**目录**

[一、知识库逻辑与物理设计基础规范（张良韬） 1](#_Toc507933371)

[1.逻辑设计 1](#_Toc507933372)

[2.物理设计 2](#_Toc507933373)

[2.1 SWRL规则语言 4](#_Toc507933374)

[2.2 SPARQL查询语言 5](#_Toc507933375)

[二、云平台及应用系统设计基础规范（江超） 6](#_Toc507933376)

[1.云平台配置 6](#_Toc507933377)

[2.服务器注意协调端口： 6](#_Toc507933378)

[3. MySQL数据库调用配置 7](#_Toc507933379)

[4.其他注意事项 7](#_Toc507933380)

[三、主流的知识库开发语言和工具（刘婷） 8](#_Toc507933381)

[1 主流的知识库开发语言 8](#_Toc507933382)

[1.1 Python 8](#_Toc507933383)

[1.2 Java 9](#_Toc507933384)

[1.3 C 10](#_Toc507933385)

[1.4 C# 11](#_Toc507933386)

[1.5 C++ 12](#_Toc507933387)

[1.6 PHP 12](#_Toc507933388)

[1.7 JavaScript 13](#_Toc507933389)

[1.8 Swift 14](#_Toc507933390)

[1.9 Scala 15](#_Toc507933391)

[1.10 GO 16](#_Toc507933392)

[1.11 Rust 16](#_Toc507933393)

[1.12 HTML5 17](#_Toc507933394)

[2 主流的知识库开发工具 17](#_Toc507933395)

[2.1 Java工具编辑 17](#_Toc507933396)

[2.2 Python开发工具 18](#_Toc507933397)

[2.3 常用工具 19](#_Toc507933398)

# 一、知识库逻辑与物理设计基础规范（张良韬）

## 1.逻辑设计

建立知识库概念模型，抽取相关知识领域内的概念以及概念之间的联系，通过某种计算机可读的语言为这些概念和联系进行明确定义，如RDF三元组形式等。以本体为例，本体是一个六元组，其中Ｃ是概念的集合，CR是概念属性的集合，Ｒ是关系的集合，AR是关系属性的集合，Ｉ是实例的集合，Ｘ是公理的集合。以下对各项展开解释：

（1）概念（C）：概念也称为类，从语义上讲它是对现实世界中个体的抽象，表示的是个体的集合，其定义一般包括概念的名称及其自然语言描述。如“drug”即可以为一个概念。

（2）概念的属性（CR）：概念之间之所以有差异是因为他们有着不同的属性，并因此对应着不同个体的集合，因此概念的属性集合又称为概念的内涵，而它所对应的个体集合成为概念的外延。如“drug”的状态如固态、液态等即为“drug”的属性。

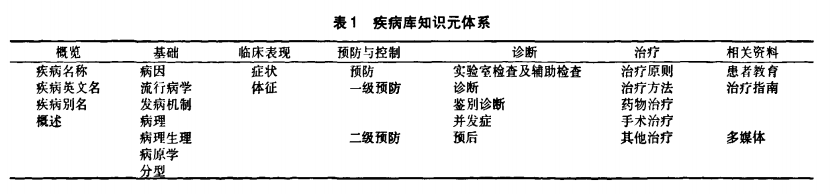
（3）关系（R）：一个关系通常包含定义域和值域两部分，送两部分限定了该关系适用的范围。在本体中，关系的定义域通常是一个概念，而值域可以是概念，也可以是具体的数值（如字符串和整数等），当值域是数值时，关系便退化为属性，所以可以认为属性是一种特殊的关系。如果只考虑值域是概念的情况，关系集合R中的每个关系r(ci,cj)就表示概念ci和cj之间的二元关系。需要注意的是，此时这个关系智能表示ci和cj所对应的实例可能存在关系r(ci,cj)，而并非这两个概念下所有的实例都一定有这样的关系。如关系“治疗”可以表示“drug”和“disease”两个概念之间的关系。

