Go 语言类型转换 Go 错误处理

Go 语言接口

Go 语言提供了另外一种数据类型即接口，它把所有的具有共性的方法定义在一起，任何其他类型只要实现了这些方法就是实现了这个接口。

实例

```
/* 定义接口 */
type interface_name interface {
   method_name1 [return_type]
   method_name2 [return_type]
   method_name3 [return_type]
   ...
   method_namen [return_type]
}

/* 定义结构体 */
type struct_name struct {
   /* variables */
}

/* 实现接口方法 */
func (struct_name_variable struct_name) method_name1() [return_type] {
   /* 方法实现 */
}
...
func (struct_name_variable struct_name) method_namen() [return_type] {
   /* 方法实现*/
}
```

实例

```
package main

import (
    "fmt"
)

type Phone interface {
    call()
}

type NokiaPhone struct {
}
```

```
func (nokiaPhone NokiaPhone) call() {
    fmt.Println("I am Nokia, I can call you!")
}

type IPhone struct {
}

func (iPhone IPhone) call() {
    fmt.Println("I am iPhone, I can call you!")
}

func main() {
    var phone Phone

    phone = new(NokiaPhone)
    phone.call()

    phone = new(IPhone)
    phone.call()

}
```

在上面的例子中，我们定义了一个接口Phone，接口里面有一个方法call()。然后我们在main函数里面定义了一个Phone类型变量，并分别为之赋值为NokiaPhone和IPhone。然后调用call()方法，输出结果如下：

```
I am Nokia, I can call you!
I am iPhone, I can call you!
```

← Go 语言类型转换 Go 错误处理 →

2 篇笔记

写笔记

给接口增加参数：

```
package main

import (
    "fmt"
)
type Man interface {
    name() string;
    age() int;
}
```

```
type Woman struct {
}

func (woman Woman) name() string {
   return "Jin Yawei"
}
func (woman Woman) age() int {
   return 23;
}

type Men struct {
}

func ( men Men) name() string {
   return "liweibin";
}
func ( men Men) age() int {
    return 27;
}

func main(){
    var man Man;

    man = new(Woman);
    fmt.Println( man.name());
    fmt.Println( man.age());
    man = new(Men);
    fmt.Println( man.name());
    fmt.Println( man.age());
}
```

Webben 10个月前 (06-01)

```
func (name string) imp() string{
    print("这是实现方法的写法")
}

func sum(x int,y int) int{
    print("这是正常写法")
}
```

杀手 6个月前 (09-18)