

Практическая работа 4

Конструкция when

1.

Определить день недели по номеру

```
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() {  
    print("Введите номер дня(1-7): ")  
    val dayNumber = readLine()!!.toInt()  
    println(getDayOfWeek(dayNumber))  
}
```

```
Введите номер дня(1-7): 2  
Вторник
```

2.

Определить тип треугольника по длинам сторон

```
fun getTriangleType(a: Double, b: Double, c: Double):  
    String {return when {  
        a <= 0 || b <= 0 || c <= 0 -> "Некорректные значения сторон"  
        a + b <= c || b + c <= a || a + c <= b -> "Треугольник не существует"  
        a == b && b == c -> "Равносторонний треугольник"  
        a == b || b == c || a == c -> "Равнобедренный треугольник"  
        a*a + b*b == c*c || b*b + c*c == a*a || a*a + c*c == b*b -> "Прямоугольный  
треугольник"  
        else -> "Разносторонний треугольник"  
    }}  
fun main() {  
    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()  
    if (a != null && b != null && c != null) {  
        println(getTriangleType(a, b, c))  
    } else {  
        println("Некорректные значения сторон")  
    }  
}
```

```
Введите длину первой стороны:  
3  
Введите длину второй стороны:  
4  
Введите длину третьей стороны:  
5  
Прямоугольный треугольник
```

3.

Вывод оценок по числовым значениям

```
fun getGrade(score: Int): String {  
    return when(score) {  
        2 -> "Неудовлетворительно"  
        3 -> "Удовлетворительно"  
        4 -> "Хорошо"  
        5 -> "Отлично"  
        else -> "Некорректное значение"  
    }  
}  
  
fun main() {  
    println("Введите оценку")  
    val score = readLine()!!.toInt()  
    println(getGrade(score))  
}
```

Введите оценку

2

Неудовлетворительно

Введите оценку

4

Хорошо

4.

Определение времени суток

```
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 -> "Некорректное время"  
    }  
}  
  
fun main() {  
    println("Введите текущий час: ")  
    val hour = readLine()!!.toInt()  
    println(getTimeOfDay(hour))  
}
```

Введите текущий час:

18

Вечер

5.

Определить знак числа

```
fun getNumberSign(number: Int): String {  
    return when {  
        number > 0 -> "Положительное"  
        number < 0 -> "Отрицательное"  
        else -> "Ноль"  
    }  
}  
  
fun main() {  
    println("Введите число: ")  
    val number = readLine()!!.toInt()  
    println(getNumberSign(number))  
}
```

Введите число:

5

Положительное

6.

Угадай число

```
fun main() {  
    val number = (1..100).random()  
  
    while (true) {  
        println("Угадайте число от 1 до 100:")  
        val guess = readLine()?.toIntOrNull() ?: continue  
  
        when {  
            guess < number -> println("Больше")  
            guess > number -> println("Меньше")  
            else -> {  
                println("Угадали!")  
                break  
            }  
        }  
    }  
}
```

```
Угадайте число от 1 до 100:  
50  
Меньше  
Угадайте число от 1 до 100:  
40  
Меньше  
Угадайте число от 1 до 100:  
30  
Меньше  
Угадайте число от 1 до 100:  
20  
Больше  
Угадайте число от 1 до 100:  
25  
Меньше  
Угадайте число от 1 до 100:  
24  
Меньше  
Угадайте число от 1 до 100:  
23  
Угадали!
```

7.

