Golang需注意的点:

1.rune占4个字节，有符号，表示一个unicode码，一般用于处理汉字乱码问题。

2.int/uint的大小都和操作系统有关系，32位操作系统为4字节，64位操作系统为8字节。

3.golang整型变量使用时，注意保小不保大的原则，即能用少的字节整数就用少的字节。

4.浮点数=符号位+尾数位+指数位。(float32保存小数可能造成尾数位丢失，导致丢失精度)

float32=1个符号位+8个指数位+23个尾数位 float64=1个符号位+11个指数位+52个尾数位

5.浮点型不受操作系统的限制，分为float32和float64两种。

6.浮点数中E表示除，e表示乘。如3e2表示3\*10^2，3E2表示3/10^2。

7.传统的字符串是一串固定长度的字符连接起来的字符序列。Go的字符串是单个字节连接构成的。

8.如果要保存的字符的utf-8码大于255，则最好使用int来保存字符，输出字符用%c格式。

9.字符类型本质是一个整数类型，是可以用来相加的。

10.golang中字符串若以双引号表示，则会识别转义字符，若以反引号表示，则不会识别。

11.golang中，数据类型的转换既可以从小精度到大精度转换，也可以从大精度往小精度转换。但是从大精度往小精度转换可能会存在数据溢出。且被转换的只是存储的数据，并非变量本身的数据类型。

12.strconv包中Format开头的方法是基本数据类型转换为字符串，Parse开头的方法是将字符串转为基本数据类型。

13.在将字符串转为基本整数类型时，需要注意string类型的字符串能够转成有效的数据。如果转换出错，则接收的变量保存其对应数据类型对应的零值。

14.所有的值类型都有对应的指针类型。

15.golang支持在if中定义一个变量，如if age:=20;age<50。不能使用var定义变量。

16.golang的if语句必须包含{}，不管其有几条执行语句。若包含else，则else必须跟在if的大括号”}”后。且程序有且仅有一个入口。

17.嵌套分支不宜过多，尽量控制在3层以内。

18.golang的switch语句后的条件表达式的值将和case的表达式进行比较，若满足，则执行。

19.golang的switch语句中case后的表达式可以有多个，使用逗号隔开，且case后的表达式若为常量，则常量值不能重复。且case语句块不需再加break语句，因为case语句块自带的有break语句。case语句块可以加fallthrough，则会继续执行下一个case。

20.golang的switch的表达式值的数据类型要和case后表达式的数据类型一致。

21.golang的switch语句后若不跟表达式，则类似于if-else分支来使用。

22.golang的switch语句的表达式也可以直接声明一个变量，用分号结束。

23.golang的变量.(type)语句可以用于type-switch语句，用于判断某个变量属于哪个类型。

24.如果判断的为具体值且值类型为整数、浮点数、，则使用switch语句，若判断的是某个区间或表达式结果为bool则使用if语句。

25.for循环共有三种使用方式。[for 初始化变量;判断条件;变量迭代]、[for 判断条件]、[for ]。

26. [for]等价于[for ;;]，需要配合break跳出循环。

27.golang中for-range可以用来遍历字符串和数组，但是每次遍历都是按照字符遍历。

28.golang中一个汉字占3个字节。使用传统的[for ;;]是按照字节[byte]遍历，故会出现中文乱码，需将其转为rune切片遍历。

29.golang中没有while和do-while，但是可以通过for来实现while和do-while。

30.break可以终止某一个语句块的执行，通常用于终止for循环和switch语句。

31.break默认跳出最近的for循环，可以指定标签，跳出指定标签的for循环。

32.continue可以终止当次循环，可以指定标签，终止指定标签的当次循环。

33.goto语句可以无条件的跳转到程序中指定的行。程序中一般不主张使用goto语句。

34.return作用在方法或函数内，用于终止函数或方法。

35.golang的每一个文件都属于一个包，即go是以包的形式管理文件和目录结构的。文件的包名通常和文件所在的文件夹名一致，一般为小写字母。

36.golang在import包时，从$PATH的src目录开始，不用带src，编译器自动从src下引入。

37.golang的函数不支持重载，且若函数的参数个数不确定时，需将不确定个数的参数防止形参最后。如：func test(a int, args… float32)

38.golang的函数和函数所使用的变量构成的整体叫闭包。

39.golang默认情况下，若发生panic，则退出程序。

40.golang使用defer+recover来处理panic异常，这样处理会终止异常发生的方法，不会影响其他的方法。

41.golang中panic是一个内置函数，可以接收任何类型的变量，golang可以使用panic+errors.New自定义错误，panic里也可直接放入错误信息。

42.初始化数组时，等号右边一定要加上[…]或者[数组长度]。还可以指定元素的下标，如：

var arr = […]int{1:10, 2:20, 0:30}

43.数组中的元素可以是值类型，也可以是引用类型，但是不能混用。数组一旦声明了，长度就不能再改变。

44.数组的长度也是数组数据类型的一部分，因此用数组传参时，数组的长度必须一致。

45.切片是数组的引用，切片的长度可以动态变化。切片的容量等于被引用的数组的长度减去切片引用数组的开始下标。[cap>=len]。

46.切片的实际是由地址，切片的长度，切片的容量构成。

47.切片定义完成后不能直接使用，需要引用一个数组或者make一个空间来供切片使用。

48.切片可以继续切片，切片也可以通过append函数追加元素，append函数一次可以追加多个元素，但是若需要追加切片，则被追加的切片后必须加...。如slice = append(slice,slice1…)。

49.切片可以使用copy函数来拷贝切片，拷入元素的切片长度若小于源切片长度，则拷满就不拷了，且copy函数的第一个入参为目标切片。但是参与拷贝的切片之间互不影响。

50.string底层是一个byte数组，因此string也可以进行切片处理。

51.string是不可变的，故不能通过[str[下标]=值]的方式来改变字符串的值，可以先将string转为byte切片或rune切片，然后修改完后再转为string。

53.排序分为内部排序和外部排序。

54.golang中继承是通过匿名内部类来实现的，在对子类对象初始化时，需同时初始化父类对象。且调用以父类对象为参数的方法时，需写为[子类.父类名]的方式。

55.golang中使用数据库必须引入相关的驱动包，该包若未使用，需使用\_标记。