



Golang struct、interface 组合嵌入类型详解

 isgiker 关注

 0.295 2016.07.20 22:48:54 字数 2,045 阅读 13,220

概述

在 Go 语言中，如果一个结构体和一个嵌入字段同时实现了相同的接口会发生什么呢？我们猜一下，可能有两个问题：

- 编译器会因为我们同时有两个接口实现而报错吗？
- 如果编译器接受这样的定义，那么当接口调用时编译器要怎么确定该使用哪个实现？

实现接口

当涉及到我们该怎么让我们的类型实现接口时，Go 语言是特别的一个。Go 语言不需要我们显式的实现类型的接口。如果一个接口里的所有方法都被我们的类型实现了，那么我们就说该类型实现了该接口。

```
1 package main
2
3 import (
4     "log"
5 )
6
7 type User struct {
8     Name string
9     Email string
10 }
11
12 func (u *User) Notify() error {
13     log.Printf("User: Sending User Email To %s<%s>\n",
14         u.Name,
15         u.Email)
16
17     return nil
18 }
19
20 type Notifier interface {
21     Notify() error
22 }
23
24 func SendNotification(notify Notifier) error {
25     return notify.Notify()
26 }
27
28 func main() {
29     user := User{
30         Name: "AriesDevil",
31         Email: "ariesdevil@xxoo.com",
32     }
33
34     SendNotification(user)
35 }
```


实例源码：<http://play.golang.org/p/KG8-Qb7gqM>

以上代码输出：

```
prog.go:34: cannot use user (type User) as type Notifier in argument to SendNotification: User does not implement Notifier (Notify method has pointer receiver)
```

为什么编译器不考虑我们的值是实现该接口的类型？接口的调用规则是建立在这些方法的接受者和接口如何被调用的基础上。下面的是语言规范里定义的规则，这些规则用来说明是否我们一个



 isgiker 关注

总资产2 (约0.22元)

Flieddata
阅读 1,542

Elasticsearch API:Search
阅读 1,095

- 推荐阅读
- 多个程序监听同一端口 - socket端口复用技术
阅读 1,983
 - 八个PHP 高级工程经典面试题
阅读 2,088
 - 关于 Golang defer 的使用规则和案例
阅读 463
 - 2020 腾讯社招Golang后端面试经验分享
阅读 6,769
 - Go的50坑：新Golang开发者要注意的陷阱、技巧和常见错误[2]
阅读 144



者可以是值类型也可以是指针类型。显然我们的例子不符合该规则，因为我们传入 SendNotification 函数的接口变量是一个值类型。

- 类型 T 的可调用方法集包含接受者为 T 的所有方法

这条规则说的是如果我们用来调用特定接口方法的接口变量是一个值类型，那么方法的接受者必须也是值类型该方法才可以被调用。显然我们的例子也不符合这条规则，因为我们 Notify 方法的接受者是一个指针类型。

语言规范里只有这两条规则，我通过这两条规则得出了符合我们例子的规则：

- 类型 T 的可调用方法集不包含接受者为 *T 的方法

我们碰巧赶上了我推断出的这条规则，所以编译器会报错。Notify 方法使用指针类型作为接受者而我们却通过值类型来调用该方法。解决办法也很简单，我们只需要传入 User 值的地址到 SendNotification 函数就好了：

```
1 package main
2
3 import (
4     "log"
5 )
6
7 type User struct {
8     Name string
9     Email string
10 }
11
12 func (u *User) Notify() error {
13     log.Printf("User: Sending User Email To %s<%s>\n",
14         u.Name,
15         u.Email)
16
17     return nil
18 }
19
20 type Notifier interface {
21     Notify() error
22 }
23
24 func SendNotification(notify Notifier) error {
25     return notify.Notify()
26 }
27
28 func main() {
29     user := &User{
30         Name: "AriesDevil",
31         Email: "ariesdevil@xxoo.com",
32     }
33
34     SendNotification(user)
35 }
```

嵌入类型

结构体类型可以包含匿名或者嵌入字段。也叫做嵌入一个类型。当我们嵌入一个类型到结构体中时，该类型的名字充当了嵌入字段的字段名。

```
1 package main
2
3 import (
4     "log"
5 )
6
7 type User struct {
8     Name string
9     Email string
10 }
11
```

写下你的评论...