Определение длины строки

```
fun main() {  
    println("Введите строку:")  
    val text = readLine()  
    println("Длина строки: ${text?.length ?: 0}")  
}
```

```
Введите строку:  
Привет, как дела?  
Длина строки: 17
```

```

fun getCookingInstructions(foodType: String): String {
    return when(foodType.lowercase()) {
        "яйца" -> """
            Время приготовления яиц:
            - Всмятку: 3 минуты
            - В мешочек: 5 минут
            - Вкрутую: 7-8 минут
            Инструкция:
            1. Вскипятите воду
            2. Аккуратно опустите яйца в кипящую воду
            3. Засеките время
            4. После готовности поместите в холодную воду
            """"

        "макароны" -> """
            Время приготовления макарон:
            - Спагетти: 8-10 минут
            - Пенне: 11-13 минут
            - Фетучини: 10-12 минут
            Инструкция:
            1. Вскипятите воду (1л на 100г макарон)
            2. Добавьте соль (1 чайная ложка на литр)
            3. Опустите макароны в воду
            4. Периодически помешивайте
            5. Проверьте готовность за минуту до конца
            """"

        "рис" -> """
            Время приготовления риса:
            - Белый рис: 15-20 минут
            - Бурый рис: 40-45 минут
            - Дикий рис: 45-50 минут
            Инструкция:
            1. Промойте рис до прозрачной воды
            2. Залейте водой (2 части воды на 1 часть риса)
            3. Доведите до кипения
            4. Уменьшите огонь, накройте крышкой
            5. Не открывайте крышку во время приготовления
            """"

        "мясо" -> """
            Время приготовления мяса:
            - Стейк (средней прожарки): 4-5 минут с каждой стороны
            - Свинина: 30-40 минут
            - Говядина (тушение): 1.5-2 часа
            Инструкция:
            1. Достаньте мясо за час до готовки
            2. Посолите и поперчите
            3. Разогрейте сковороду/духовку
            4. Для стейка: обжарьте на сильном огне
            5. Для тушения: обжарьте, затем тушите на медленном огне
            """"

        "курица" -> """
            Время приготовления курицы:
            - Филе: 15-20 минут
            - Окорочка: 25-30 минут
            - Целая курица: 60-90 минут
            Инструкция:
            1. Промойте и обсушите мясо
            2. Замаринуйте (по желанию)
            3. Филе: нарежьте одинаковыми кусками
            4. Приправьте специями
            5. Готовьте на среднем огне
            6. Проверяйте готовность разрезом
            """"
    }
}

```

```

"""

"рыба" -> """
    Время приготовления рыбы:
    - Филе: 10-15 минут
    - Стейк: 15-20 минут
    - Целая рыба: 20-25 минут
    Инструкция:
    1. Проверьте рыбу на наличие костей
    2. Сбрызните лимонным соком
    3. Приправьте солью и специями
    4. Готовьте на среднем огне
    5. Рыба готова, когда мясо стало непрозрачным
    6. Не пересушите!
"""

"суп" -> """
    Время приготовления супа:
    - Овощной: 30-40 минут
    - Куриный: 40-50 минут
    - Борщ: 1.5-2 часа
    Инструкция:
    1. Подготовьте бульон (если требуется)
    2. Нарезьте все ингредиенты
    3. Соблюдайте порядок закладки:
        - Сначала твёрдые овощи
        - Затем мягкие
        - В конце зелень
    4. Варите на медленном огне
    5. Проверяйте готовность овощей
"""

"овощи" -> """
    Время приготовления овощей:
    - На пару: 10-15 минут
    - Варка: 12-20 минут
    - Запекание: 20-30 минут
    Инструкция:
    1. Помойте и почистите овощи
    2. Нарезьте одинаковыми кусочками
    3. Время приготовления:
        - Морковь: 15-20 минут
        - Брокколи: 5-7 минут
        - Картофель: 20-25 минут
        - Цветная капуста: 10-12 минут
    4. Не переваривайте, сохраняйте хрустящность
"""

else -> "Неизвестное блюдо. Пожалуйста, выберите из списка: яйца, макароны,
рис, мясо, курица, рыба, суп, овощи"
}
}

fun main() {
    println("Введите тип блюда (яйца, макароны, рис, мясо, курица, рыба, суп, овощи):")
    val food = readLine()
    println(getCookingInstructions(food ?: ""))
}

Введите тип блюда (яйца, макароны, рис, мясо, курица, рыба, суп, овощи):
рис

Время приготовления риса:
- Белый рис: 15-20 минут
- Бурый рис: 40-45 минут
- Дикий рис: 45-50 минут
Инструкция:
1. Промойте рис до прозрачной воды
2. Залейте водой (2 части воды на 1 часть риса)
3. Доведите до кипения
4. Уменьшите огонь, накройте крышкой
5. Не открывайте крышку во время приготовления

