

Практическая 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 -> "Неверный номер дня недели"
    }
}

fun main() {
    println(getDayOfWeek(3))
    println(getDayOfWeek(8))
}
```

2.

```
fun getTriangleType(a: Double, b: Double, c: Double): String {
    return when {
        a == b && b == c -> "Равносторонний"
        a == b || a == c || b == c ->
            "Равнобедренный"
        else -> "Разносторонний"
    }
}

fun main() {
    println(getTriangleType(5.0, 5.0, 5.0))
    println(getTriangleType(5.0, 5.0, 7.0))
    println(getTriangleType(3.0, 4.0, 5.0))
}
```

3.

```
fun getGrade(score: Int): String {
    return when (score) {
        in 90..100 -> "5"
        in 70..89 -> "4"
        in 60..69 -> "3"
        else -> "2"
    }
}

fun main () {
    println(getGrade(95))
    println(getGrade(83))
    println(getGrade(44))
}
```

4.

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

fun main() {
    val hour = 14
    val timeOfDay = getTimeOfDay(hour)
    println("Время суток: $timeOfDay")
}
```

5.

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

fun main() {
    val number = -5
    val sign = getSign(number)
    println("Знак числа: $sign")
}
```

6.

```
import kotlin.random.Random

fun main() {

    val secretNumber = Random.nextInt(1, 101)
    var guessed = false
    println("Угадай число от 1 до 100:")
    while (!guessed) {
        val userInput =
            readLine()?.toIntOrNull()
        when {
            userInput == null ->
                println("Введите валидное число!")
            userInput < secretNumber ->
                println("Слишком маленькое число!")
            userInput > secretNumber ->
                println("Слишком большое число!")
            else -> {
                println("Поздравляем! Вы угадали число: $secretNumber")
                guessed = true
            }
        }
    }
}
```

7.

```
fun main() {
    println("Введите строку:")
    val input = readLine() ?: ""
    when (input.length) {

        0 -> println("Строка пуста.")
        in 1..10 -> println("Строка короткая (длина от 1 до 10 символов).")
        else -> println("Строка средней длины (длина от 11 до 20 символов).")
    }
}
```

8.

```
fun main() {
    println("Введите тип пищи(блюдо):")
    val foodType = readLine()?.toLowerCase() ?: ""
    val cookingTime = when (foodType)
    {
        "паста" -> "Время приготовления: 10 минут."
        "мясо" -> "Время приготовления: 30 минут."
        "овощи" -> "Время приготовления: 15 минут."
        "рыба" -> "Время приготовления: 20 минут."
        "суп" -> "Время приготовления: 40 минут."
        else -> "Неизвестный тип пищи. Попробуйте снова."
    }
    println(cookingTime)
}
```

9.

```
fun main() {
    println("Введите строку:")
    val inputString = readLine() ?: ""
    val lengthMessage = when(inputString.length) {
        0 -> "Строка пустая."
        in 1..10 -> "Строка короткая."
        in 11..20 -> "Строка средней длины."
        else -> "Строка длинная."
    }
    println(lengthMessage)
}
```

10.

```
fun main() {
    println("Выберите способ оплаты(наличные, кредитная карта, PayPal):")
    val paymentMethod = readLine() ?: ""
    val message = when(paymentMethod.toLowerCase()) {
        "наличные" -> "Вы выбрали оплату наличными."
        "кредитная карта" -> "Вы выбрали оплату кредитной картой."
        "paypal" -> "Вы выбрали оплату через PayPal."
        else -> "Неверный способ оплаты."
    }
    println(message)
}
```

11.

```
fun main() {
    println("Введите группу крови(A, B, AB, 0):")
    val bloodType = readLine()?.toUpperCase() ?: ""
    val compatibleBloodTypes = when(bloodType) {
        "A" -> "A, A0, AB"
        "B" -> "B, B0, AB"
        "AB" -> "AB, A, B, 0"
        "0" -> "A, B, AB, 0"
        else -> "Неверная группа крови."
    }
    println("Совместимые типы крови для переливания: $compatibleBloodTypes")
}
```

12.

```
fun main() {
    println("Введите страну (США, Россия, Япония и т.д.):")
    val country = readLine()?.toUpperCase() ?: ""
    val nationality = when (country) {
        "США" -> "Американец/Американка"
        "РОССИЯ" -> "Русский/Русская"
        "ЯПОНИЯ" -> "Японец/Японка"
        "ГЕРМАНИЯ" -> "Немец/Немка"
        "ФРАНЦИЯ" -> "Француз/Француженка"
        "ИСПАНИЯ" -> "Испанец/Испанка"
        else -> "Неизвестная национальность"
    }
    println("Национальность: $nationality")
}
```

13.

```
fun main() {
    println("Введите код ошибки (100, 200, 300):")
    val errorCode = readLine()?.toIntOrNull() ?: -1
    val errorMessage = when(errorCode) {
        100 -> "Ошибка сети"
        200 -> "Ошибка сервера"
        300 -> "Ошибка авторизации"
        else -> "Неизвестный код ошибки"
    }
    println("Сообщение об ошибке: $errorMessage")
}
```