



Основы

Tinkoff.ru



Павел Пермяков

Руководитель отдела архитектуры

Заместитель СТО Тинькофф Инвестиции

7 лет в Тинькофф



План занятия

1. Краткая история языка
2. Golang в продуктах Тинькофф
3. Типы данных
4. Переменные, указатели
5. Синтаксические конструкции



Краткая история языка



A long time ago in a Google
far far away ...

Краткая история языка. Кто и когда?



Роберт Гризмер, Роб Пайк, Кен Томпсон

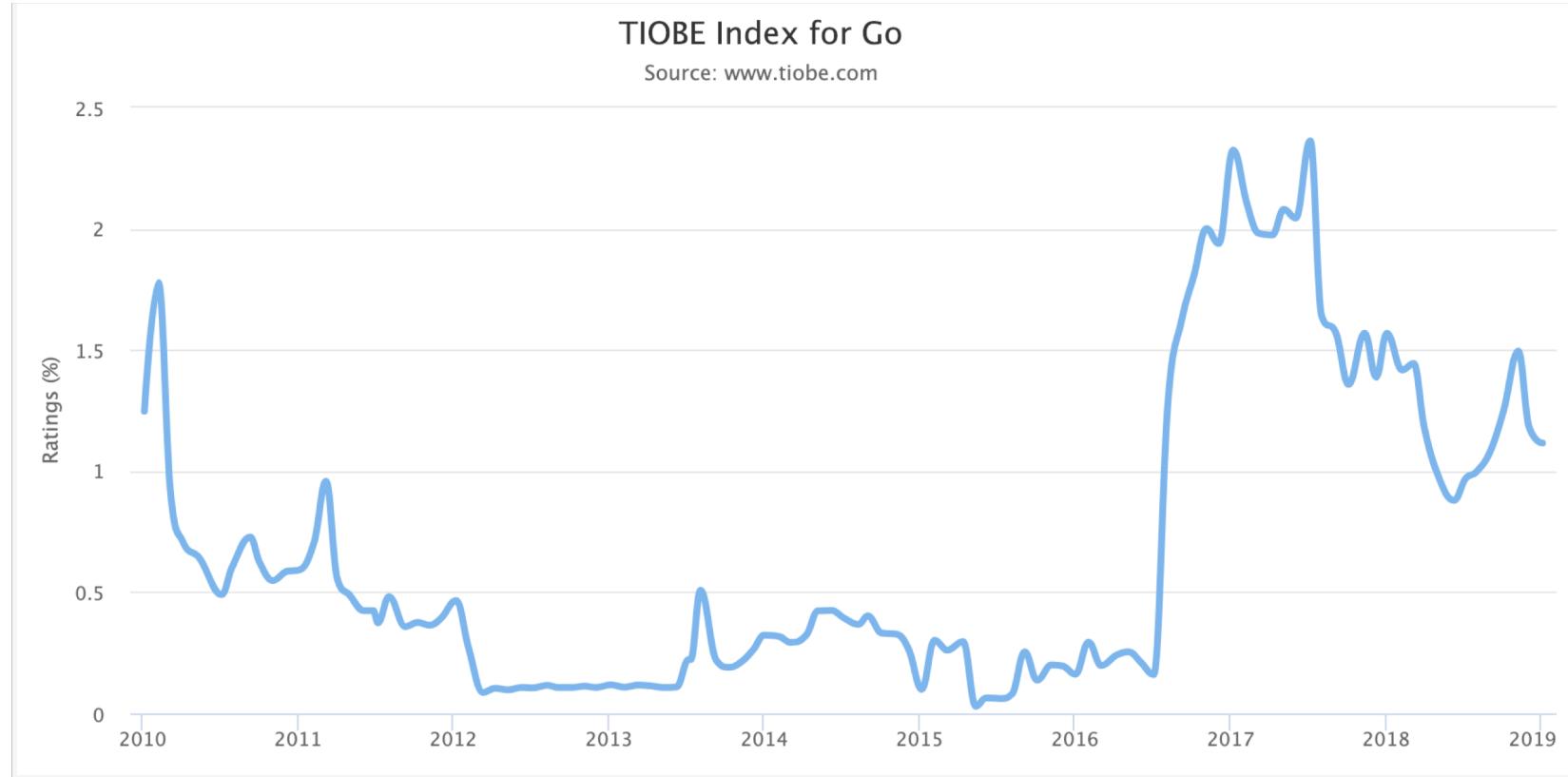




Краткая история языка. Зачем?

