

# 第一章 环境搭建

作者：少林

# 目录

1.课程介绍

2. Go发展历史

3. 环境搭建

4. 第一个程序

5. 常用工具介绍

# 课程介绍

## 1. 课程名字

**Golang从入门到精通**

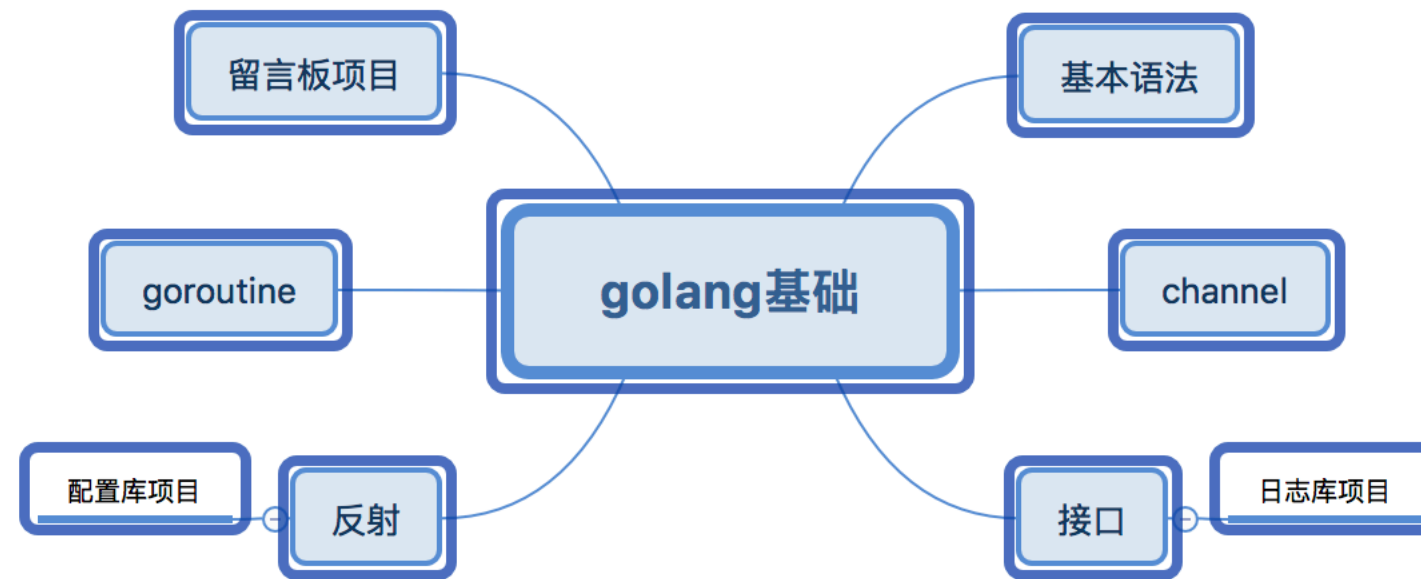
## 2. 课程内容

第一部分：Golang编程基础

第二部分：Golang实战

第三部分：Golang高级编程

# 课程介绍



# 课程介绍



# 课程介绍



# Golang发展历史

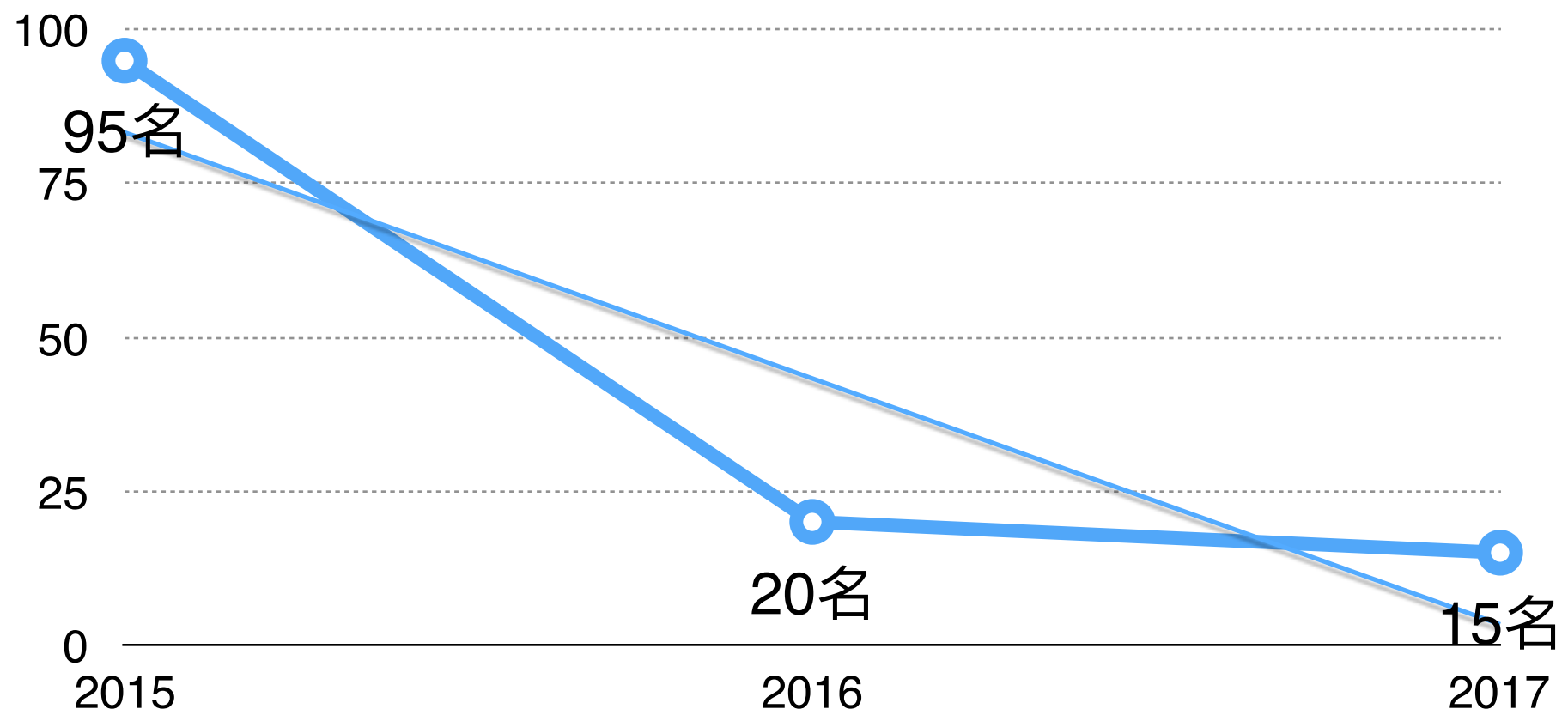
## 1. 诞生历史

- a. 诞生与2006年1月2号下午15点4分5秒
- b. 2009发布并正式开源
- c. 2012年第一个正式版本Go 1.0发布
- d. 截止到2017年8月24号，Go 1.9版本正式发布

# Golang发展历史

## 2. 发展情况

Go语言排名变化





# 开发环境搭建

## 1. Go安装

- a. 打开Golang官网, <https://golang.org/dl/>
- b. 选择window安装包进行下载
- c. 双击安装

**<https://github.com/pingguoxueyuan/gostudy/tree/master/listen1>**

# Golang环境搭建

## 2. Git 工具安装

a. 高效的、分布式的代码管理工具

b. 下载地址：<https://git-scm.com/downloads>

c. 点击安装包进行安装

# Golang环境搭建

## 3. IDE搭建 (vscode)

- a. 打开网址: <https://code.visualstudio.com/>
- b. 根据操作系统选择对应的安装包
- c. 点击安装包进行安装 (linux直接解压)
- d. 选择查看-》扩展-》搜索go, 安装第二个