```

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

```
fun processPayment(paymentMethod: String): String {
    return when(paymentMethod.lowercase()) {
        "наличные" -> """
            Оплата наличными:
            1. Сумма к оплате будет указана на кассе
            2. Подготовьте точную сумму или близкую к ней
            3. Получите сдачу и кассовый чек
            4. Проверьте правильность суммы и сдачи
            Комиссия: нет
            Время обработки: моментально
        """

        "кредитная карта", "карта" -> """
            Оплата картой:
            1. Вставьте карту в терминал или приложите к считывателю
            2. Выберите способ оплаты (кредит/дебет)
            3. Введите PIN-код при необходимости
            4. Дождитесь подтверждения операции
            5. Сохраните чек
            Комиссия: согласно условиям банка
            Время обработки: 1-2 минуты
        """

        "paypal" -> """
            Оплата через PayPal:
            1. Войдите в свой аккаунт PayPal
            2. Проверьте email получателя платежа
            3. Укажите сумму перевода
            4. Выберите источник средств
            5. Подтвердите транзакцию
            Комиссия: 0-2% в зависимости от типа операции
            Время обработки: 1-24 часа
        """

        "сбп", "система быстрых платежей" -> """
            Оплата через СБП:
            1. Откройте банковское приложение
            2. Выберите оплату по QR-коду
            3. Отсканируйте QR-код продавца
            4. Проверьте детали платежа
            5. Подтвердите операцию
            Комиссия: до 0.7%
            Время обработки: моментально
            Лимиты: до 1 млн руб в день
        """

        else -> """
            Доступные способы оплаты:
            1. Наличные
            2. Кредитная карта
            3. PayPal
            4. СБП (Система быстрых платежей)

            Пожалуйста, выберите один из указанных способов оплаты.
        """
    }
}

fun main() {
    println("Выберите способ оплаты (наличные, кредитная карта, PayPal, СБП):")
    val paymentMethod = readLine()
    println(processPayment(paymentMethod ?: ""))
}
```

Выберите способ оплаты (наличные, кредитная карта, PayPal, СБП):

наличные

Оплата наличными:

1. Сумма к оплате будет указана на кассе
2. Подготовьте точную сумму или близкую к ней
3. Получите сдачу и кассовый чек
4. Проверьте правильность суммы и сдачи

Комиссия: нет

Время обработки: моментально

10.

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

```
fun getBloodTypeCompatibility(bloodType: String): String {
    return when (bloodType.uppercase()) {
        "A", "A" -> ""
            Группа крови: A (II)

            Может получать кровь от:
            ✓ A (II)
            ✓ O (I)

            Может отдавать кровь:
            ✓ A (II)
            ✓ AB (IV)

            Резус-фактор:
            • Если у вас A+: можете получать A+, A-, O+, O-
            • Если у вас A-: можете получать только A-, O-
            ""

        "B", "B" -> ""
            Группа крови: B (III)

            Может получать кровь от:
            ✓ B (III)
            ✓ O (I)

            Может отдавать кровь:
            ✓ B (III)
            ✓ AB (IV)

            Резус-фактор:
            • Если у вас B+: можете получать B+, B-, O+, O-
            • Если у вас B-: можете получать только B-, O-
            ""

        "AB", "AB" -> ""
            Группа крови: AB (IV)

            Может получать кровь от:
            ✓ AB (IV)
            ✓ A (II)
            ✓ B (III)
            ✓ O (I)

            Может отдавать кровь:
```

✓ Только АВ (IV)

Резус-фактор:

- Если у вас АВ+: можете получать кровь от всех групп
- Если у вас АВ-: можете получать кровь от всех отрицательных групп

Примечание: Универсальный реципиент

"""

"O", "0", "0" -> ""

Группа крови: O (I)

Может получать кровь от:

✓ Только O (I)

Может отдавать кровь:

✓ O (I)

✓ A (II)

✓ B (III)

✓ АВ (IV)

Резус-фактор:

- Если у вас O+: можете получать O+, O-
- Если у вас O-: можете получать только O-

Примечание: Универсальный донор

"""

else -> ""

Ошибка ввода!

Пожалуйста, введите корректную группу крови:

- A (или A)
- B (или B)
- АВ (или АВ)
- O (или O, 0)

Важно: При переливании крови также учитывается резус-фактор!

"""

}

}

```
fun main() {  
    println("Введите группу крови (A, B, AB, O):")  
    val bloodType = readLine()  
    println(getBloodTypeCompatibility(bloodType ?: ""))  
}
```

Введите группу крови (A, B, AB, O):

A

Группа крови: A (II)

Может получать кровь от:

✓ A (II)

✓ O (I)

Может отдавать кровь:

✓ A (II)

✓ АВ (IV)

Резус-фактор:

- Если у вас A+: можете получать A+, A-, O+, O-
- Если у вас A-: можете получать только A-, O-