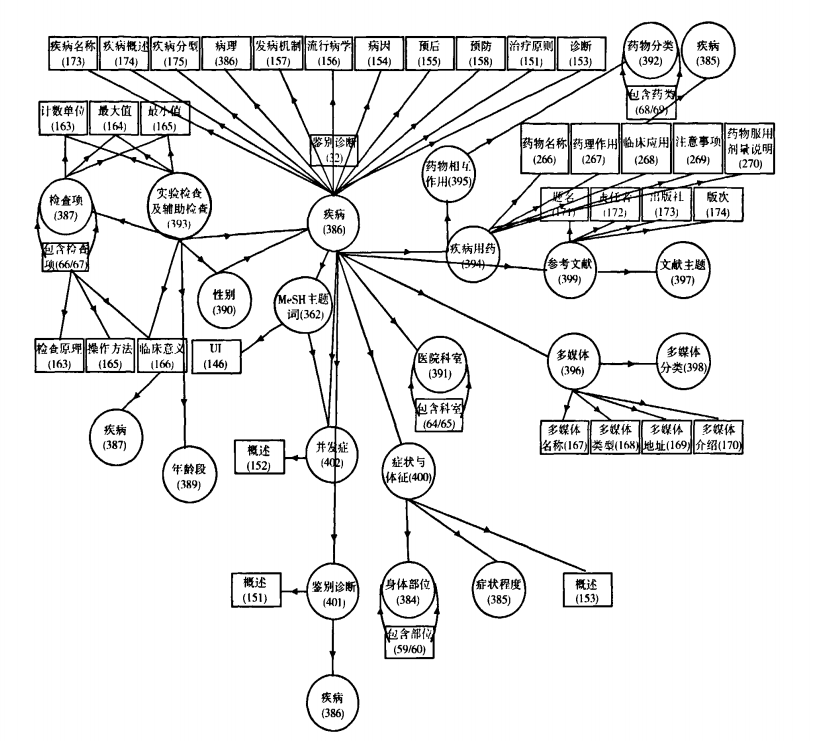
（4）关系的属性（AR）：关系的属性描述了对关系的限制，例如一个表示亲生母亲的关系，记作＂hasmother＂，其定义域和值域都是表示人的概念＂Person＂，我们可通过＂hasmother＂的基数属性规定一个人只有一个亲生母亲。

（5）实例（I）：一个实例是现实世界中唯一的个体，它对应着本体中一个或多个概念，并为概念的属性赋予了具体的属性值。由于现实世界中个体的数量无法确定，所以实例相对于本体的其他成分来说是动态的，因此，本体建模时一般不考虑或只考虑少数重要的实例，只有在本体与实际应用相结合的时候，才需要将特定应用内的个体作为示例添加到本体中。如阿莫西林是“drug”的一个实例