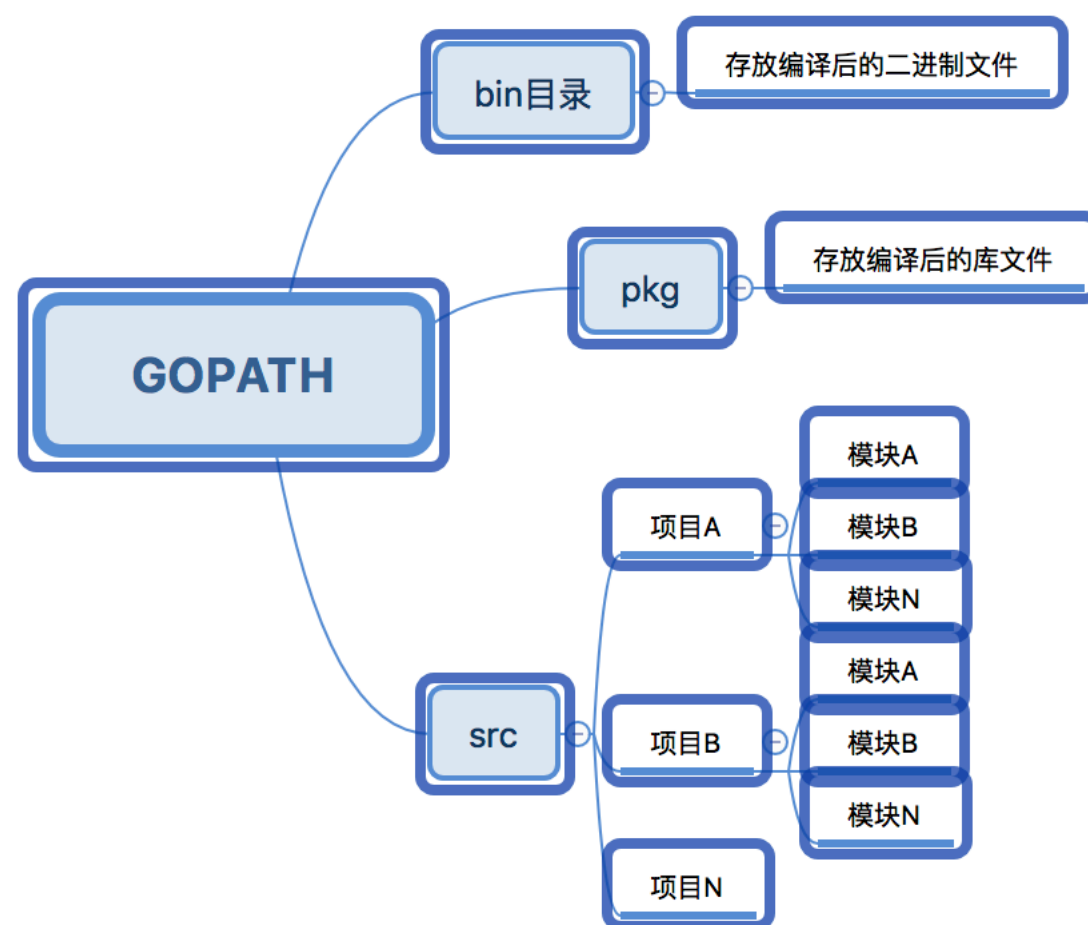
# Golang环境搭建

## 3. GOPATH设置

- a. GOPATH是一个环境变量，用来表明go项目的存放路径
- b. GOPATH路径最好只设置一个，所有的项目都放到GOPATH的src目录下
- c. GOPATH的目录结构介绍

