所有的Go源程序文件头部必须由一个包声明语句，Go通过包来管理命名空间

Go源代码的特征：

源程序以.go为后缀；

默认UTF-8编码；

标识符区分大小写；

语句结尾的分号可以省略；

函数以func开头， 函数体开头的“{”必须在函数头所在行尾部，不能单独起一行;

字符串使用双引号；

调用包里的方法使用. 例如fmt.Printf;

main函数所在的包名必须是main。

Go是一种强类型静态编译语言，再定义变量和常量时需要指明数据类型。

Go的变量声明有两种形式：

1. 显式的完整声明：

var name datatype [ = value]

注意：关键字var 用于声明变量；

name是变量名，dataType是基本类型

value可以是初始值，或者一个表达式，不指定为零值

1. 短类型声明

varName := value

注意： := 只能出现再函数内，此时Go编译器自动进行数据类型的判断

Go支持多个类型变量的同事声明并赋值： a,b := 1,”hello”

Go 语言内置七类基本数据类型( 20 个具体子类型)。

布尔类型: bool

整型: byte int int8 int16 init32 int64 uint uint8 uint16 uint32 uint64 uintptr

浮点型 : float32 float64

复数: complex64 complex128

字符 : rune

字符串: string

错误类型：error

布尔类型：

关键字bool 只有两个值：true和false，例如

var ok bool

ok = true

或者

OK := false

注意：声明的布尔型变量不指定初始值，默认为false

整型：

var a int =1

var b int32 = 1

Go复合数据类型

指针

数组

切片

map

struct 结构

if语句特点：

if后面的条件判断子句不需要使用小括号括起来；

{必须放在行尾，和if或者if else放在一行；

if后面可以带一个简单的变量初始化声明语句，并以分号分割，该变量的作用域是整个if语句块，包括后面的else if 和 else分支；

Go没有条件运算符（三目运算）；

if分支遇到return后直接返回，遇到break则跳过break下方的if语句块

例如：

if x < y {

return y

} else {

return x

}

再例如：

if x :=f(); x<y {//初始化语句中的声明变量x

return x

} else if x > z { //x在else if里面一样可以被访问

return z

} else { y

return y

}

switch 语句

for语句

函数：

包括：函数声明关键字func、函数名、参数列表、返回列表和函数体；

函数名规则：首字母的大小写决定函数在其他包的可见性，大写其他包可见，小写只能相同包可以访问；

函数的参数和返回值都得（）包裹，如果只有一个返回值，还是非命名的参数，可省略（）；

包裹函数体的{必须位于函数返回值同行末尾。

func add(a int,b int) (sum int){

sum = a+b

return sum

}

func swap(a,b int) (int,int){

return b,a

}