（6）公理（X）：公理集合Ｘ中的每条公理都代表一条领域知识中的永真断言，例如声明表示男人和女人的概念＂Man＂和＂Woman＂是不相交的断言。

逻辑设计可以得到知识库知识元体系及关系网络等：





（圆圈表示概念，方框表示概念属性，箭头表示关系可以写上具体关系如药物对疾病的关系为“治疗”）

## 2.物理设计

使用本体作为知识库构建的主要工具，其技术可行性为：

（1）知识的明示化，本体对描述领域知识的相关概念及概念之间的联系都给出了明确的定义。

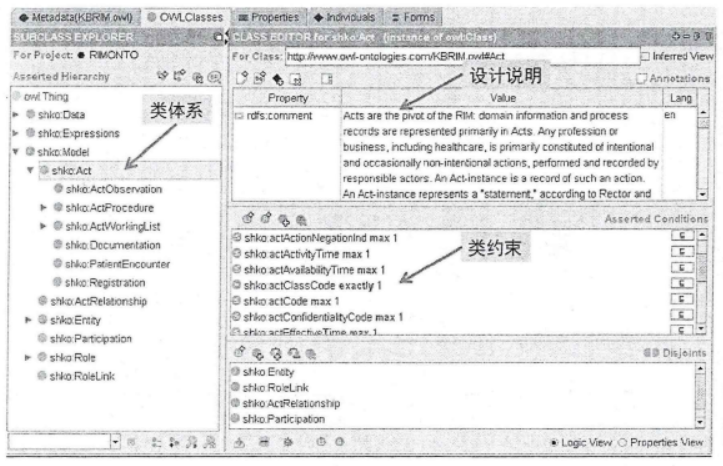
（2）知识的形式化，本体基于描述逻辑和框架模型，具有计算机可理解的表现形式。

（3）知识的规范化，本体提供了引用某个领域知识的标准方式。

（4）知识的共享化，本体表达的领域知识反映了不同应用对该领域的共同认知

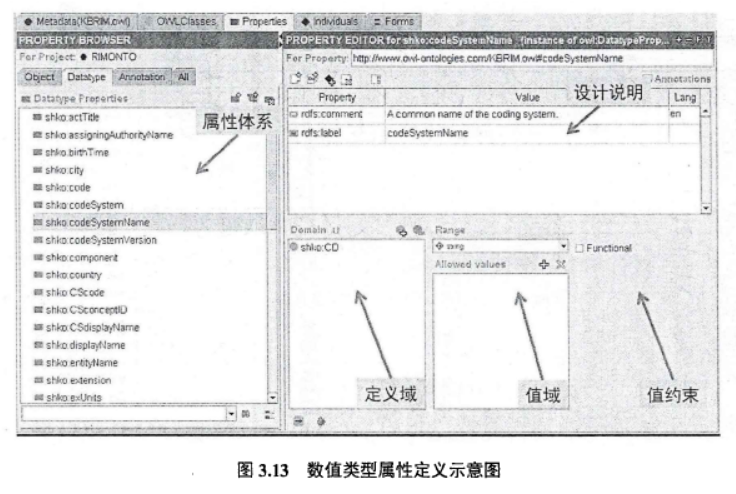
（5）知识的推理化，本体可以通过规则语言形成推理规则，通过Jena语义框架可以实现对知识的推理、查询等等。

采用protege作为本体的开发工具进行类、属性的定义和实例的创建。类定义主要包括类的层级体系设计、公理约束以及设计说明，如下所示：



属性定义包括属性的层级体系设计、定义域、值域、值约束以及设计说明五部分，对象类型属性和数值类型属性的定义如下：





知识库规则主要用于对本体的概念模型进行推理和查询，以jena语义网规则表示语言和SPARQL语义网查询语言分别进行实现。

### 2.1 SWRL规则语言

SWRL（Semantic Web Rule Language）是由以语义的方式呈现规则的一种语言，SWRL的规则部分概念是由RuleML所演变而来，再结合OWL本体论形成。在RuleML中以head表示推理结果，body表示推理前提的基本形态被保留在SWRL中，所以SWRL可视为规则和本体论的结合。透过两者的组合可使撰写规则时，直接使用本体论中所描绘的关系和词汇，这些类别之间的关系本来可能还需要额外的法则描述，但SWRL中可以直接使用本体论描述。例如在本体论中定义了下列关系：

hasParent( x2, x1 )

hasBrother( x1, x3 )

透过本体论描述我们可以知道x1是x2的父母，x1有一个兄弟是x3。此时欲使用SWRL设计一条规则说明x2和x3之间的关系，即在何种情况下x2和x3有叔叔的关系，如下：

**Body**

hasParent( x2, x1 )

hasBrother( x1, x3 )

**Head**

hasUncle( x2, x3 )

此规则所表达的具体语义是：当x1是x2的父母且x1和x3是兄弟时，则可以推理得出x3是x2的叔叔。

### 2.2 SPARQL查询语言

SPARQL (SPARQL Protocol and RDF Query Language)，是为RDF开发的一种查询语言和数据获取协议，它是为W3C所开发的RDF数据模型所定义，但是可以用于任何可以用RDF来表示的信息资源。SPARQL 协议和 RDF 查询语言（SPARQL）于2008年1月15日正式成为一项W3C推荐标准。SPARQL构建在以前的 RDF 查询语言（例如 rdfDB、RDQL 和 SeRQL）之上，拥有一些有价值的新特性。格式如下