# Golang目录结构介绍

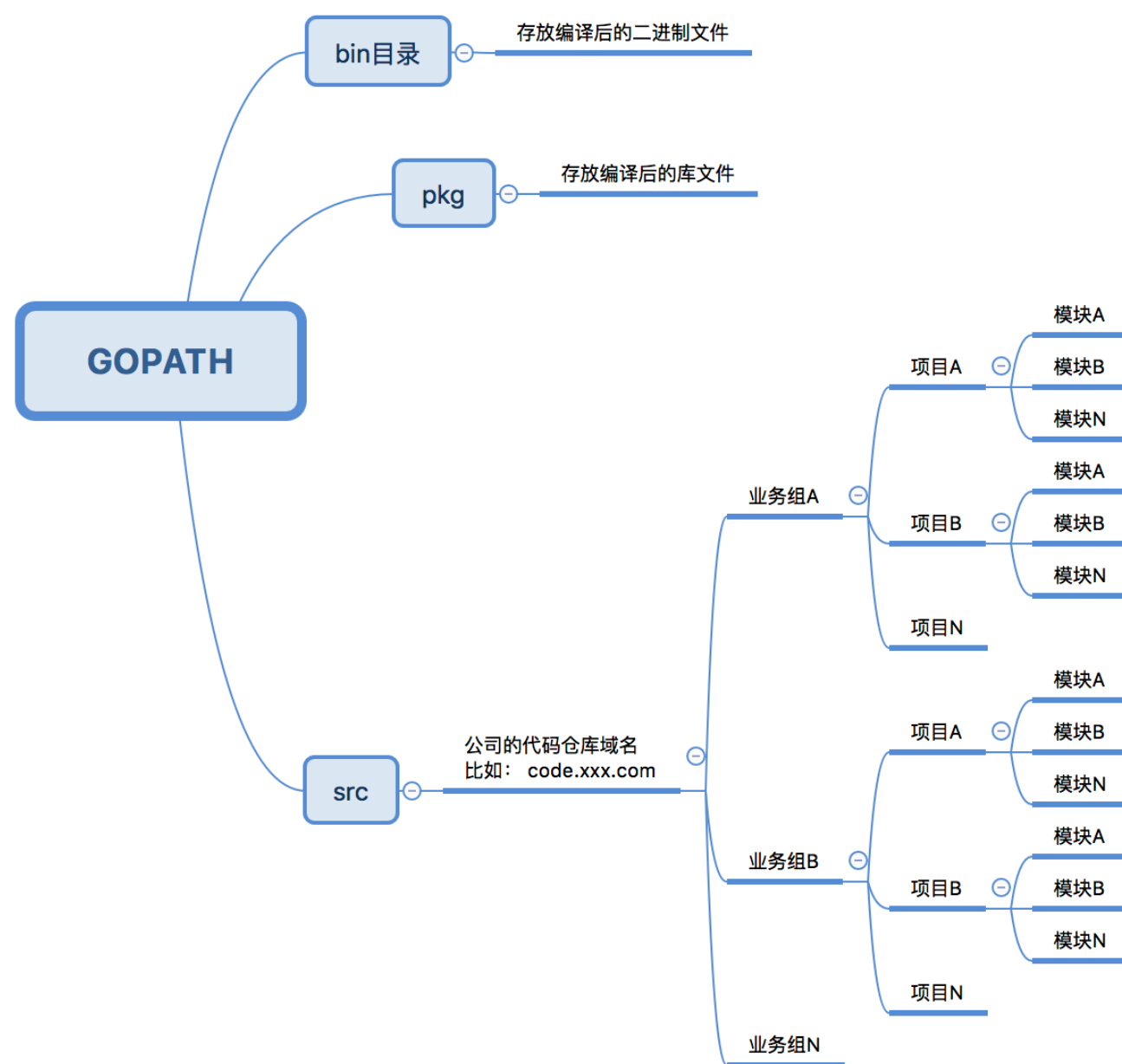
## 1. 目录结构划分一



适用个人

# Golang目录结构介绍

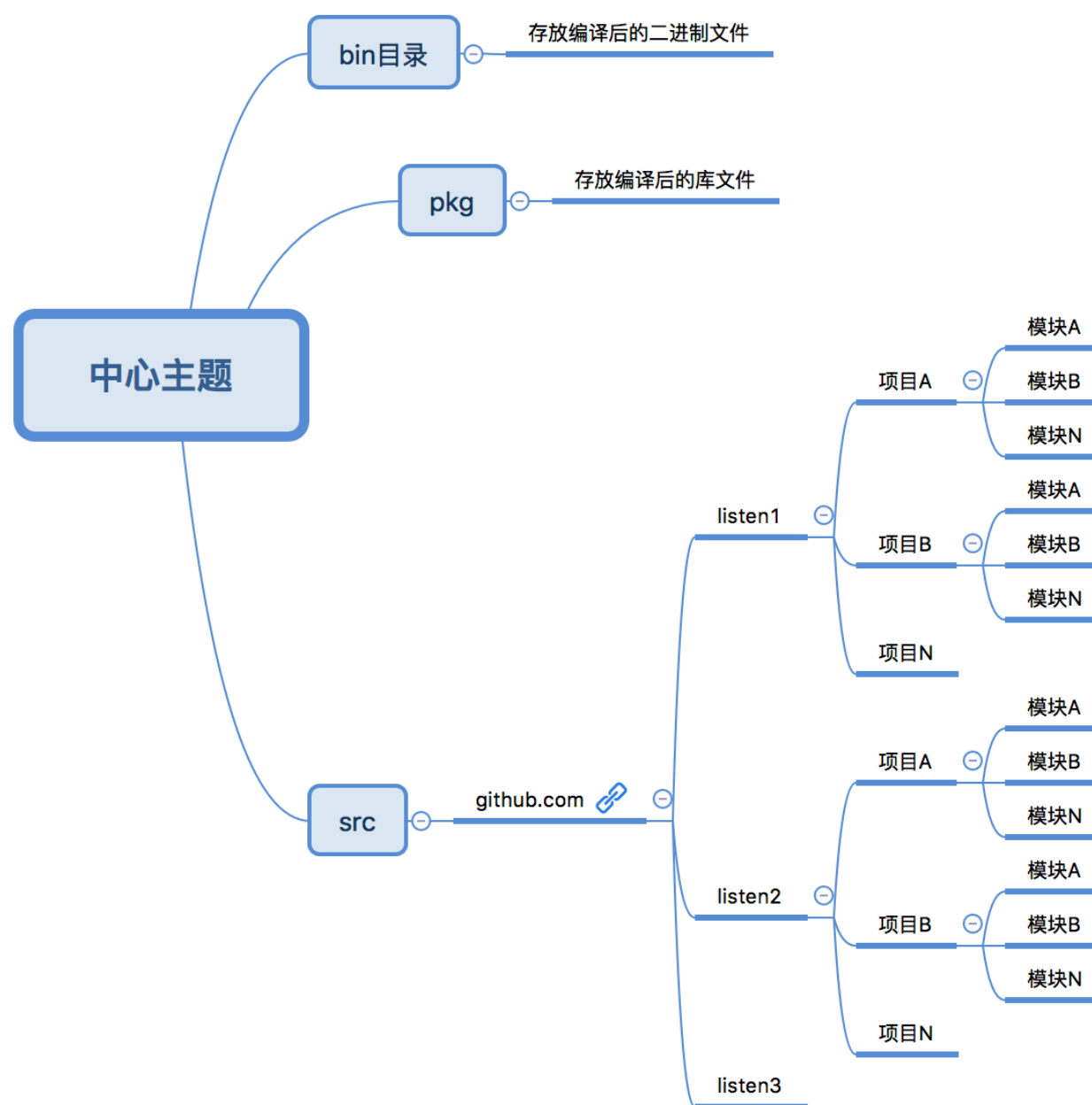
## 2. 目录结构划分二



适用公司

# Golang目录结构介绍

## 3. 本课程的目录结构



# 第一个golang程序

## 1. 演示hello程序的代码



# 包的概念

1. 和python一样，把相同功能的代码放到一个目录，称之为包
2. 包可以被其他包引用
3. main包是用来生成可执行文件，每个程序只有一个main包
4. 包的主要用途是提高代码的可复用性

# Go 基本命令介绍

## 1. 基本命令

- a. `go run` 快速执行go文件，就像执行脚本一样
- b. `go build` 编译程序，生成二进制可执行文件
- c. `go install` 安装可执行文件到bin目录

# Go 基本命令介绍

## 2. 基本命令

- a. `go test` 执行单元测试或压力测试
- b. `go env` 显示go相关的环境变量
- c. `go fmt` 格式化源代码

# Go 基本命令介绍

## 3. Go程序结构

- a. go源码按package进行组织，并且package要放到非注释的第一行
- b. 一个程序只有一个main包和一个main函数
- c. main函数是程序的执行入口

# Go 基本命令介绍

## 4. 注释

a. 单行注释 //

b. 多行注释 /\* \*/

# golang语言特性

## 1. 垃圾回收

- a. 内存自动回收，再也不需要开发人员管理内存
- b. 开发人员专注业务实现，降低了心智负担
- c. 只需要new分配内存，不需要释放

# golang语言特性

## 2. 天然并发

- a. 从语言层面支持并发，非常简单
- b. goroutine，轻量级线程，创建成千上万个goroutine成为可能
- c. 基于CSP（Communicating Sequential Process）模型实现

```
func calc() {  
    //大量计算  
}  
  
func main() {  
    go calc()  
}
```

# golang语言特性

## 3. channel

- a. 管道，类似unix/linux中的pipe
- b. 多个goroutine之间通过channel进行通信
- c. 支持任何类型



# golang语言特性

## 4. 多返回值

### a. 一个函数返回多个值

QA