

Практическая работа 4

Конструкция when

1. Определить день недели по номеру

```
fun main() {  
    val a = readLine()!!.toInt()  
    val b = arrayOf("Понедельник", "Вторник", "Среда", "Четверг", "Пятница",  
"Суббота", "Воскресенье")  
    println(if (a in 1..7) b[a - 1] else "Неверный номер дня")  
}
```

4

Четверг

2. Определить тип треугольника по длинам сторон

```
fun main() {  
    val a = readLine()!!.toInt()  
    val b = readLine()!!.toInt()  
    val c = readLine()!!.toInt()  
    println(when {  
        a == b && b == c -> "Равносторонний"  
        a == b || b == c || a == c -> "Равнобедренный"  
        else -> "Разносторонний"  
    })  
}
```

4	4	4
4	4	5
4	5	6
Равносторонний	Равнобедренный	Разносторонний

3. Вывод оценок по числовым значениям

```
fun main() {  
    val a = readLine()!!.toDouble()  
    println(when {  
        a >= 90 -> "Отлично"  
        a >= 75 -> "Хорошо"  
        a >= 60 -> "Удовлетворительно"  
        else -> "Неудовлетворительно"  
    })  
}
```

48	64	89	99
Неудовлетворительно	Удовлетворительно	Хорошо	Отлично

4. Определение времени суток

```
fun main() {  
    val a = readLine()!!.toInt()  
    println(when {
```

```

a in 6..11 -> "Утро"
a in 12..17 -> "День"
a in 18..23 -> "Вечер"
else -> "Ночь"
})
}

```

5	9	15	23
Ночь	Утро	День	Вечер

5. Определить знак числа

```

fun main() {
    val a = readLine()!!.toDouble()
    println(when {
        a > 0 -> "Положительное"
        a < 0 -> "Отрицательное"
        else -> "Ноль"
    })
}

```

30	-2	0
Положительное	Отрицательное	Ноль

6. Угадай число

```

fun main() {
    val a = (1..100).random()
    var b: Int
    do {
        println("Угадайте число от 1 до 100:")
        b = readLine()!!.toInt()
    } while (b != a)
    println("Вы угадали!")
}

```

Угадайте число от 1 до 100:

40

Угадайте число от 1 до 100:

41

Вы угадали!

7. Определение длины строки

```

fun main() {
    val a = readLine()!!
    println("Длина строки: ${a.length}")
}

```

aaaaaaaa

Длина строки: 7

8. Определить время приготовления по типу пищи

```
fun main() {
    val a = readLine()!!.toLowerCase()
    println(when (a) {
        "мясо" -> "Время приготовления: 1 час"
        "овощи" -> "Время приготовления: 30 минут"
        "десерт" -> "Время приготовления: 15 минут"
        else -> "Неизвестный тип пищи"
    })
}
```

рис

овощи

Неизвестный тип пищи

Время приготовления: 30 минут

9. Определение длины строки

```
fun main() {
    val a = readLine()!!
    println("Длина строки: ${a.length}")
}
```

aaaaaaa

Длина строки: 7

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

```
fun main() {
    val a = readLine()!!.toLowerCase()
    val b = mapOf(
        "налик" to "Оплата наличными",
        "карта" to "Оплата картой",
        "paypal" to "Оплата через PayPal"
    )
    println(b[a] ?: "Неверный способ оплаты")
}
```

налик

карта

paypal

Оплата наличными Оплата картой Оплата через PayPal

paypal

в РФ нельзя -_-

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

```
fun main() {
    val a = readLine()!!.toUpperCase()
    val b = mapOf(
        "A" to "A, AB",
        "B" to "B, AB",
        "AB" to "AB",
        "O" to "A, B, AB, O"
    )
}
```

```
)
println("Можно переливать: ${b[a] ?: "Неверная группа"}")
}
```

```
A      B
Можно переливать: A, AB  Можно переливать: B, AB
```

```
AB      0
Можно переливать: AB  Можно переливать: A, B, AB, 0
SS
Можно переливать: Неверная группа
```

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

```
fun main() {
    val a = readLine()!!.toLowerCase()
    val b = mapOf(
        "сша" to "Американец",
        "россия" to "Русский",
        "япония" to "Японец"
    )
    println(b[a] ?: "Неизвестная страна")
}
```

```
сша      россия      япония      Украина
Американец  Русский  Японец  Неизвестная страна
```

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() {
    val a = readLine()!!.toInt()
    val b = mapOf(
        100 to "Ошибка сети",
        200 to "Ошибка сервера",
        300 to "Ошибка базы данных"
    )
    println(b[a] ?: "Неверный код ошибки")
}
```

```
100      200      300
Ошибка сети  Ошибка сервера  Ошибка базы данных
400
Неверный код ошибки
```