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

```
fun getNationalityInfo(country: String): String {
    return when(country.lowercase()) {
        "россия" -> "Национальность: русские\nЯзык: русский\nПрилагательное:
российский"
        "сша", "соединенные штаты америки" -> "Национальность: американцы\nЯзык:
английский\nПрилагательное: американский"
        "япония" -> "Национальность: японцы\nЯзык: японский\nПрилагательное: японский"
        "китай" -> "Национальность: китайцы\nЯзык: китайский\nПрилагательное:
китайский"
        "индия" -> "Национальность: индийцы\nЯзык: хинди, английский\nПрилагательное:
индийский"
        "германия" -> "Национальность: немцы\nЯзык: немецкий\nПрилагательное: немецкий"
        "франция" -> "Национальность: французы\nЯзык: французский\nПрилагательное:
французский"
        "великобритания" -> "Национальность: британцы\nЯзык:
английский\nПрилагательное: британский"
        "италия" -> "Национальность: итальянцы\nЯзык: итальянский\nПрилагательное:
итальянский"
        "испания" -> "Национальность: испанцы\nЯзык: испанский\nПрилагательное:
испанский"
        "португалия" -> "Национальность: португальцы\nЯзык:
португальский\nПрилагательное: португальский"
        "греция" -> "Национальность: греки\nЯзык: греческий\nПрилагательное: греческий"
        "польша" -> "Национальность: поляки\nЯзык: польский\nПрилагательное: польский"
        "украина" -> "Национальность: украинцы\nЯзык: украинский\nПрилагательное:
украинский"
        "белоруссия" -> "Национальность: белорусы\nЯзык: белорусский,
русский\nПрилагательное: белорусский"
        "швеция" -> "Национальность: шведы\nЯзык: шведский\nПрилагательное: шведский"
        "норвегия" -> "Национальность: норвежцы\nЯзык: норвежский\nПрилагательное:
норвежский"
        "финляндия" -> "Национальность: финны\nЯзык: финский, шведский\nПрилагательное:
финский"
        "дания" -> "Национальность: датчане\nЯзык: датский\nПрилагательное: датский"
        "нидерланды" -> "Национальность: голландцы\nЯзык:
нидерландский\nПрилагательное: нидерландский"
        "бельгия" -> "Национальность: бельгийцы\nЯзык: нидерландский, французский,
немецкий\nПрилагательное: бельгийский"
        "швейцария" -> "Национальность: швейцарцы\nЯзык: немецкий, французский,
итальянский, романшский\nПрилагательное: швейцарский"
        "австрия" -> "Национальность: австрийцы\nЯзык: немецкий\nПрилагательное:
австрийский"
        "чехия" -> "Национальность: чехи\nЯзык: чешский\nПрилагательное: чешский"
        "словакия" -> "Национальность: словаки\nЯзык: словацкий\nПрилагательное:
словацкий"
        "венгрия" -> "Национальность: венгры\nЯзык: венгерский\nПрилагательное:
венгерский"
        "румыния" -> "Национальность: румыны\nЯзык: румынский\nПрилагательное:
румынский"
        "болгария" -> "Национальность: болгары\nЯзык: болгарский\nПрилагательное:
болгарский"
        "сербия" -> "Национальность: сербы\nЯзык: сербский\nПрилагательное: сербский"
        "хорватия" -> "Национальность: хорваты\nЯзык: хорватский\nПрилагательное:
хорватский"
        "словения" -> "Национальность: словенцы\nЯзык: словенский\nПрилагательное:
словенский"
        "албания" -> "Национальность: албанцы\nЯзык: албанский\nПрилагательное:
албанский"
        "турция" -> "Национальность: турки\nЯзык: турецкий\nПрилагательное: турецкий"
        "израиль" -> "Национальность: израильтяне\nЯзык: иврит,
арабский\nПрилагательное: израильский"
        "иран" -> "Национальность: иранцы\nЯзык: персидский\nПрилагательное: иранский"
```

"ирак" -> "Национальность: иракцы\nЯзык: арабский, курдский\nПрилагательное: иракский"

"саудовская аравия" -> "Национальность: саудовцы\nЯзык: арабский\nПрилагательное: саудовский"

"египет" -> "Национальность: египтяне\nЯзык: арабский\nПрилагательное: египетский"

"марокко" -> "Национальность: марокканцы\nЯзык: арабский, берберский\nПрилагательное: марокканский"

"алжир" -> "Национальность: алжирцы\nЯзык: арабский, берберский\nПрилагательное: алжирский"

"тунис" -> "Национальность: туниисцы\nЯзык: арабский\nПрилагательное: туниисский"

"ливия" -> "Национальность: ливийцы\nЯзык: арабский\nПрилагательное: ливийский"

