面向对象编程1

作者：少林之巅

目录

struct声明和定义

1. Go中面向对象是通过struct来实现的, struct是用户自定义的类型

```
type User struct {
    Username  string
    Sex       string
    Age       int
    AvatarUrl string
}
```

注意：type是用来定义一种类型

struct声明和定义

2. struct初始化方法1

```
var user User
user.Age = 18
user.Username = “user01”
user.Sex = “男”
user.AvatarUrl = “http://my.com/xxx.jpg"
```

注意：使用变量名+ ‘.’ + 字段名访问结构体中的字段

struct声明和定义

3. struct初始化方法2

```
var user User = User {
    “Username” : “user01”,
    “Age”: 18,
    “Sex”: “男”,
    “AvatarUrl”: “http://my.com/xxx.jpg”,
}
```

注意：也可以部分初始化

更简单的写法：

```
user := User {
    “Username” : “user01”,
    “Age”: 18,
    “Sex”: “男”,
    “AvatarUrl”: “http://my.com/xxx.jpg”,
}
```

struct声明和定义

4. struct初始化的默认值

```
var user User
fmt.Printf("%#v\n", user)
```

struct声明和定义

5. 结构体类型的指针

```
var user *User = &User{}
fmt.Printf(“%p %#v\n”, user)
```

```
var user *User = &User {
    “Username” : “user01”,
    “Age”: 18,
    “Sex”: “男”,
    “AvatarUrl”: “http://my.com/xxx.jpg”,
}
```

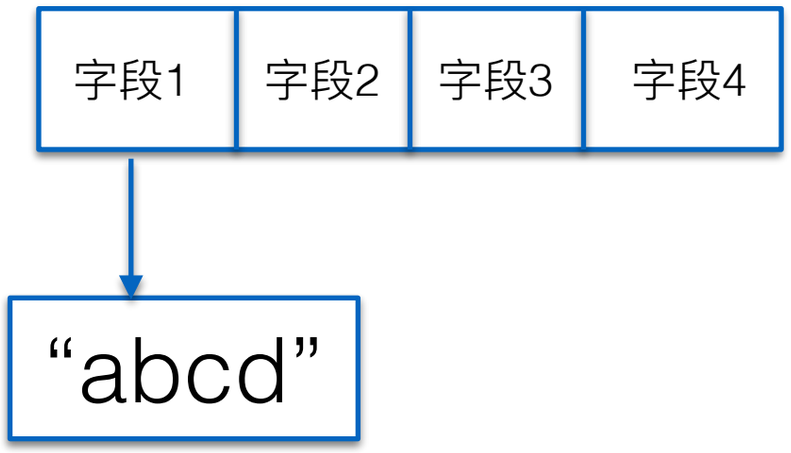
```
var user User = new(User)

user.Age = 18
user.Username = “user01”
user.Sex = “男”
user.AvatarUrl = “http://my.com/xxx.jpg"
```

注意：&User{}和new(User)本质上是一样的，都是返回一个结构体的地址

struct内存布局

6. 结构体的内存布局： 占用一段连续的内存空间



struct内存布局

7. 结构体没有构造函数， 必要时需要自己实现

匿名字段和嵌套

8. 匿名字段: 即没有名字的字段

```
type User struct {
    Username  string
    Sex       string
    Age       int
    AvatarUrl string
}
```

```
type User struct {
    Username  string
    Sex       string
    Age       int
    AvatarUrl string
    int
    string
}
```

注意：匿名字段默认采用类型名作为字段名

匿名字段和嵌套

9. 结构体嵌套

```
type Address struct {
    City           string
    Province       string
}
```

```
type User struct {
    Username  string
    Sex       string
    Age       int
    AvatarUrl string
    address Address
}
```

匿名字段和嵌套

10.匿名结构体

```
type Address struct {
    City            string
    Province        string
}
```

```
type User struct {
    Username  string
    Sex       string
    Age       int
    AvatarUrl string
    Address
}
```

匿名字段和嵌套

11.匿名结构体与继承

```
type Animal struct {
    City            string
    Province        string
}
```

```
type User struct {
    Username  string
    Sex       string
    Age       int
    AvatarUrl string
    Address
}
```

匿名字段和嵌套

12. 冲突解决

```
type Address struct {
    City            string
    Province        string
    CreateTime      string
}
```

```
type Email struct {
    Account             string
    CreateTime          string
}
```

```
type User struct {
    Username  string
    Sex       string
    Age       int
    AvatarUrl string
    Address
    Email
    CreateTime string
}
```

结构体与tag应用

13.字段可见性，大写表示可公开访问，小写表示私有

```
type User struct {
    Username  string
    Sex       string
    Age       int
    avatarUrl string
    CreateTime string
}
```

结构体与tag应用

14. tag是结构体的元信息，可以在运行的时候通过反射的机制读取出来

```
type User struct {
    Username  string    `json:"username",db:"user_name"`
    Sex       string    `json:"sex"`
    Age       int       `json:"age"`
    avatarUrl string
    CreateTime string
}
```

字段类型后面，以反引号括起来的key-value结构的字符串，多个tag以逗号隔开。

课后练习

1. 实现一个简单的学生管理系统，每个学生有分数、年级、性别、名字等字段，用户可以在控制台添加学生、修改学生信息、打印所有学生列表的功能。

```
package main
import (
    “fmt”
    “flag”
)
type Student struct {
    Username string
    Score float32
    Grade string
    Sex int
}

func main() {
}
```