大厂面试课之Go语言

第3课：Go语言概述

同学们好！在这节课里，我们将对Go语言做一个概要性的介绍，旨在帮助大家建立一些感性认识。

在这一节，我们将动手编写第一个Go语言程序，结合该程序了解一些Go语言的基本特点，在GoLand集成开发环境中设置终端的小技巧在某些时候会很有帮助，最后我们会介绍几个Go语言开发工具集里的常用命令，为后续课程的学习打好基础。

第一个Go语言程序

象人类的自然语言一样，计算机语言最好的学习方法就是在使用中学习，边学边用，边用边学，学用结合，学以致用。因此，一开始我们就亲自动手，写一个Go语言程序看看。

代码实践

GOPATH/src/Hello/main.go

```
package main

import "fmt"

func main() {
    fmt.Println("Hello World")
}
```

输出

```
Hello World
```


首先我们启动GoLand集成开发环境。点击“New Project”。左边选择“Go (GOPATH)”。右边在“Location”编辑框中填入我们的工程路径，如：C:\Users\Minwei\Projects\Go\src\Hello。注意工程目录必须位于GOPATH\src目录下。最后点击“Create”，完成工程创建。

在位于主界面左侧的工程浏览器中，选择“Hello”工程，点击鼠标右键，弹出菜单中选择“New>Go File”，将新建Go文件命名为“main.go”。

在编辑窗格里看到的“main.go”中只包含一行代码：

```
1 package Hello
```

我们需要将默认包名“Hello”修改为“main”，即：

```
1 package main
```

导入fmt包，这步不做也可以，稍后在用到该包的时候，开发环境会自动加上：

```
package main

import "fmt"
```

定义main函数：

```
package main

import "fmt"

func main() {

}
```

在main函数中调用fmt包的Println函数，打印字符串“Hello World”：

```
package main

import "fmt"

func main() {
    fmt.Println("Hello World")
}
```

这就是我们的第一个Go语言程序。编译运行该程序，可以点击main函数左侧的绿色小三角，也可以选择“Run>Run...”菜单项，或者点击顶部工具栏里的绿色小三角。GoLand会将我们的源代码编译链接成可执行程序并运行该程序，运行结果显示在位于主界面底部的运行窗格中。

知识讲解

从第一个Go语言程序中可以看出

- 每个Go程序都必须有一个main包
- 每个Go程序都是.go结尾的
- Go程序中没有.h也没有.o，只有.go
- import用于导入其它包
- fmt包用于格式化输出和输入
- 所有函数都必须以func开头
- 函数的返回值类型不在函数名之前，而是在形参表之后
- 函数体的左花括号必须与函数名同行，不能写到下一行
- 借助“.”调用包中的函数，包名相当于名字空间
- fmt包中的Println函数会在打印完参数指定的内容后追加一个换行符
- Go语言语句不强制要求以分号结尾

这个程序虽然简单，只有寥寥五行代码，但它却展示了Go语言很多最基本的特征。首先我们看到一个能够独立运行的Go程序必须有一个main包，main包里必须有一个main函数或者叫main方法。每个Go语言的源代码文件都带有“.go”扩展名。在Go语言程序中既看不到扩展名为“.h”或“.hpp”的所谓头文件，也看不到扩展名为“.o”或“.obj”的所谓目标文件。其它象“.a”、“.so”、“.lib”、“.dll”之类的各种库文件压根就不存

在。除了最终生成的可执行文件外，就只有“.go”一种文件。在一个包比如main包中可以通过import关键字导入其它包，之后通过包名访问其中的方法或数据，比如这里就通过导入fmt包，调用了其中的Println方法，在控制台打印字符串。fmt包专门用于格式化输入和输出。在Go语言中，所有函数都必须以func开头。函数的返回值类型不是写在函数名前面，而是写在形参表后面。函数体的左花括号必须与函数名、形参表及返回值类型写在同一行里。访问包里的方法或数据，要在包名后面加“.”。Go语言里的包相当于名字空间的概念。fmt包里的Println函数会在打印完参数指定的内容后追加一个换行符，这也是该函数名字中“ln”的含义，ln即line，行，打印独立的一行，再打印另起一行。Go语言语句不强制要求以分号结尾，因此如果不是非要把多条语句写在同一行里，一般都不写分号。

Go语言的特点

下面我们对Go语言的基本特点做一个简单的归纳。

知识讲解

没有头文件，只有.go

强类型语言

编译型语言

编译链接产生的可执行程序，可以独立运行，无需依赖任何外部库

- 所有的运行时依赖都被打包到可执行程序中
- 可执行程序文件的体积通常较大
- 不允许导入未被用到的包

在Go语言程序中没有头文件，也没有库文件，只有Go文件。Go语言是一种强类型语言，也是一种编译型语言。这与弱类型语言、解释型语言有着本质性的不同。Go语言源代码一经编译链接得到可执行程序，即可独立运行，运行时不再依赖其它任何外部库，尤其是动态链接库，或称共享库，在Go语言的世界里是根本不存在的。无依赖性为Go语言程序的发布和部署带来了极大的方便，但同时也导致Go语言程

