

### 10.5.2 看一个需求

一个结构体的声明是这样的:

```
package model

type Student struct {
    Name string...
}
```

因为这里的 Student 的首字母 S 是大写的, 如果我们在其它包创建 Student 的实例(比如 main 包), 引入 model 包后, 就可以直接创建 Student 结构体的变量(实例)。但是问题来了, 如果首字母是小写的, 比如是 type student struct {...} 就不不行了, 怎么办---> 工厂模式来解决。

### 10.5.3 工厂模式来解决问题

使用工厂模式实现跨包创建结构体实例(变量)的案例:

如果 model 包的 结构体变量首字母大写, 引入后, 直接使用, 没有问题

```
1 package model
2
3 //定义一个结构体
4 type Student struct{
5     Name string
6     Score float64
7 }
```

```
package main
import (
    "fmt"
    "go_code/chapter10/factory/model"
)

func main() {
    //创建要给学生实例
    var stu = model.Student{
        Name : "tom",
        Score : 78.9,
    }

    fmt.Println(stu)
}
```

## 工厂模式图解

### Student.go

```
1 package model
2
3 //定义一个结构体
4 type student struct{
5     Name string
6     Score float64
7 }
8
9 //因为student结构体首字母是小写，因此是只能在model使用
10 //我们通过工厂模式来解决
11
12 func NewStudent(n string, s float64) *student {
13     return &student{
14         Name : n,
15         Score : s,
16     }
17 }
18
```

### Main.go

```
package main
import (
    "fmt"
    "go_code/chapter10/factory/model"
)

func main() {
    //创建要给Student实例
    // var stu = model.Student{
    //     Name : "tom",
    //     Score : 78.9,
    // }

    //定student结构体是首字母小写，我们可以通过工厂模式来解决
    var stu = model.NewStudent("tom~", 88.8)

    fmt.Println(*stu) //&{....}
    fmt.Println("name=", stu.Name, " score=", stu.Score)
}
```

不能导出（小写）的结构体，写一个能导出（大写）函数在其他包调

用，传入不能导出结构体的参数值，设置函数返回值是指针类型的该结构体，从而实现将不能导出的结构体在别的包使用，美其名曰工厂模式

如果结构体重某一个字段小写，从而导致在别的包无法使用，学要写一个方法，传入该结构体要传递的字段，从而实现在别的包使用该结构体小写的字段。类似 java 的 `get()`

这样可以提高保护性，如果你将字段改为大写的话，就是一个 `public`，但是你将方法改为大写的话，可以只限制为只读，提高了安全性