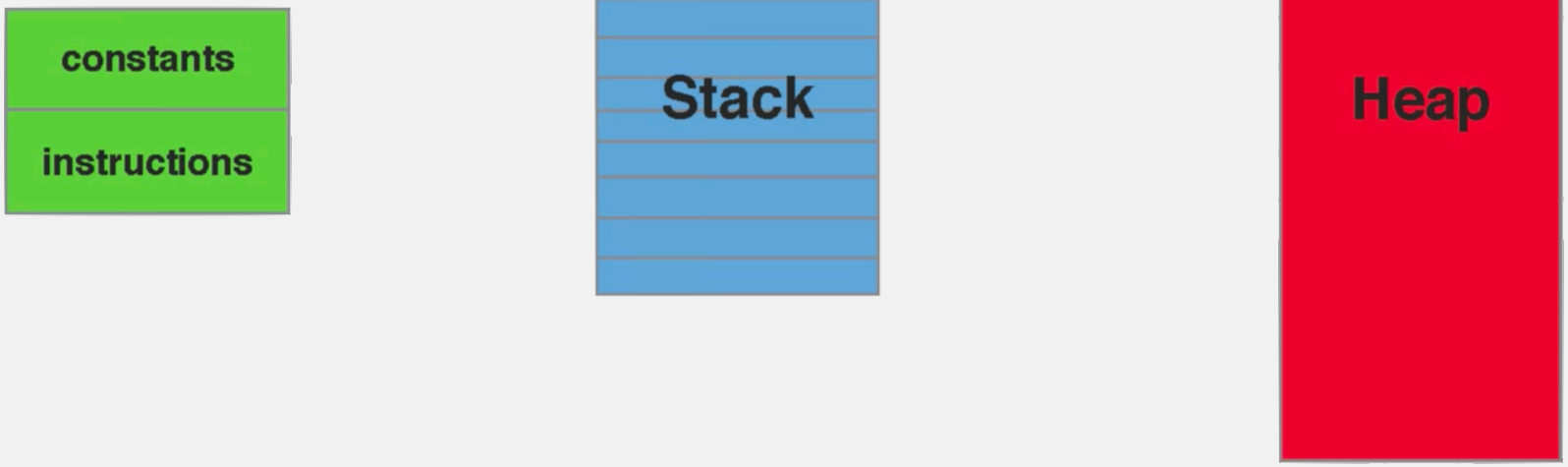
ТЕХНОСФЕРА

Производительность

Дмитрий Дорофеев

Пожалуйста, отметьтесь на лекции

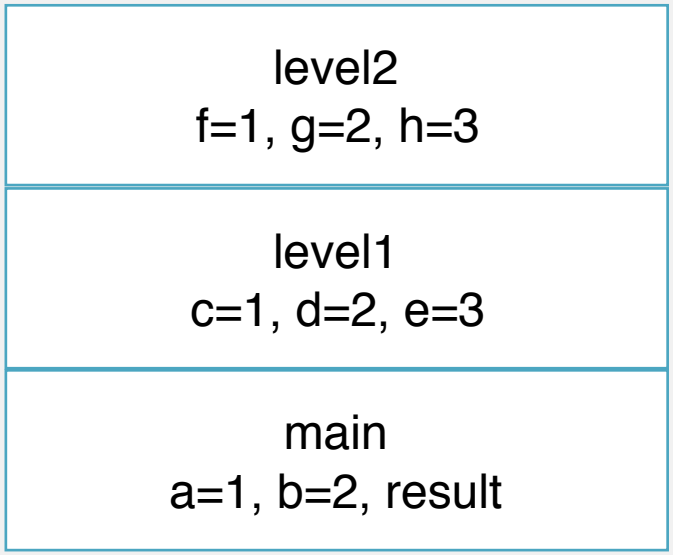
Модель памяти Go



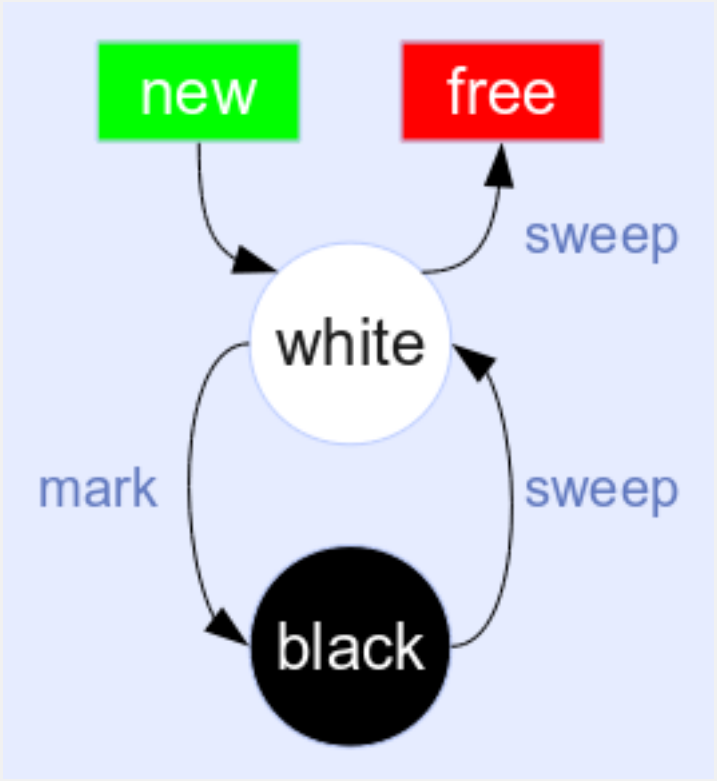
Stack

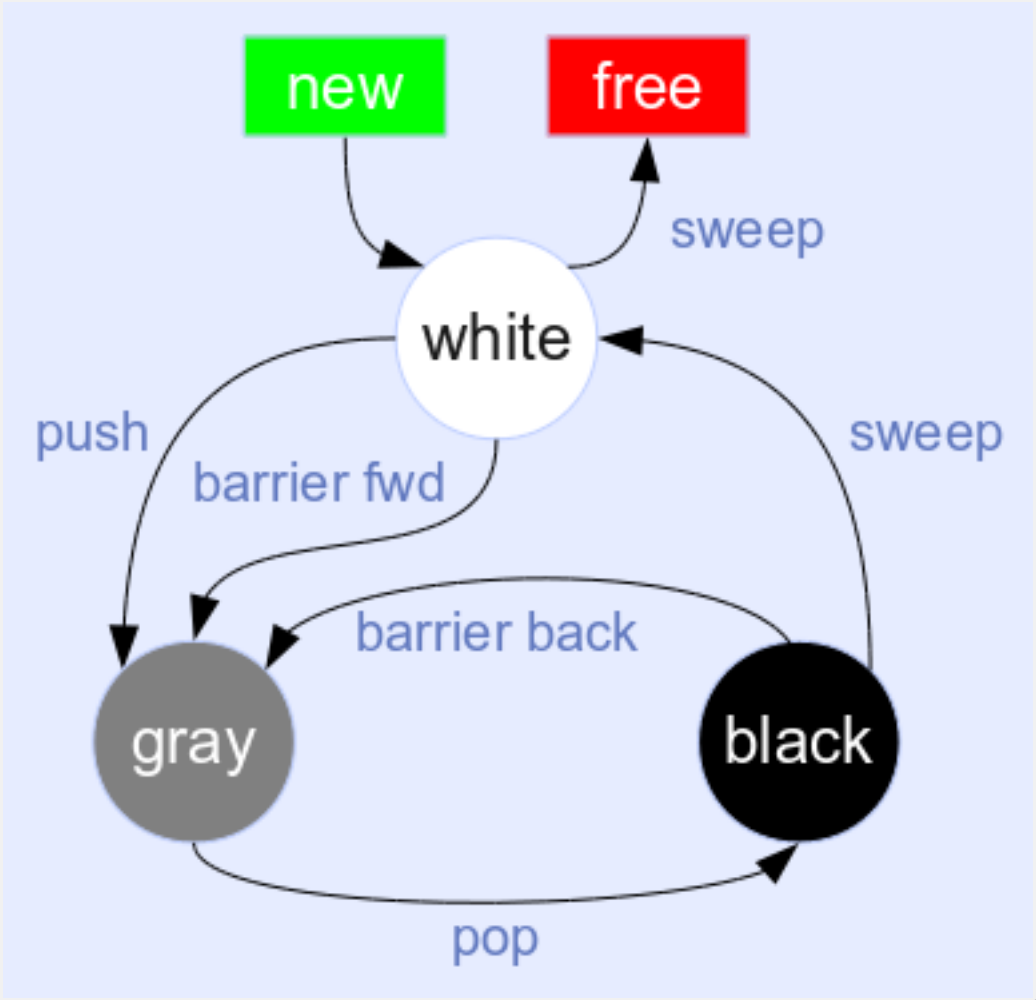
```
1.  package main

3.  func main() {
4.    a, b := 1, 2
5.    result := level1(a, b)
6.    fmt.Println(result)
7.  }
8.
9.  func level1(c, d int) int {
10.   e := 3
11.   return level2(c, d, e)
12. }
13.
14. func level2(f, g, h int) int {
15.   return f + g + h
16. }
```