- Простота написания кода
- Скорость компиляции
- Производительность

Краткая история языка. Пик популярности.





Golang в продуктах Тинькофф

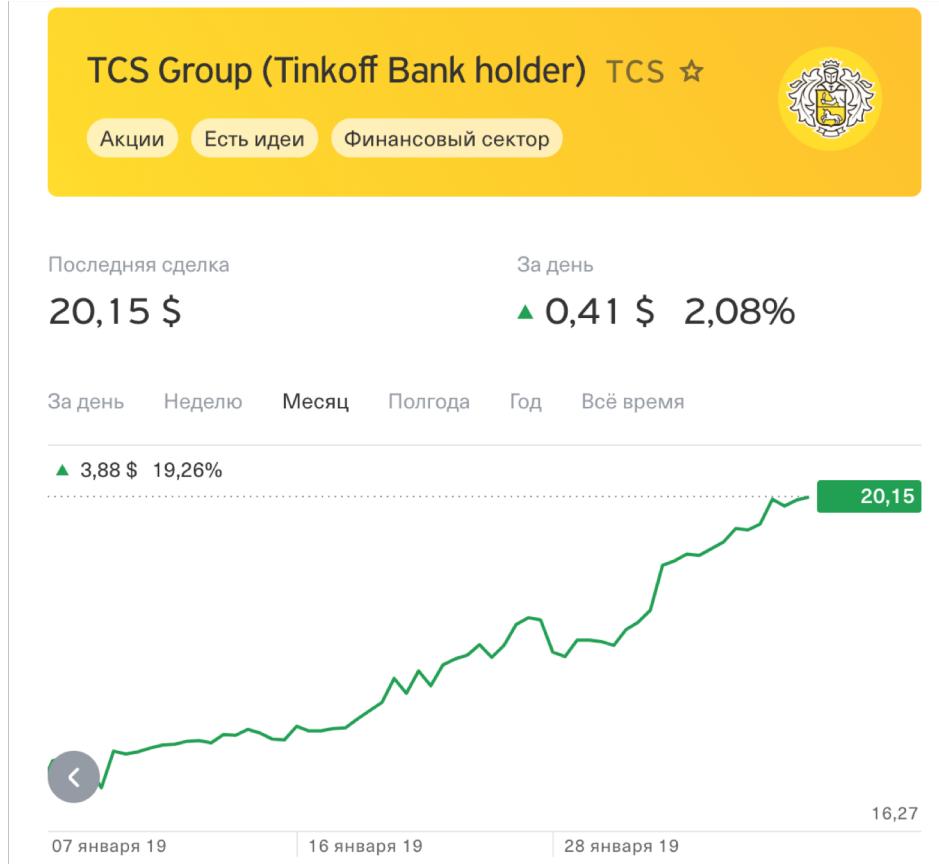
Golang в продуктах Тинькофф



Тинькофф Инвестиции



Golang в продуктах Тинькофф



Golang в продуктах Тинькофф



Все

Идеи

Новости



Google

Россия стала лидером по числу запросов госорганов к Google об удалении контента

В первом полугодии 2018 года от российских госорганов к Google поступило 19,2 тыс. запросов на удаление контента, что составляет больше 70% от всех мировых запросов, следует из ежегодного отчета компании.

Вчера в 19:37 · RNS

Избранное



TCS Group (Tink... 20,15 \$ ▲ 2,08%



Типы данных



Типы данных

Go – язык программирования со статической типизацией.