序可执行文件的体积通常都比较大。为此，任何未被用到的包，在Go语言中是绝对被禁止导入的。

Go语言不区分平台，允许在不同操作系统和硬件架构间交叉编译

```
> go env -w GOOS=linux
> go env -w GOARCH=amd64
```

知识讲解

- 通过环境变量GOOS指定目标操作系统
  - windows
  - darwin
  - linux
- 通过环境变量GOARCH指定目标硬件架构
  - 386
  - amd64
  - arm

8

Go语言本身是平台中立的，允许在不同操作系统和硬件架构间交叉编译，即在A平台上编译生成可在B平台上运行的可执行文件。比如在Windows上执行命令：

```
1 > go env -w GOOS=linux
```

即将环境变量GOOS的值设置为linux，此后编译生成的都是可运行在Linux系统上的可执行程序。除了将目标操作系统指定为Linux以外，还可以用darwin表示Mac OS系统，或用windows表示Windows系统。除了能够指定目标操作系统以外，还可以指定目标硬件架构，比如在安装了32位Intel处理器的计算机上执行命令：

```
1 > go env -w GOARCH=amd64
```

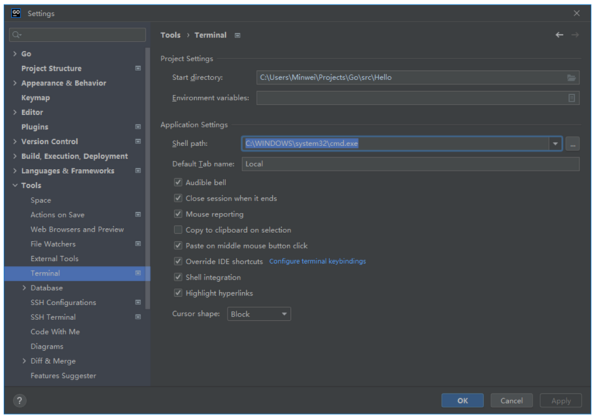
即将环境变量GOARCH的值设置为amd64，此后编译生成的都是可运行在64位Intel处理器上的可执行程序。当然也可以用386、arm或arm64表示32位Intel处理器、32位ARM处理器或64位ARM处理器等硬件架构。

设置终端

在使用GoLand集成开发环境开发Go语言程序的过程中，有时我们也会希望执行一些系统命令。如果每次都要切换到独立的终端或控制台窗口，会非常不方便。为此GoLand集成了终端窗格，就位于主界面的下方，这样完全无需离开集成开发环境，就可以很方便地执行系统命令。但GoLand默认提供的终端程序未必是开发者习惯或喜爱使用的，为此有时我们需要为终端选择一个非默认的程序。

为GoLand设置终端应用程序路径

知识讲解



- GoLand->File->Settings...->Tools->Terminal->Shell path: <终端应用程序路径>

选择“File>Settings...”菜单项，打开设置对话框。左侧选择“Tools>Terminal”。在右侧“Shell path”编辑框中填入终端应用程序的路径，如：C:\WINDOWS\system32\cmd.exe。

常用命令

11

除了通过集成开发环境提供的按钮、菜单完成编译、链接、运行以外，我们也可以在终端窗口或终端窗格中，直接执行Go语言开发工具提供的命令，在命令行完成同样的工作。

知识讲解

编译链接生成可执行程序

```
> go build
> go build Hello
> go build -o Hello.exe Hello
```

解释执行.go文件

```
> go run main.go
```

将可执行程序安装到GOPATH\bin目录下

```
> go install
> go install Hello
```

12

比如在Hello项目的项目目录下执行命令：

```
1 > go build
```

编译链接当前目录下的所有Go文件，在当前目录下生成可执行文件Hello.exe。也可以在任意路径下执行命令：

```
1 > go build Hello
```

编译链接GOPATH\src\Hello目录下的所有Go文件，在当前目录下生成可执行文件Hello.exe。通过“-o”选项可以指定编译链接输出的可执行文件路径：

```
> go build -o Hello.exe Hello
```

在调试阶段，我们可能更希望不做编译链接，直接执行Go语言程序，比如：

```
go run main.go
```

在部署阶段，我们可以将最终生成的可执行程序安装到GOPATH\bin目录下。比如在Hello项目的项目目录下执行命令：

```
> go install
```

或者在任意路径下执行命令：

```
> go install Hello
```

都会在GOPATH\bin目录下得到Hello.exe文件。

更多精彩，敬请期待

谢谢大家，我们下节课再见！