Heap + GC

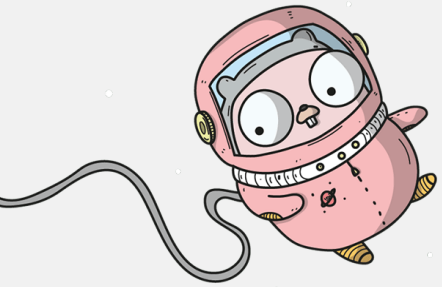

new
free
sweep
white
mark
sweep
black

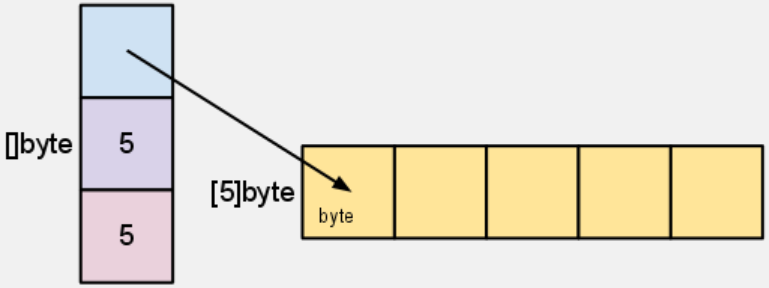

new
free
sweep
white
push
barrier fwd
sweep
barrier back
gray
black
pop

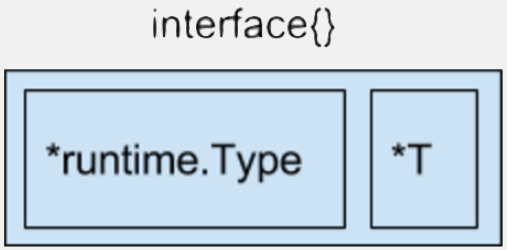
Escape analysis

```
// Sum returns the sum of the numbers 1 to 100.
func Sum() int {
        numbers := make([]int, 100)
        for i := range numbers {
                numbers[i] = i + 1
        }
        var sum int
        for _, i := range numbers {
                sum += i
        }
        return sum
}
```

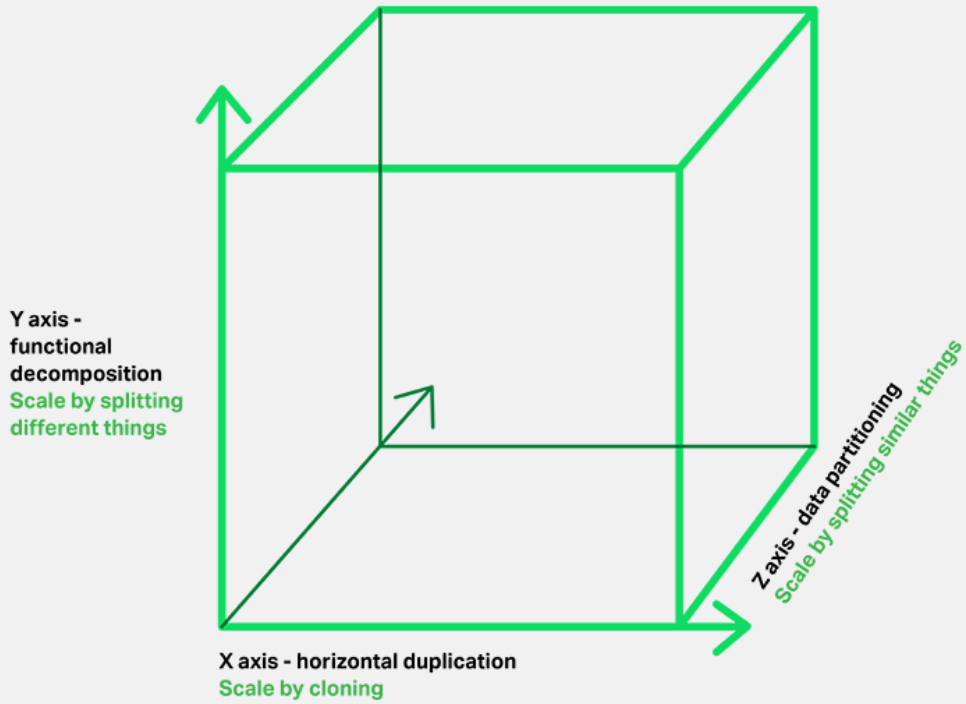
numbers never escapes Sum()