SELECT \* WHERE {

？city rdf：type < http://dbpedia.org/class/yago/WikicatCitiesInTexas >

}

运行查询，将获得一个包含德克萨斯州相当长的城市名单的表格。

# 二、云平台及应用系统设计基础规范（江超）

## 1.云平台配置

|  |  |  |  |
| --- | --- | --- | --- |
| 操作系统 | CentOS 6.7 64位 | CPU | 4核 |
| 内存 | 8 G | IP地址 | 218.197.150.149 |
| 远程管理地址段 | 202.114.66.51/56 | 远程管理协议 | ssh |
| 账号 | root | 密码 | lumen2017 |

## 2.服务器注意协调端口：

关系型数据库：

mySQL：3306

SQL Server：1433

NOSQL非关系型数据库

MongoDB：27017

Redis：6379

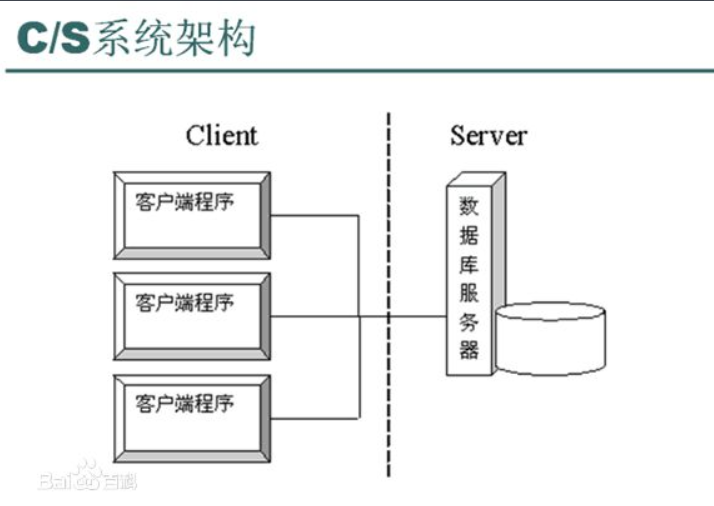
Memcached：11211

主要有两种应用模式：

**（1）C/S模式**

Client/Server (客户机/服务器) 结构，是大家熟知的软件系统体系结构，通过将任务合理分配到Client端和Server端，降低了系统的通讯开销，需要安装客户端才可进行管理操作。本模式常见的形式为：计算机软件/移动设备APP。

客户端和服务器端的程序不同，用户的程序主要在客户端，服务器端主要提供数据管理、数据共享、数据及系统维护和并发控制等，客户端程序主要完成用户的具体的业务。



其通讯模式为：客户端向服务器发起请求，服务器对请求进行响应，将处理结果返回给客户端。

采用此模式开发需要注意协调服务器端口，避免冲突。

**（2）B/S（浏览器/服务器）模式**

即Browser/Server (浏览器/服务器) 结构，是随着Internet技术的兴起，对C/S结构的一种变化或者改进的结构。在这种结构下，用户界面完全通过WWW浏览器实现。

该模式以web浏览器作为客户端，这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器，通过服务器同数据库进行数据交换。

使用本模式开发的项目发布可参考以下步骤：

* + 1. 将web项目发布到Tomcat 服务器上的三种方法 - KevinLM 工作室 - CSDN博客：http://blog.csdn.net/kevinwuwenboa/article/details/52895813
    2. 使用Tomcat的web项目，统一使用端口80/8080
    3. 使用IDE将项目打包成war文件后上传到服务器tomcat/webapps目录下，运行命令： ./catalina.sh run即可执行tomcat。
    4. 浏览器输入：218.197.150.149:8080/项目名，即可访问项目网站。