"судан" -> "Национальность: суданцы\nЯзык: арабский, английский\nПрилагательное: суданский"

"эфиопия" -> "Национальность: эфиопы\nЯзык: амхарский\nПрилагательное: эфиопский"

"кения" -> "Национальность: кенийцы\nЯзык: суахили, английский\nПрилагательное: кенийский"

"танзания" -> "Национальность: танзанийцы\nЯзык: суахили, английский\nПрилагательное: танзанийский"

"уганда" -> "Национальность: угандийцы\nЯзык: английский, суахили\nПрилагательное: угандийский"

"нигерия" -> "Национальность: нигерийцы\nЯзык: английский\nПрилагательное: нигерийский"

"гана" -> "Национальность: ганцы\nЯзык: английский\nПрилагательное: ганский"

"сенегал" -> "Национальность: сенегальцы\nЯзык: французский\nПрилагательное: сенегальский"

"юар" -> "Национальность: южноафриканцы\nЯзык: африкаанс, английский\nПрилагательное: южноафриканский"

"казахстан" -> "Национальность: казахи\nЯзык: казахский, русский\nПрилагательное: казахстанский"

"узбекистан" -> "Национальность: узбеки\nЯзык: узбекский\nПрилагательное: узбекский"

"таджикистан" -> "Национальность: таджики\nЯзык: таджикский\nПрилагательное: таджикский"

"киргизия" -> "Национальность: киргизы\nЯзык: киргизский, русский\nПрилагательное: киргизский"

"туркмения" -> "Национальность: туркмены\nЯзык: туркменский\nПрилагательное: туркменский"

"монголия" -> "Национальность: монголы\nЯзык: монгольский\nПрилагательное: монгольский"

"южная корея" -> "Национальность: корейцы\nЯзык: корейский\nПрилагательное: южнокорейский"

"северная корея" -> "Национальность: корейцы\nЯзык: корейский\nПрилагательное: северокорейский"

"вьетнам" -> "Национальность: вьетнамцы\nЯзык: вьетнамский\nПрилагательное: вьетнамский"

"таиланд" -> "Национальность: тайцы\nЯзык: тайский\nПрилагательное: тайский"

"камбоджа" -> "Национальность: кхмеры\nЯзык: кхмерский\nПрилагательное: камбоджийский"

"малайзия" -> "Национальность: малайзийцы\nЯзык: малайский\nПрилагательное: малайзийский"

"индонезия" -> "Национальность: индонезийцы\nЯзык: индонезийский\nПрилагательное: индонезийский"

"филиппины" -> "Национальность: филиппинцы\nЯзык: филиппинский, английский\nПрилагательное: филиппинский"

"австралия" -> "Национальность: австралийцы\nЯзык: английский\nПрилагательное: австралийский"

"новая зеландия" -> "Национальность: новозеландцы\nЯзык: английский, маори\nПрилагательное: новозеландский"

"канада" -> "Национальность: канадцы\nЯзык: английский, французский\nПрилагательное: канадский"

"мексика" -> "Национальность: мексиканцы\nЯзык: испанский\nПрилагательное: мексиканский"

"бразилия" -> "Национальность: бразильцы\nЯзык: португальский\nПрилагательное: бразильский"

"аргентина" -> "Национальность: аргентинцы\nЯзык: испанский\nПрилагательное: аргентинский"

```

аргентинский"
    "чили" -> "Национальность: чилийцы\nЯзык: испанский\nПрилагательное: чилийский"
    "перу" -> "Национальность: перуанцы\nЯзык: испанский\nПрилагательное:
перуанский"
    "колумбия" -> "Национальность: колумбийцы\nЯзык: испанский\nПрилагательное:
колумбийский"
    "венесуэла" -> "Национальность: венесуэльцы\nЯзык: испанский\nПрилагательное:
венесуэльский"
    "боливия" -> "Национальность: боливийцы\nЯзык: испанский\nПрилагательное:
боливийский"
    "эквадор" -> "Национальность: эквадорцы\nЯзык: испанский\nПрилагательное:
эквадорский"
    "парагвай" -> "Национальность: парагвайцы\nЯзык: испанский,
гуарани\nПрилагательное: парагвайский"
    "уругвай" -> "Национальность: уругвайцы\nЯзык: испанский\nПрилагательное:
уругвайский"
    else -> "Страна не найдена. Пожалуйста, проверьте правильность написания."
}
}
fun main() {
    println("Введите название страны:")
    val country = readLine()
    println(getNationalityInfo(country ?: ""))
}

```

```

Введите название страны:
албания
Национальность: албанцы
Язык: албанский
Прилагательное: албанский

```

12.