Список встроенных типов данных:

- Целочисленные типы
- Числа с плавающей запятой
- Комплексные числа
- Логические значения
- Строки



Целочисленные типы

11 типов

- int8, int16, int32, int64, int
- uint8, uint16, uint32, uint64, uint
- byte – синоним uint8
- rune – синоним uint32
- uintptr



Целочисленные типы

int8	uint8 (=byte)
signed integer	unsigned integer
8 бит	8 бит
от -128 до 127	от 0 до 255



Целочисленные типы

```
1 var a int8 = -1
2 var b uint8 = 2
3 var c byte = 2 // синоним типа uint8
4 var d int16 = -15
5 var f uint16 = 63
6 var g int32 = -21
7 var h rune = -21 // синоним типа int32
8 var j uint32 = 42
9 var k int64 = -9
10 var l uint64 = 10
11 var m int = 102
12 var n uint = 105
```



Числа с плавающей запятой

2 типа

- float32
- float64

```
1 var f float32 = 18
2 var g float32 = 4.5
3 var d float64 = 0.23
4 var pi float64 = 3.14
5 var e float64 = 2.7
```



Комплексные числа

2 типа

- complex64
- complex128

```
1 var f complex64 = 1+2i  
2 var g complex128 = 4+3i
```



Логические значения

1 тип

- bool

1	var isActive bool = true
2	var isEnabled bool = false



Строки

1 тип

- string

1

```
var name string = "Том Сойер"
```



Переменные, указатели



Переменные

Определение переменной в Go означает передачу компилятору информации о типе данных, а так же о месте и размере выделяемой памяти, которую займет переменная.

```
var [список переменных] [тип данных]
```

```
var x int
```

```
var tinkoff, sberbank, alfabank string
```



Переменные

Переменная может быть инициализирована при объявлении

```
var x int = 100
```

```
var (
```

```
    x int = 100
```

```
    tinkoff, sberbank string = "bank"
```

```
)
```



Переменные

Если переменная не инициализирована при объявлении, то ей присваивается значения по умолчанию, в зависимости от типа переменной:

числовые типы	0
логический тип	false
строки	“”



Указатели

Указатели представляют собой объекты, значением которых служат **адреса** других объектов (например, переменных).

Указатели определяются как обычные переменные, только с указанием символа ***** перед типом данных.

```
1 var p *int
```



Синтаксические конструкции



Синтаксические конструкции

- Условный оператор
- Оператор цикла
- Оператор множественного выбора



Условный оператор

Определение условного оператора в Go выглядит следующим образом:

```
if [условие]  {  
    ...  
}
```



Условный оператор

```
1 if true {  
2   fmt.Println("Это выражение выполнится всегда")  
3 }  
4  
5 if i >= 0 && i < 10 {  
6   fmt.Println(i)  
7 }
```



Оператор цикла

В языке Go один оператор цикла — это **for**, который повторяет список инструкций заданное количество раз. Цикл состоит из трёх выражений:

- **инициализация.** Выполняется перед первой итерацией.
- **условие.** Вычисляется перед каждой итерацией.
- **инкремент.** Выполняется после каждой итерации.



Оператор цикла

Определения оператора цикла в Go выглядит следующим образом:

```
for [условие | (инициализация; условие; инкремент) |  
диапазон] {  
    [тело цикла]  
}
```



Оператор цикла

```
1  for { // бесконечный цикл
2      // Выход из цикла должен быть организован вручную,
3      // обычно это делается с помощью конструкций return или break
4  }
5
6  for { // Аналог цикла do-while в Си
7      ... // Операции тела цикла
8      if i >= 10 { // Условие выхода
9          break
10     }
11 }
12
13 for i < 10 { // цикл с предусловием, аналог while в Си
14 }
15
16 for i := 0; i < 10; i++ { // цикл со счётчиком, аналог for в Си
17 }
```



Оператор множественного выбора

Определения оператора множественного выбора в Go выглядит следующим образом:

```
switch [переменная] {  
    case [значение] :  
        ...  
    case [значение, значение] :  
        ...  
}  
}
```



Оператор множественного выбора

```
1 switch value {  
2     case 1:  
3         fmt.Println("One")  
4     fallthrough // Далее будет выполнена ветвь "case 0:"  
5     case 0:  
6         fmt.Println("Zero")  
7 }
```



Обратная связь

Tinkoff.ru



Спасибо за внимание

Tinkoff.ru