Раздел 5. Циклы

Циклы в Go - это инструменты, которые позволяют программистам выполнять повторяющиеся действия в течение определенного количества раз. В Go, циклы выполняются при помощи ключевых слов `for`, `range`.

Цикл for

Цикл `for` используется для повторения блока кода заданное количество раз. Вот пример использования цикла `for`, который нам уже хорошо знаком :

```
for i := 0; i < 5; i++ {
    fmt.Println(i)
}
```

В этом примере мы повторяем блок кода пять раз, начиная с 0 и заканчивая 4.

Цикл range

Цикл `range` используется для повторения блока кода для каждого элемента в массиве, срезе или карте. Вот пример использования цикла `range`:

```
numbers := []int{1, 2, 3, 4, 5}
for _, number := range numbers {
    fmt.Println(number)
}
```

В этом примере мы повторяем блок кода для каждого элемента в срезе `numbers`, который содержит числа от 1 до 5.

Ключевое слово break

Ключевое слово `break` используется для прерывания выполнения цикла. Вот пример использования ключевого слова `break`:

```
for i := 0; i < 5; i++ {
    fmt.Println(i)
    break
}
```

В этом примере мы повторяем блок кода четыре раза, начиная с 0 и заканчивая 4. Однако мы прерываемся после первой же итерации, из-за наличия инструкции `break`, в будущем мы научимся использовать этот функционал полезнее, через ряд новых инструкций.

Ключевое слово continue

Ключевое слово `continue` используется для пропуска текущей итерации цикла и перехода к следующей. Вот пример использования ключевого слова `continue`:

```
for i := 0; i < 5; i++ {
    if i == 2 {
        continue
    }
    fmt.Println(i)
}
```

В этом примере мы повторяем блок кода пять раз, начиная с 0 и заканчивая 4. Однако мы пропускаем итерацию, когда `i` равно 2, из-за наличия инструкции `continue`. В результате на экран выводятся числа от 0 до 4, кроме 2.

Пример использования циклов

Вот пример использования циклов для разных коллекций:

```
package main

import "fmt"

func main() {
```

```
    var arr [5]int = [5]int{1, 2, 3, 4, 5}

    for i, v := range arr {
        fmt.Printf("Индекс: %d, Значение: %d\n", i, v)
    }
}
```

```
package main

import "fmt"

func main() {
    slice := []string{"Go", "C++", "Python", "Java", "JavaScript

    for i, v := range slice {
        fmt.Printf("Индекс: %d, Значение: %s\n", i, v)
    }
}
```

```
package main

import "fmt"

func main() {
    m := map[string]int{
        "Apple":  5,
        "Banana": 8,
        "Cherry": 3,
    }

    for k, v := range m {
        fmt.Printf("Ключ: %s, Значение: %d\n", k, v)
    }
}
```

Циклы в Go - это инструменты, которые позволяют программистам выполнять повторяющиеся действия в течение определенного количества раз. Они позволяют повторять блок кода заданное количество раз или для каждого элемента в массиве, срезе или карте. Ключевые слова `for`, `range` и `break` используются для создания циклов и управления их выполнением.

Домашнее задание

1. **Простой цикл for**: Напишите цикл for, который выводит на экран числа от 1 до 10.
2. **Цикл for с условием**: Напишите цикл for, который выводит на экран все четные числа от 1 до 20.
3. **Цикл for с continue**: Модифицируйте предыдущий цикл так, чтобы он использовал команду continue для пропуска нечетных чисел.
4. **Цикл for с break**: Напишите цикл for, который выводит числа от 1 до 100, но прерывает цикл, как только сумма выводимых чисел превысит 50.
5. **Цикл for-range с массивом**: Создайте массив с несколькими элементами. Напишите цикл for-range, который выводит каждый элемент массива на экран.
6. **Цикл for-range с map**: Создайте map с несколькими парами ключ-значение. Напишите цикл for-range, который выводит каждую пару ключ-значение на экран.
7. **Цикл for-range с слайсом**: Создайте слайс с несколькими элементами. Напишите цикл for-range, который выводит каждый элемент слайса на экран.
8. **Цикл for-range с строкой**: Создайте строку. Напишите цикл for-range, который выводит каждый символ строки на экран.
9. **Наполнение массива:** Создайте массив типа int с 5 элементами, с помощью цикла запишите в массив значения, которые будут равняться индексу, по которому значение будет находиться
10. **Цикл в цикле:** Используя два цикла выведите таблицу умножения.

11. **Одинаковый элемент в слайсе:** Создайте два слайса типа int со значениями 1, 2, 6, 5, 8 и 9, 5, 3, 4, 1 соответственно. С помощью циклов выведите значения, которые повторяются в слайсах. Попробуйте слайсы с разными значениями и длиной для такого же итога.