评论1

赞11

...

```
17 func (u *User) Notify() error {
18     log.Printf("User: Sending User Email To %s<%s>\n",
19         u.Name,
20         u.Email)
21
22     return nil
23 }
24
25 type Notifier interface {
26     Notify() error
27 }
28
29 func SendNotification(notify Notifier) error {
30     return notify.Notify()
31 }
32
33 func main() {
34     admin := &Admin{
35         User: User{
36             Name: "AriesDevil",
37             Email: "ariesdevil@xxoo.com",
38         },
39         Level: "super",
40     }
41
42     SendNotification(admin)
43 }
```

详细代码: <http://play.golang.org/p/ivzzk78TC>

事实证明, 我们可以 Admin 类型的一个指针来调用 SendNotification 函数。现在 Admin 类型也通过来自嵌入的 User 类型的方法提升实现了该接口。

如果 Admin 类型包含了 User 类型的字段和方法, 那么它们在结构体中的关系是怎么样的呢?

当我们嵌入一个类型, 这个类型的方法就变成了外部类型的方法, 但是当它被调用时, 方法的接受者是内部类型(嵌入类型), 而非外部类型。— Effective Go

因此嵌入类型的名字充当着字段名, 同时嵌入类型作为内部类型存在, 我们可以使用下面的调用方法:

```
1 admin.User.Notify()
2 // Output: User: Sending User Email To AriesDevil<ariesdevil@xxoo.com>
```

这儿我们通过类型名称来访问内部类型的字段和方法。然而, 这些字段和方法也同样被提升到了外部类型:

```
1 admin.Notify()
2 // Output: User: Sending User Email To AriesDevil<ariesdevil@xxoo.com>
```

所以通过外部类型来调用 Notify 方法, 本质上是内部类型的方法。

下面是 Go 语言中内部类型方法集提升的规则:

给定一个结构体类型 S 和一个命名为 T 的类型, 方法提升像下面规定的这样被包含在结构体方法集中:

- 如果 S 包含一个匿名字段 T, S 和 *S 的方法集都包含接受者为 T 的方法提升。

这条规则说的是当我们嵌入一个类型，可以被外部类型的指针调用的方法集只有嵌入类型的接受者为指针类型的方法集，也就是说，当外部类型使用指针调用内部类型的方法时，只有接受者为指针类型的内部类型方法集将被提升。

- 如果 S 包含一个匿名字段 *T，S 和 *S 的方法集都包含接受者为 T 或者 *T 的方法提升

这条规则说的是当我们嵌入一个类型的指针，嵌入类型的接受者为值类型或指针类型的方法将被提升，可以被外部类型的值或者指针调用。

这就是语言规范里方法提升中仅有的三条规则，我根据这个推导出一条规则：

如果 S 包含一个匿名字段 T，S 的方法集不包含接受者为 *T 的方法提升。

这条规则说的是当我们嵌入一个类型，嵌入类型的接受者为指针的方法将不能被外部类型的值访问。这也是跟我们上面陈述的接口规则一致。

回答开头的问题

现在我们可以写程序来回答开头提出的两个问题了，首先我们让 Admin 类型实现 Notifier 接口：

Admin 类型实现的接口显示一条 admin 方面的信息。当我们使用 Admin 类型的指针去调用函数 SendNotification 时，这将帮助我们确定到底是哪个接口实现被调用了。

