\jdk-17.0.11.9-hotspot\bin\java.exe" "-javaagent:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=51171:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -Didea.config.path=C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\config -Didea.system.path=C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin -Didea.version=2024.2.1 -jar C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin\idea_rt.jar 51171

Введите час (0-23):
3
Время суток: Ночь
Process finished with exit code 0

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

```
}
```

```
fun main() {  
  
    println("Введите число:")  
    val input = readLine()  
  
    if (input != null) {  
        try {  
            val number = input.toInt()  
            val sign = determineSign(number)  
            println(sign)  
        } catch (e: NumberFormatException) {  
            println("Пожалуйста, введите целое число.")  
        }  
    } else {  
        println("Ошибка ввода.")  
    }  
}
```



The screenshot shows an IDE with a Kotlin file named Main.kt. The code defines a function `determineSign` that takes an `Int` and returns a `String`. It uses a `when` expression to return "Положительное число" for positive numbers, "Отрицательное число" for negative numbers, and "Ноль" for zero. The `main` function prompts the user to enter a number, reads the input, and uses a `try-catch` block to handle `NumberFormatException`. The output window shows the program running successfully, with the user inputting "7" and the program outputting "Положительное число".

```
1 fun determineSign(number: Int): String { new *  
2     return when {  
3         number > 0 -> "Положительное число"  
4         number < 0 -> "Отрицательное число"  
5         else -> "Ноль"  
6     }  
7 }  
8  
9 fun main() { new *  
10     println("Введите число:")  
11     val input = readLine()  
12  
13     if (input != null) {  
14         try {  
15             val number = input.toInt()  
16             val sign = determineSign(number)  
17             println(sign)  
18         } catch (e: NumberFormatException) {  
19             println("Пожалуйста, введите целое число.")  
20         }  
21     } else {  
22         println("Ошибка ввода.")  
23     }  
24 }
```

Run Main.kt
C:\Program Files\Eclipse Adoptium\jdk-17.0.11.9-hotspot\bin\java.exe "-javaagent:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=51360:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -Didea.config.path=C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\config -Didea.system.path=C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin -Didea.version=2024.2.1 -jar C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin\idea_rt.jar 51360
Введите число:
7
Положительное число
Process finished with exit code 0

6

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

```
fun main() {  
  
    val numberToGuess = Random.nextInt(1, 101) // Случайное число от 1 до 100  
    var attempts = 0  
    var guessed = false
```

```
println("Угадайте число от 1 до 100:")

while (!guessed) {
    println("Введите ваше предположение:")
    val input = readLine()

    if (input != null) {
        try {
            val guess = input.toInt()
            attempts++

            when {
                guess < numberToGuess -> println("Слишком низко, попробуйте снова.")
                guess > numberToGuess -> println("Слишком высоко, попробуйте снова.")
                else -> {
                    guessed = true
                    println("Поздравляю! Вы угадали число $numberToGuess за $attempts попыток.")
                }
            }
        } catch (e: NumberFormatException) {
            println("Пожалуйста, введите целое число.")
        }
    } else {
        println("Ошибка ввода.")
    }
}
```



```

fun main() {

    println("Введите тип пищи (мясо, рыба, овощи, крупы):")

    val типПищи = readLine()?.toLowerCase()

    val времяПриготовления: String = when (типПищи) {

        "мясо" -> "Время приготовления: 1.5 - 2 часа"

        "рыба" -> "Время приготовления: 20 - 30 минут"

        "овощи" -> "Время приготовления: 15 - 25 минут"

        "крупы" -> "Время приготовления: 20 - 40 минут"

        else -> "Неизвестный тип пищи. Пожалуйста, введите один из предложенных типов."

    }

    println(времяПриготовления)

}

```

The screenshot shows an IDE window with a Kotlin file named 'Main.kt'. The code is identical to the one provided in the previous block. Below the editor, the 'Run' console shows the execution output. The user entered 'мясо' (meat), and the program correctly output 'Время приготовления: 1.5 - 2 часа' (Cooking time: 1.5 - 2 hours). The console also shows the Java command used to run the program and the exit code 0.