## 3. MySQL数据库调用配置

<driver-url>: jdbc:mysql://localhost:3306/数据库

<user>: root

<password>: lumen2017

## 4.其他注意事项

知识处理的结果表现形式为知识库本体Owl文件，应用形式为利用owl文件围绕各自项目需求，基于JAVA或Python开发web项目，并提供知识库集成的API（各自owl的调用方法），为以后知识库集成做准备。

# 三、主流的知识库开发语言和工具（刘婷）

## 1 主流的知识库开发语言

### 1.1 Python

Python是一种面向对象的解释型计算机程序设计语言，由荷兰人Guido van Rossum于1989年发明，第一个公开发行版发行于1991年。

Python是纯粹的自由软件，源代码和解释器CPython遵循 GPL(GNU General Public License)协议。Python语法简洁清晰，特色之一是强制用空白符(white space)作为语句缩进。

Python具有丰富和强大的库。它常被昵称为胶水语言，能够把用其他语言制作的各种模块（尤其是C/C++）很轻松地联结在一起。常见的一种应用情形是，使用Python快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更合适的语言改写，比如3D游戏中的图形渲染模块，性能要求特别高，就可以用C/C++重写，而后封装为Python可以调用的扩展类库。需要注意的是在您使用扩展类库时可能需要考虑平台问题，某些可能不提供跨平台的实现。

**优点**

（1）简单。Python是一种代表简单主义思想的语言。阅读一个良好的Python程序就感觉像是在读英语一样。它使你能够专注于解决问题而不是去搞明白语言本身。

（2）易学。Python极其容易上手，因为Python有极其简单的说明文档。

（3）速度快。Python 的底层是用 C 语言写的，很多标准库和第三方库也都是用 C 写的，运行速度非常快。

（4）免费、开源。Python是FLOSS（自由/开放源码软件）之一。使用者可以自由地发布这个软件的拷贝、阅读它的源代码、对它做改动、把它的一部分用于新的自由软件中。FLOSS是基于一个团体分享知识的概念。

（5）高层语言。用Python语言编写程序的时候无需考虑诸如如何管理你的程序使用的内存一类的底层细节。

（6）可移植性.由于它的开源本质，Python已经被移植在许多平台上（经过改动使它能够工作在不同平台上）。这些平台包括Linux、Windows、FreeBSD、Macintosh、Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acom RISC OS、VxWorks、PlayStation、Sharp Zaurus、Windows CE、PocketPC、Symbian以及Google基于linux开发的android平台。

（7）解释性。一个用编译性语言比如C或C++写的程序可以从源文件（即C或C++语言）转换到一个你的计算机使用的语言（二进制代码，即0和1）。这个过程通过编译器和不同的标记、选项完成。运行程序的时候，连接/转载器软件把你的程序从硬盘复制到内存中并且运行。而Python语言写的程序不需要编译成二进制代码。你可以直接从源代码运行 程序。在计算机内部，Python解释器把源代码转换成称为字节码的中间形式，然后再把它翻译成计算机使用的机器语言并运行。这使得使用Python更加简单。也使得Python程序更加易于移植。

（8）面向对象。Python既支持面向过程的编程也支持面向对象的编程。在“面向过程”的语言中，程序是由过程或仅仅是可重用代码的函数构建起来的。在“面向对象”的语言中，程序是由数据和功能组合而成的对象构建起来的。

（9）可扩展性.如果需要一段关键代码运行得更快或者希望某些算法不公开，可以部分程序用C或C++编写，然后在Python程序中使用它们。

（10）可嵌入性.可以把Python嵌入C/C++程序，从而向程序用户提供脚本功能。

（11）丰富的库：Python标准库确实很庞大。它可以帮助处理各种工作，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV文件、密码系统、GUI（图形用户界面）、Tk和其他与系统有关的操作。这被称作Python的“功能齐全”理念。除了标准库以外，还有许多其他高质量的库，如wxPython、Twisted和Python图像库等等。

