需求：

产品：编写一个go计算器

- 加法
- 减法
- 乘法
- 除法

项目划分开发阶段：

P1:阶段

时间：2019年10 ~ 2019年12月31

功能：

1. 加法
2. 减法

P2:阶段

时间：2020年2 ~ 2020年4月

功能：

1. 乘法
2. 除法

计算器开发

1. 加法功能

calc/add.go

```go
package calc

func Add(a, b int) int {
    return a + b
}
```

main.go

```go
package main

import (
    "fmt"
    "go5期/gitTest/calc"
)

func main() {
    res := calc.Add(10, 20)

    fmt.Println("Add(10 ,20) :", res)
}
```

2. 创建仓库

```
git init
```

1. 将当前的目录变成一个代码仓库
2. 可以有很多个仓库，多个仓库之间是独立的，无法相互提交代码

3.查看当前状态

```
git status
```

此时，会看到main.go ， calc是红色标识的，说明需要处理

4. git追踪代码（暂存区）

```
git add main.go calc
```

5.查看当前状态

```
git status
```

此时，main.go，calc会变成绿色的，说明已经添加到暂存区

6. 提交代码到本地仓库

```
git commit
```

第一次提交，可能会遇到下面的提示

问题1：设置用户信息：

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

请替换为自己的名字和邮箱，告知git系统提交人的信息

问题2：编辑器不是vim，需要执行如下命令，配置成vim后，重新commit

```
git config --global core.editor "vim"
或
export GIT_EDITOR=vim
```

此时会弹出vim界面，需要添加本次的注释，保存退出

7.查看当前状态

```
git status
```

main.go和calc不见了，红、绿都不见了，说明本次提交成功了。

当前问题：

1. git命令是可以自动补全，如果不能补全，需要配置一下（git-bash-complete.sh，自己查找）
    1. 不要全部手巧
    2. 慢，容易出错
2. 如果第一次提交，会要求配置提交人的信息，这样后续可以跟踪某个代码时谁提交的，用于问责
    1. git blame <文件>

    ```
    git config --global user.email "duke@itcast.cn"
    git config --global user.name "duke"
    ```

3. git一般使用vim作为commit时的编辑器，如果不是，请配置一下：

    ```
    git config --global core.editor "vim"
    或
    export GIT_EDITOR=vim
    ```

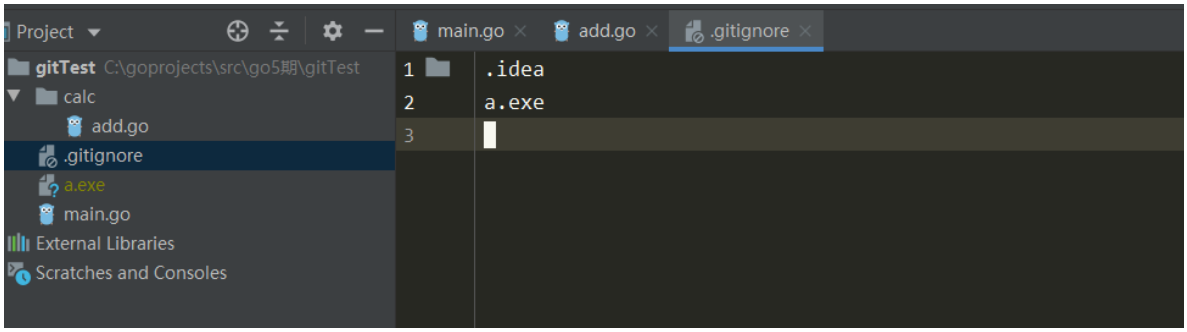
4. 执行git init时，他会在当前的目录创建一个代码仓库，不同的文件夹仓库不同。多个仓库之间是相互独立的
    1. 前端仓库
    2. 后端的仓库
        1. 模块1仓库 ==》 A开发人员
        2. 模块2仓库==》 B开发人员

8. .gitignore忽略文件

在当前文件夹下存在.ideal的文件夹，它并不是我们的代码，我们不想提交，也不想总看见它的提示，为了避免误提交，可以将这个文件（夹）添加到一个特定的文件中：.gitignore

```
.idea
```

如图：



需要将.gitignore添加到仓库

1. git add .gitignore
2. git commit .gitignore

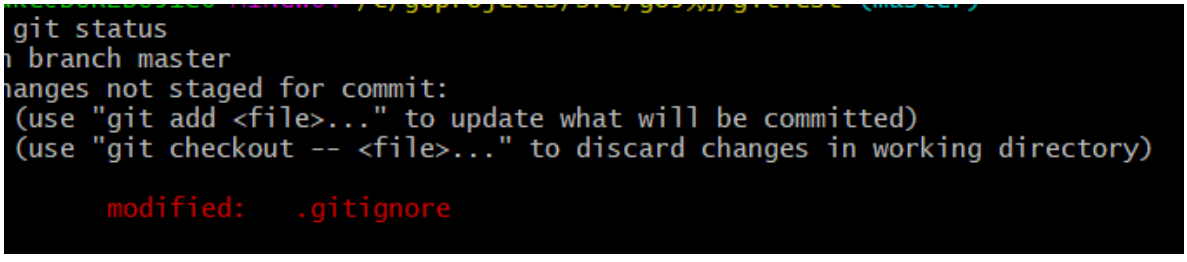
两种添加commit备注的方式：

1. git commit ,回车，进入vim模式，填写描述 =》适合于注释较多的情况
2. git commit -m "添加.gitignore文件" ==》 适合备注较少的情况

9. 修改后的文件提交/丢弃

1. git add <文件> ==>提交本次修改
2. git checkout <文件> ==》丢弃本次修改

在.gitignore中修改内容，查看文件状态，会发现提示文件被修改：



如果这是一个误操作，使用status发现文件被修改了。

```go
import (
    "fmt"
    "go5期/gitTest/calc"
)

func main() {
    res := calc.Add( a: 10,  b: 20);
    fmt.Println( a...: "Add(10 ,20) :", res)
}
```

```
duke@DUKEDU51C6 MINGW64 /c/goprojects/src/go5期/gitTest (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   .gitignore
        modified:   main.go

no changes added to commit (use "git add" and/or "git commit -a")
```

9.如何知道修改了哪些内容

```
git diff ==> 查看所有被修改的文件，打印对比信息
git diff main.go ==>指定文件查看
```

```
duke@DUKEDU51C6 MINGW64 /c/goprojects/src/go5期/gitTest (master)
$ git diff main.go
warning: LF will be replaced by CRLF in main.go.
The file will have its original line endings in your working directory
diff --git a/main.go b/main.go
index 5c99f17..42fc9fc 100644
--- a/main.go
+++ b/main.go
@@ -6,7 +6,6 @@ import (
 )

 func main() {
-       res := calc.Add(10, 20)
-
+       res := calc.Add(10, 20);
        fmt.Println("Add(10 ,20) :", res)
 }

duke@DUKEDU51C6 MINGW64 /c/goprojects/src/go5期/gitTest (master)
```

如果想提交本次的修改，在使用git add 提交。

```
git add main.go
```

如果这是误操作，或者想将当前的修改丢弃掉，使用git checkout <文件名字>

```
git checkout main.go
```