9.

```

fun main() {

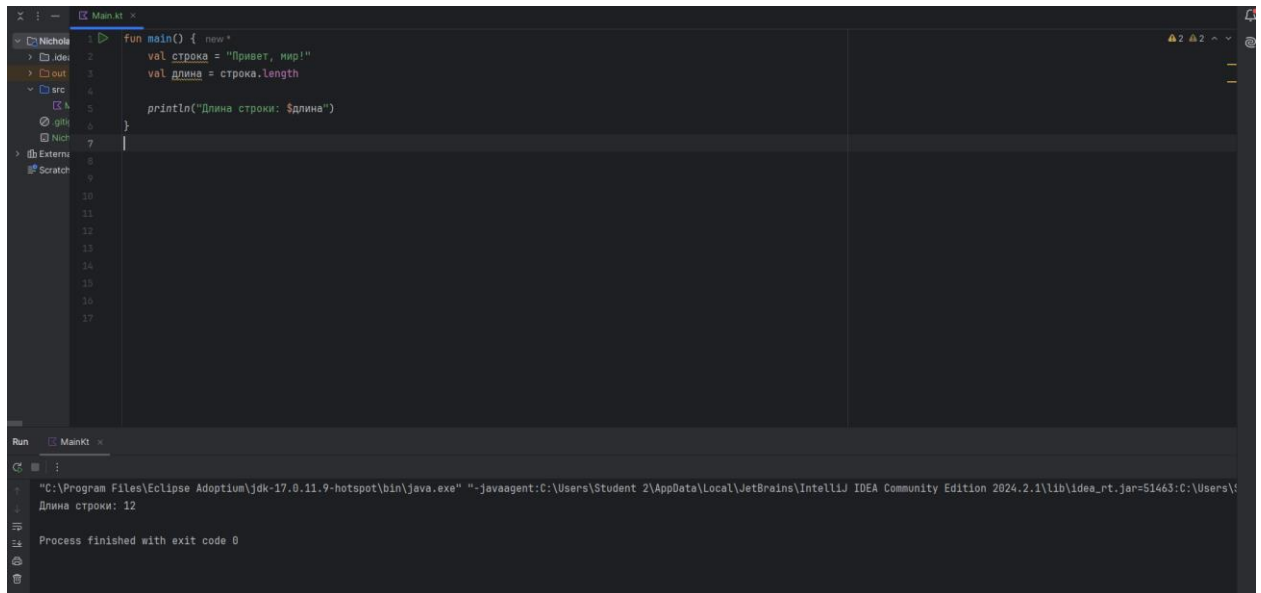
    val строка = "Привет, мир!"

    val длина = строка.length

    println("Длина строки: $длина")

}

```

```
1 fun main() { new +
2     val строка = "Привет, мир!"
3     val длина = строка.length
4
5     println("Длина строки: $длина")
6 }
7
8
9
10
11
12
13
14
15
16
17
```

Run MainKt

"C:\Program Files\Eclipse Adoptium\jdk-17.0.11.9-hotspot\bin\java.exe" "-javaagent:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=51463:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -Didea.config.path=C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\config -Didea.system.path=C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin -Didea.version=2024.2.1

Длина строки: 12

Process finished with exit code 0

10.

```
fun main() {

    println("Введите способ оплаты (наличные, кредитная карта, PayPal):")

    val способОплаты = readLine()

    when (способОплаты) {

        "наличные" -> println("Вы выбрали оплату наличными.")

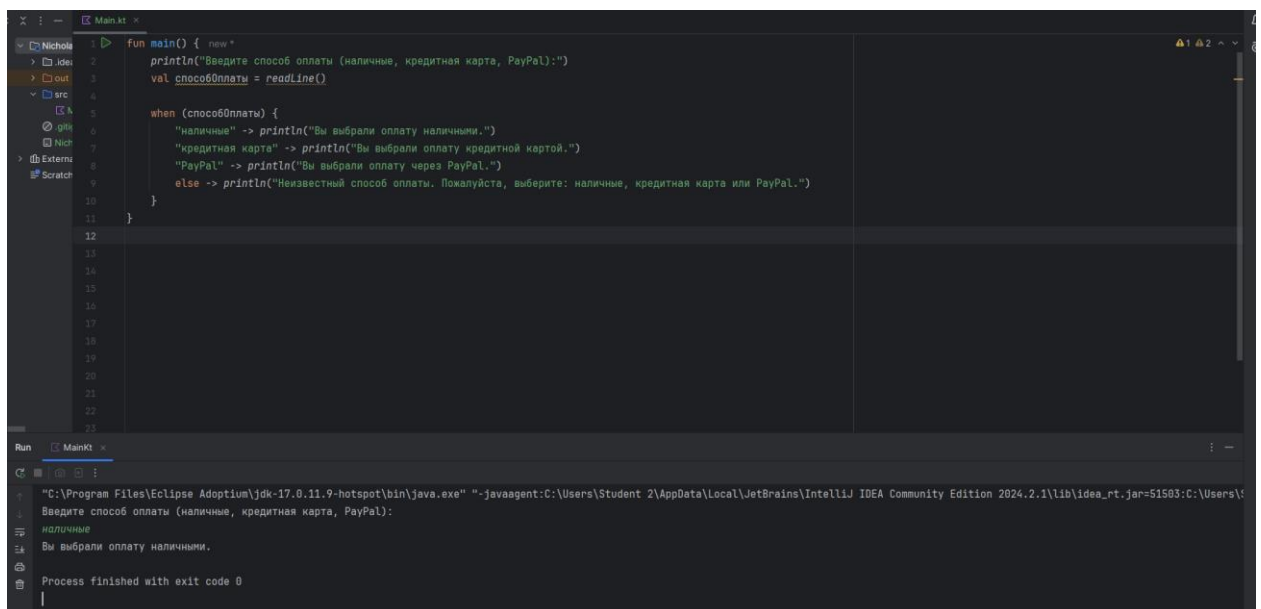
        "кредитная карта" -> println("Вы выбрали оплату кредитной картой.")

        "PayPal" -> println("Вы выбрали оплату через PayPal.")

        else -> println("Неизвестный способ оплаты. Пожалуйста, выберите: наличные, кредитная карта или PayPal.")

    }

}
```



```
1 fun main() { new +
2     println("Введите способ оплаты (наличные, кредитная карта, PayPal):")
3     val способОплаты = readLine()
4
5     when (способОплаты) {
6         "наличные" -> println("Вы выбрали оплату наличными.")
7         "кредитная карта" -> println("Вы выбрали оплату кредитной картой.")
8         "PayPal" -> println("Вы выбрали оплату через PayPal.")
9         else -> println("Неизвестный способ оплаты. Пожалуйста, выберите: наличные, кредитная карта или PayPal.")
10    }
11 }
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```

Run MainKt

"C:\Program Files\Eclipse Adoptium\jdk-17.0.11.9-hotspot\bin\java.exe" "-javaagent:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\lib\idea_rt.jar=51593:C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin" -Didea.config.path=C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\config -Didea.system.path=C:\Users\Student 2\AppData\Local\JetBrains\IntelliJ IDEA Community Edition 2024.2.1\bin -Didea.version=2024.2.1

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

Process finished with exit code 0

11.