（12）规范的代码：Python采用强制缩进的方式使得代码具有较好可读性。而Python语言写的程序不需要编译成二进制代码。

**缺点**

（1）单行语句和命令行输出问题。很多时候不能将程序连写成一行，如import sys;for i in sys.path:print i。而perl和awk就无此限制，可以较为方便的在shell下完成简单程序，不需要如Python一样，必须将程序写入一个.py文件。

（2）独特的语法。这也许不应该被称为局限，但是它用缩进来区分语句关系的方式还是给很多初学者带来了困惑。即便是很有经验的Python程序员，也可能陷入陷阱当中。

（3）运行速度慢。这里是指与C和C++相比。

### 1.2 Java

Java是一门面向对象编程语言，不仅吸收了C++语言的各种优点，还摒弃了C++里难以理解的多继承、指针等概念，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程，主要有Java Web、Java EE、Java SE等

Java具有简单性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、多线程、动态性等特点。Java可以编写桌面应用程序、Web应用程序、分布式系统和嵌入式系统应用程序等。

**优点：**

（1）简单、安全、稳定、跨平台

（2）开源。一次编写，到处运行。

（3）系统的多平台支持。

（4）强大的可伸缩性。

（5）多样化和功能强大的开发工具支持。

**缺点：**

（1）需要运行环境、不适合开发桌面应用程序

（2）复杂性。由于为了跨平台的功能，为了极度的伸缩能力，所以极大的增加了产品的复杂性。

（2）Java的运行速度是用class常驻内存来完成的，所以它在一些情况下所使用的内存比起用户数量来说确实是“最低性能价格比”了。从另一方面，它还需要硬盘空间来储存一系列的java文件和.class文件，以及对应的版本文件。

（3）跨平台性，虽然是很好的优势，可以让JAVA在任何的系统中运行，但是前提是该系统要安装JAVA虚拟机，这样导致一些JAVA编写的桌面程序无法在一般的用户（没有安装JAVA虚拟机的用户）上使用

（4）JAVA的身后是SUN公司，提供及时的API更新，但是过多的版本导致JAVA的向下兼容性很差。而SUN公司在J2EE上的标准定义的并不是完美，导致在J2EE的开发中多数使用开源框架，开源框架是好东西，但是更新速度并不快。

（5）都说面向对象的语言先进与面向过程，JAVA是OOP语言，但是它适合的是BS系统，在WEB项目中JAVA的实力毋庸置疑，但是转到了底层的程序却无法同C++抗衡。

（6）JAVA的最大优点是开源，但是这往往也成了最大的缺陷了，无法定义一个好的标准使得开发时使用了框架，在新的程序员来到公司时必须先了解框架，延缓了开发的时间。

### 1.3 C

C语言是一门通用计算机编程语言，应用广泛。C语言的设计目标是提供一种能以简易的方式编译、处理低级存储器、产生少量的机器码以及不需要任何运行环境支持便能运行的编程语言。

尽管C语言提供了许多低级处理的功能，但仍然保持着良好跨平台的特性，以一个标准规格写出的C语言程序可在许多电脑平台上进行编译，甚至包含一些嵌入式处理器（单片机或称MCU）以及超级电脑等作业平台。

**优点：**

（1）简洁紧凑、灵活方便。C语言一共只有32个关键字，9种控制语句，程序书写形式自由，区分大小写。把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。[12]

（2）运算符丰富。C语言的运算符包含的范围很广泛，共有34种运算符。C语言把括号、赋值、强制类型转换等都作为运算符处理。从而使C语言的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其它高级语言中难以实现的运算。[12]

（3）数据类型丰富。C语言的数据类型有：整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能用来实现各种复杂的数据结构的运算。并引入了指针概念，使程序效率更高。

（4）表达方式灵活实用。C语言提供多种运算符和表达式值的方法，对问题的表达可通过多种途径获得，其程序设计更主动、灵活。它语法限制不太严格，程序设计自由度大，如对整型量与字符型数据及逻辑型数据可以通用等。

