

Конструкция when

Задача 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 недели" // обработка неправильного  
ввода  
    }  
}  
  
fun main() {  
    println("Введите номер дня недели (1-7):")  
    val dayNumber = readLine()?.toIntOrNull() // считываем номер дня  
    if (dayNumber != null) {  
        val day = getDayOfWeek(dayNumber)  
        println("День недели: $day")  
    } else {  
        println("Введите корректное число.")  
    }  
}
```

Ответ:

Введите номер дня недели (1-7):

3

День недели: Среда

Process finished with exit code 0

Задача 2

```
fun main() {  
    // Ввод длин сторон треугольника  
    println("Введите длину первой стороны:")  
    val a = readLine()!!.toDouble()  
    println("Введите длину второй стороны:")  
    val b = readLine()!!.toDouble()  
    println("Введите длину третьей стороны:")  
    val c = readLine()!!.toDouble()  
  
    // Определение типа треугольника  
    when {  
        a <= 0 || b <= 0 || c <= 0 -> println("Длины сторон должны быть  
положительными")  
        a == b && b == c -> println("Треугольник равносторонний")  
        a == b || b == c || a == c -> println("Треугольник равнобедренный")  
        else -> println("Треугольник разносторонний")  
    }  
}
```

Ответ:

Введите длину первой стороны:

23

Введите длину второй стороны:

35

Введите длину третьей стороны:

45

Треугольник разносторонний

Process finished with exit code 0

Задача 3

```
fun main() {
    println("Введите числовое значение от 0 до 100:")
    val input = readLine()

    if (input != null) {
        val score = input.toIntOrNull() // Преобразуем введенную строку в
число

        if (score != null && score in 0..100) {
            val grade = getGrade(score)
            println("Ваше оценка: $grade")
        } else {
            println("Ошибка: введите число в диапазоне от 0 до 100.")
        }
    } else {
        println("Ошибка: введено некорректное значение.")
    }
}

fun getGrade(score: Int): String {
    return when {
        score >= 90 -> "A"
        score >= 80 -> "B"
        score >= 70 -> "C"
        score >= 60 -> "D"
        score >= 50 -> "E"
        else -> "F"
    }
}
```

Ответ:

Введите числовое значение от 0 до 100:

54

Ваше оценка: E

Process finished with exit code 0

Задача 4

```
import java.time.LocalDateTime

fun getTimeOfDay(): String {
    val now = LocalDateTime.now()
    return when {
        now.isBefore(LocalTime.NOON) -> "Утро"
        now.isBefore(LocalTime.of(18, 0)) -> "День"
        else -> "Вечер"
    }
}

fun main() {
    val timeOfDay = getTimeOfDay()
    println("Сейчас: $timeOfDay")
}
```

Ответ:

Сейчас: День

Process finished with exit code 0

Задача 5

```
fun main() {
    val number = -5 // Здесь вы можете заменить -5 на любое другое число

    val sign = when {
        number > 0 -> "Положительное число"
        number < 0 -> "Отрицательное число"
        else -> "Ноль"
    }

    println(sign)
}
```

Ответ:

Отрицательное число

Process finished with exit code 0

Задача 6

```
import java.util.Random
import java.util.Scanner

fun main() {
    val random = Random()
    val targetNumber = random.nextInt(1, 101) // Генерируем случайное число
    от 1 до 100
    var guess: Int? = null
    val scanner = Scanner(System.`in`)

    println("Угадайте число от 1 до 100!")

    while (guess != targetNumber) {
        print("Введите ваше число: ")
        guess = scanner.nextInt()

        when {
            guess < targetNumber -> println("Загаданное число больше.")
            guess > targetNumber -> println("Загаданное число меньше.")
            else -> println("Поздравляю! Вы угадали число $targetNumber!")
        }
    }

    scanner.close()
}
```

Ответ:

Угадайте число от 1 до 100!

Введите ваше число: 17

Поздравляю! Вы угадали число 17!

Process finished with exit code 0

Задача 7

```
fun main() {
    val inputString = "Привет, мир!" // Замените на вашу строку
    val length = inputString.length

    when (length) {
        0 -> println("Строка пустая.")
        in 1..10 -> println("Строка короткая.")
        in 11..20 -> println("Строка средней длины.")
        else -> println("Строка длинная.")
    }
}
```

Ответ:

Строка средней длины.

Process finished with exit code 0

Задача 8

```
fun getCookingTime(foodType: String): Int {
    return when (foodType) {
        "паста" -> 10 // время в минутах
        "рис" -> 20
        "мясо" -> 30
        "овощи" -> 15
        else -> throw IllegalArgumentException("Неизвестный тип пищи: $foodType")
    }
}

fun main() {
    val foodType = "паста"
    val cookingTime = getCookingTime(foodType)
    println("Время приготовления для $foodType: $cookingTime минут")
}
```

Ответ:

Время приготовления для пасты: 10 минут

Process finished with exit code 0

Задача 9

```
fun main() {
    val str = "Привет, мир!"
    val length = str.length

    when (length) {
        0 -> println("Строка пустая.")
        in 1..10 -> println("Строка короткая (длина $length).")
        in 11..20 -> println("Строка средней длины (длина $length).")
        else -> println("Строка длинная (длина $length).")
    }
}
```

Ответ:

Строка средней длины (длина 12).

Process finished with exit code 0

Задача 10

```
fun main() {  
    println("Введите способ оплаты (наличные, кредитная карта, Мир): ")  
  
    val paymentMethod = readLine() // Читаем способ оплаты от пользователя  
  
    // Используем конструкцию when для обработки введенного способа оплаты  
    when (paymentMethod) {  
        "наличные" -> println("Вы выбрали оплату наличными.")  
        "кредитная карта" -> println("Вы выбрали оплату кредитной картой.")  
        "Мир" -> println("Вы выбрали оплату картой Мир.")  
        else -> println("Неверный способ оплаты. Пожалуйста, попробуйте  
снова.")  
    }  
}
```

Ответ:

Введите способ оплаты (наличные, кредитная карта, Мир):

наличные

Вы выбрали оплату наличными.

Process finished with exit code 0

Задача 11

```
fun main() {  
    println("Введите группу крови (A, B, AB, O):")  
    val bloodType = readLine()?.uppercase()  
  
    when (bloodType) {  
        "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("Недопустимая группа крови")  
    }  
}
```

Ответ:

Введите группу крови (A, B, AB, O):

A

Можно переливать: A, O

Process finished with exit code 0

Задача 12

```
fun main() {
    println("Введите название страны:")
    val country = readLine()

    val nationality = when (country?.toLowerCase()) {
        "сша" -> "американец"
        "россия" -> "русский"
        "япония" -> "японец"
        "франция" -> "француз"
        "германия" -> "немец"
        "китай" -> "китаец"
        "индия" -> "индиец"
        "бразилия" -> "бразилец"
        "италия" -> "итальянец"
        "португалия" -> "португалец"
        else -> "неизвестная национальность"
    }

    println("Национальность для страны $country: $nationality")
}
```

Ответ:

Введите название страны:

США

Национальность для страны США: американец

Process finished with exit code 0

Задача 13

```
fun main() {
    // Пример кода ошибки
    val errorCode = 200 // Замените значение на 100, 200 или 300

    // Получаем сообщение об ошибке на основе кода
    val errorMessage = when (errorCode) {
        100 -> "Ошибка сети"
        200 -> "Ошибка сервера"
        300 -> "Ошибка клиента"
        else -> "Неизвестная ошибка"
    }

    // Выводим сообщение об ошибке
    println(errorMessage)
}
```

Ответ:

Ошибка сервера

Process finished with exit code 0