现在创建一个 Admin 类型的值并把它的地址传入 SendNotification 函数，来看看发生了什么：

```
1 package main
2
3 import (
4     "log"
5 )
6
7 type User struct {
8     Name string
9     Email string
10 }
11
12 type Admin struct {
13     User
14     Level string
15 }
16
17 func (u *User) Notify() error {
18     log.Printf("User: Sending User Email To %s<%s>\n",
19         u.Name,
20         u.Email)
21
22     return nil
23 }
24
25 //func (a *Admin) Notify() error {
26 // log.Printf("Admin: Sending Admin Email To %s<%s>\n",
27 //     a.Name,
28 //     a.Email)
29 //
30 // return nil
31 //}
32
33 type Notifier interface {
34     Notify() error
35 }
36
37 func SendNotification(notify Notifier) error {
38     return notify.Notify()
39 }
40
```

写下你的评论...

评论1

赞11

...

源代码实例分享：<https://play.golang.org/p/JGhFaJnGpS>

- **编译器会因为我们同时有两个接口实现而报错吗？**
不会，因为当我们使用嵌入类型时，类型名充当了字段名。嵌入类型作为结构体的内部类型包含了自己的字段和方法，且具有唯一的名字。所以我们可以有同一接口的内部实现和外部实现。
- **如果编译器接受这样的定义，那么当接口调用时编译器要怎么确定该使用哪个实现？**
如果外部类型包含了符合要求的接口实现，它将会被使用。否则，通过方法提升，任何内部类型的接口实现可以直接被外部类型使用。
- **总结**
在 Go 语言中，方法，接口和嵌入类型一起工作方式是独一无二的。这些特性可以帮助我们像面向对象那样组织结构然后达到同样的目的，并且没有其它复杂的东西。用本文中谈到的语言特色，我们可以以极少的代码来构建抽象和可伸缩性的框架。

转自：<http://www.jb51.net/article/56831.htm>

11人点赞 >

Golang

"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



isgiker Don't view things as static and isolated.
总资产2 (约0.22元) 共写了4.8W字 获得68个赞 共36个粉丝

关注

瓦努阿图护照办理，最快20天批准

护照激活使用，新加坡全球护照交流中心行业首推真正教会客人，F
好用活一本护照，护照配欧盟身份

写下你的评论...

全部评论 1 只看作者

按时间倒序 按时间正序

咕咕鸡
2楼 2016.11.15 09:30
第一个例子也可以这么改。
func main() {
user := User{

写下你的评论... 评论1 赞11 ...

简书

首页

下载APP

搜索

Aa

beta

登录

注册

SendNotification(&user)

}

2

回复

被以下专题收入，发现更多相似内容

- Golang
- Python/...

推荐阅读

更多精彩内容>

百战程序员V1.2——尚学堂旗下高端培训_ Java1573题

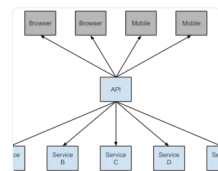
百战程序员_ Java1573题 QQ群： 561832648489034603 掌握80%年薪20万掌握50%年薪...

Albert陈凯 阅读 12,338 评论 2 赞 32

Spring Cloud

Spring Cloud为开发人员提供了快速构建分布式系统中一些常见模式的工具（例如配置管理，服务发现，断路器，智...

卡卡罗2017 阅读 87,588 评论 13 赞 122



Java初级面试题

1. Java基础部分 基础部分的顺序：基本语法，类相关的语法，内部类的语法，继承相关的语法，异常的语法，线程的语...

会飞的鱼69 阅读 24,505 评论 18 赞 392

iOS面试题及详解(上篇)

多线程、特别是NSOperation 和 GCD 的内部原理。运行时机制的原理和运用场景。SDWebImage的原...

LZM轮回 阅读 1,167 评论 0 赞 10



不让妈妈出门的孩子

周末早起准备去驾校，没想到3岁的儿子也早早醒来，一个劲地喊妈妈。过去床边想哄他继续入睡，但是他坚持要起来，哭瘪瘪地...

山河万朵 阅读 157 评论 0 赞 3