（5）允许直接访问物理地址，对硬件进行操作。由于C语言允许直接访问物理地址，可以直接对硬件进行操作，因此它既具有高级语言的功能，又具有低级语言的许多功能，能够像汇编语言一样对位（bit）、字节和地址进行操作，而这三者是计算机最基本的工作单元，可用来写系统软件。

（6）生成目标代码质量高，程序执行效率高。C语言描述问题比汇编语言迅速，工作量小、可读性好，易于调试、修改和移植，而代码质量与汇编语言相当。C语言一般只比汇编程序生成的目标代码效率低10%～20%。

（7）可移植性好。C语言在不同机器上的C编译程序，86%的代码是公共的，所以C语言的编译程序便于移植。在一个环境上用C语言编写的程序，不改动或稍加改动，就可移植到另一个完全不同的环境中运行。

（8）表达力强。C语言有丰富的数据结构和运算符。包含了各种数据结构，如整型、数组类型、指针类型和联合类型等，用来实现各种数据结构的运算。C语言的运算符有34种，范围很宽，灵活使用各种运算符可以实现难度极大的运算。

（9）C语言能直接访问硬件的物理地址，能进行位（bit）操作。兼有高级语言和低级语言的许多优点。它既可用来编写系统软件，又可用来开发应用软件，已成为一种通用程序设计语言。

（10）C语言具有强大的图形功能，支持多种显示器和驱动器。且计算功能、逻辑判断功能强大

**缺点：**

（1）C语言的缺点主要表现在数据的封装性上，这一点使得C在数据的安全性上有很大缺陷，这也是C和C++的一大区别。

（2）C语言的语法限制不太严格，对变量的类型约束不严格，影响程序的安全性，对数组下标越界不作检查等。从应用的角度，C语言比其他高级语言较难掌握。也就是说，对用C语言的人，要求对程序设计更熟练一些。

### 1.4 C#

C#是微软公司在2000年6月发布的一种新的编程语言，主要由安德斯·海尔斯伯格（Anders Hejlsberg）主持开发，它是第一个面向组件的编程语言，其源码会编译成msil再运行。它借鉴了Delphi的一个特点，与COM(组件对象模型)是直接集成的，并且新增了许多功能及语法糖，而且它是微软公司.NET windows网络框架的主角。

**优点：**

（1）是一种安全的、稳定的、简单的、优雅的，由C和C++衍生出来的面向对象的编程语言。

（2）在继承C和C++强大功能的同时去掉了一些它们的复杂特性(例如没有宏以及不允许多重继承)

（3）综合了VB简单的可视化操作和C++的高运行效率，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件编程的支持成为.NET开发的首选语言。

**缺点：**

（1）不适用于编写时间急迫或性能非常高的代码，例如一个要运行1000或者1050次的循环，并在不需要这些循环时，立即其管理它们所占的资源。

（2）缺乏性能极高的应用程序所需要的关键功能，包括保证在所代码的特定地方运行的内联函数和析构函数。但这类应用程序非常少。

### 1.5 C++

C++是C语言的继承，它既可以进行C语言的过程化程序设计，又可以进行以抽象数据类型为特点的基于对象的程序设计，还可以进行以继承和多态为特点的面向对象的程序设计。C++擅长面向对象程序设计的同时，还可以进行基于过程的程序设计，因而C++就适应的问题规模而论，大小由之。