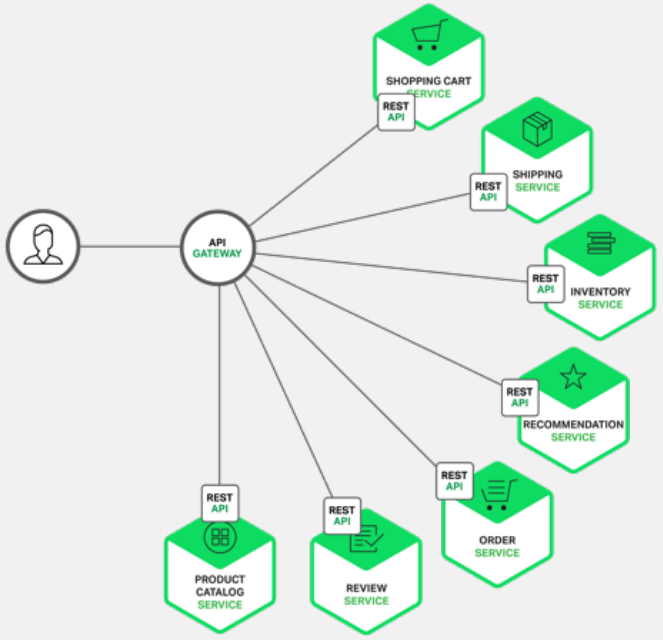



[]byte
5
5
[5]byte
byte

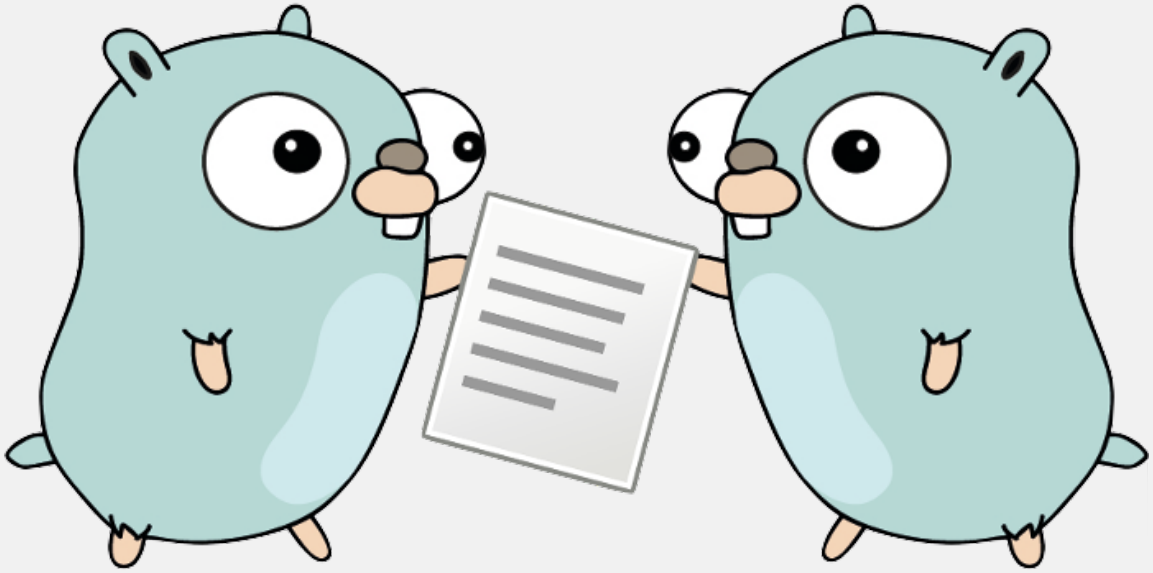

interface{}
*runtime.Type
*T

Scalability





Оставьте нам обратную связь



Это поможет сделать курс лучше

ТЕХНОСФЕРА

Спасибо за внимание!

Оставьте отзыв о занятии на портале