```
fun main() {
```

```

println("Введите группу крови (A, B, AB, O):")

val группаКрови = readLine()?.toUpperCase()

when (группаКрови) {

    "A" -> println("Можно переливать: A, O")

    "B" -> println("Можно переливать: B, O")

    "AB" -> println("Можно переливать: A, B, AB, O")

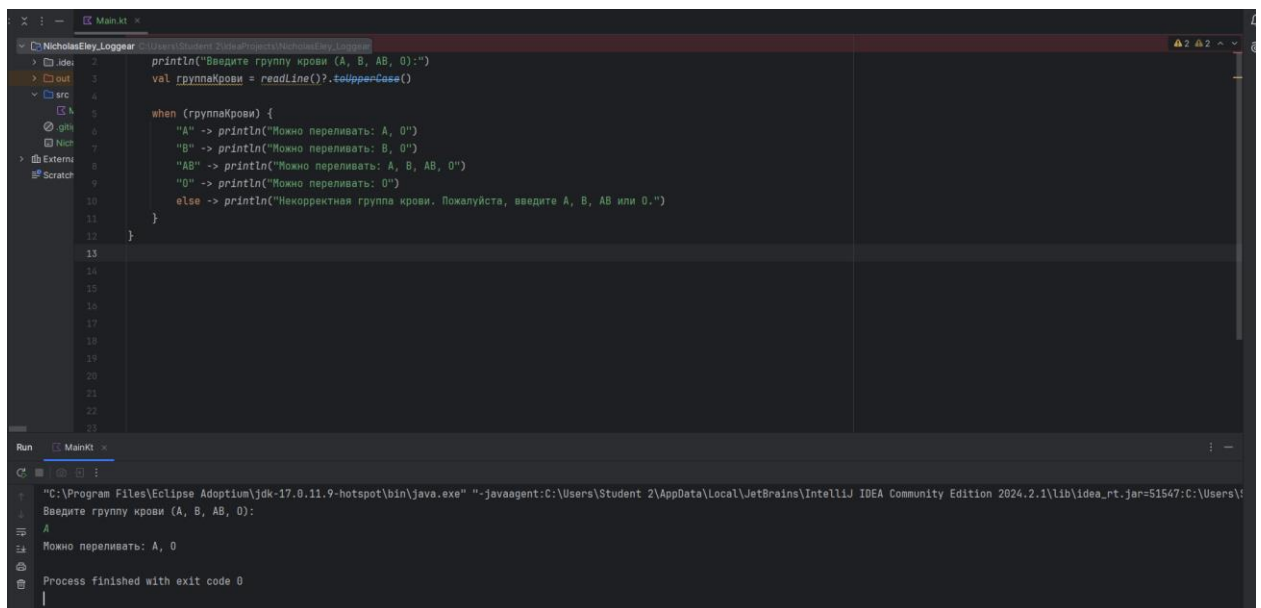
    "O" -> println("Можно переливать: O")

    else -> println("Некорректная группа крови. Пожалуйста, введите A, B, AB или O.")

}

}

```



12

```

fun main() {

    println("Введите название страны (США, Россия, Япония):")

    val страна = readLine()?.toLowerCase()

    when (страна) {

        "сша" -> println("Национальность: американская")

        "россия" -> println("Национальность: русская")

        "япония" -> println("Национальность: японская")

        "германия" -> println("Национальность: немецкая")

        "франция" -> println("Национальность: французская")

        else -> println("Неизвестная страна. Пожалуйста, введите одну из предложенных стран.")

    }

}

```