Коды ошибок: Программа принимает код ошибки (100, 200, 300) и выводит сообщение об ошибке (например, "Ошибка сети", "Ошибка сервера" и т.д.).

```

fun getErrorDescription(errorCode: Int): String {
    return when(errorCode) {
        // 1xx - Информационные
        100 -> "100 Continue: Продолжение запроса"
        101 -> "101 Switching Protocols: Переключение протоколов"
        102 -> "102 Processing: Идёт обработка"
        103 -> "103 Early Hints: Ранние подсказки"

        // 2xx - Успешные
        200 -> "200 OK: Успешный запрос"
        201 -> "201 Created: Ресурс создан"
        202 -> "202 Accepted: Запрос принят"
        203 -> "203 Non-Authoritative Information: Информация не авторитетна"
        204 -> "204 No Content: Нет содержимого"
        205 -> "205 Reset Content: Сбросить содержимое"
        206 -> "206 Partial Content: Частичное содержимое"
        207 -> "207 Multi-Status: Многостатусный"

        // 3xx - Перенаправления
        300 -> "300 Multiple Choices: Множественный выбор"
        301 -> "301 Moved Permanently: Ресурс перемещён навсегда"
        302 -> "302 Found: Ресурс временно перемещён"
        303 -> "303 See Other: Смотреть другое"
        304 -> "304 Not Modified: Не изменялось"
        305 -> "305 Use Proxy: Использовать прокси"
        307 -> "307 Temporary Redirect: Временное перенаправление"
        308 -> "308 Permanent Redirect: Постоянное перенаправление"

        // 4xx - Ошибки клиента
        400 -> "400 Bad Request: Некорректный запрос"
    }
}

```

```

401 -> "401 Unauthorized: Не авторизован"
402 -> "402 Payment Required: Требуется оплата"
403 -> "403 Forbidden: Доступ запрещён"
404 -> "404 Not Found: Ресурс не найден"
405 -> "405 Method Not Allowed: Метод не поддерживается"
406 -> "406 Not Acceptable: Неприемлемый запрос"
407 -> "407 Proxy Authentication Required: Требуется аутентификация прокси"
408 -> "408 Request Timeout: Истекло время ожидания"
409 -> "409 Conflict: Конфликт"
410 -> "410 Gone: Ресурс удалён"
411 -> "411 Length Required: Необходима длина"
412 -> "412 Precondition Failed: Условие ложно"
413 -> "413 Payload Too Large: Полезная нагрузка слишком велика"
414 -> "414 URI Too Long: URI слишком длинный"
415 -> "415 Unsupported Media Type: Неподдерживаемый тип данных"
416 -> "416 Range Not Satisfiable: Диапазон не достижим"
417 -> "417 Expectation Failed: Ожидание не удалось"
418 -> "418 I'm a teapot: Я - чайник"
429 -> "429 Too Many Requests: Слишком много запросов"

// 5xx - Ошибки сервера
500 -> "500 Internal Server Error: Внутренняя ошибка сервера"
501 -> "501 Not Implemented: Не реализовано"
502 -> "502 Bad Gateway: Плохой шлюз"
503 -> "503 Service Unavailable: Сервис недоступен"
504 -> "504 Gateway Timeout: Шлюз не отвечает"
505 -> "505 HTTP Version Not Supported: Версия HTTP не поддерживается"
506 -> "506 Variant Also Negotiates: Вариант также проводит согласование"
507 -> "507 Insufficient Storage: Недостаточно места"
508 -> "508 Loop Detected: Обнаружено бесконечное перенаправление"
510 -> "510 Not Extended: Не расширено"
511 -> "511 Network Authentication Required: Требуется сетевая аутентификация"

else -> ""
    Код ошибки не найден!

    Основные группы кодов:
    1xx - Информационные
    2xx - Успешные
    3xx - Перенаправления
    4xx - Ошибки клиента
    5xx - Ошибки сервера

    Пожалуйста, введите корректный код ошибки.
    ""
}
}

```

```

fun main() {
    println("Введите код ошибки: ")
    val errorCode = readLine()?.toIntOrNull()
    if (errorCode != null) {
        println(getErrorDescription(errorCode))
    } else {
        println("Пожалуйста, введите числовой код ошибки")
    }
}

```

Введите код ошибки:

503

503 Service Unavailable: Сервис недоступен