C++不仅拥有计算机高效运行的实用性特征，同时还致力于提高大规模程序的编程质量与程序设计语言的问题描述能力。

**优点：**

（1）C++实现了面向对象程序设计。在高级语言当中，处理运行速度是最快的，大部分的游戏软件，系统都是由C++来编写的。

（2）C++语言非常灵活，功能非常强大。如果说C语言的优点是指针，那么C++的优点就是性能和类层次结构的设计。

（3）C++非常严谨、精确和数理化，标准定义很细致。

（4）C++语言的语法思路层次分明、相呼应;语法结构是显式的、明确的。

**缺点：**

（1）语言的过度复杂。一个指针就使C++比关键字数目相同的语言复杂许多，导入模板后各种应用精巧的应用使这门语言进一步复杂化，而C++0x使这复杂性又升华了。

（2）标准库的过度苍白。C++语言已经无法简化，如果编译器能够提供可定制编译也许对这个语言会有帮助。比如：设定某个选项后，C++就变成了有对象的C。设置某个选项后，多重继承就不支持了。可是这样的发展也还是需要很长一段时间的发展的。

### 1.6 PHP

PHP（外文名:PHP: Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。语法吸收了C语言、Java和Perl的特点，利于学习，使用广泛，主要适用于Web开发领域。PHP 独特的语法混合了C、Java、Perl以及PHP自创的语法。它可以比CGI或者Perl更快速地执行动态网页。用PHP做出的动态页面与其他的编程语言相比，PHP是将程序嵌入到HTML（标准通用标记语言下的一个应用）文档中去执行，执行效率比完全生成HTML标记的CGI要高许多；PHP还可以执行编译后代码，编译可以达到加密和优化代码运行，使代码运行更快。

**优点：**

（1）开放源代码。所有的PHP源代码事实上都可以得到。

（2）免费性。和其它技术相比，PHP本身免费且是开源代码。

（3）快捷性。程序开发快，运行快，技术本身学习快。嵌入于HTML：因为PHP可以被嵌入于HTML语言，它相对于其他语言。编辑简单，实用性强，更适合初学者。

（4）跨平台性强。由于PHP是运行在服务器端的脚本，可以运行在UNIX、LINUX、WINDOWS、Mac OS、Android等平台

（5）效率高。PHP消耗相当少的系统资源。

（6）图像处理。用PHP动态创建图像,PHP图像处理默认使用GD2。且也可以配置为使用image magick进行图像处理。

（7）面向对象。在php4,php5 中，面向对象方面都有了很大的改进，php完全可以用来开发大型商业程序。

（8）专业专注。PHP支持脚本语言为主，同为类C语言。

**缺点：**

（1）函数命名不规范。驼峰法和下滑线，传参位置不一

（2）单线程。PHP本身，一直以来php就是个单进程的程序；虽然php的pthreads扩展早就有了。但是它不够稳定，运行运行着就会莫名其妙的自己挂掉；php的扩展都是C写的，这也就意味着任何一个扩展出现线程竞争资源控制问题都能让整个挂掉

（3）核心异步网络不支持（当然在linux只有同步非阻塞网络模型）。却少了这个使得很难开发一个能够承受大并发的网络应用。传统的网络模型和io都阻塞的。这样基本的编程的做法就是一个进程（或者线程）响应一个用户链接请求。因此无法完成像实时网游那样需要成千上万网络连接的任务。尽管php也有Libevent、eio扩展对此算是某种程度上面的弥补，但是感觉都不是那么完善

（4）只支持web开发，不方便做 .exe文件,不方便做桌面应用程序. 不方便做手机程序.

（5）不适合做爬虫、自动运行脚本.科学运算项目,这语言基本构架就不适合，虽然有很多方法实现。

（6）后期维护困难。后期提速空间局限性较大。

### 1.7 JavaScript

JavaScript一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。它的解释器被称为JavaScript引擎，为浏览器的一部分，广泛用于客户端的脚本语言，最早是在HTML（标准通用标记语言下的一个应用）网页上使用，用来给HTML网页增加动态功能。

**优点：**

（1）减少网络传输。在JavaScript这样的用户端脚本语言出现之前，传统的数据提交和验证工作均由用户端浏览器通过网络传输到服务器开发上进行。如果数据量很大，这对于网络和服务器开发的资源来说实在是一种无形的浪费。而使用JavaScript就可以在客户端进行数据验证。

（2）方便操纵html对象。JavaScript可以方便地操纵各种页面中的对象，用户可以使用JavaScript来控制页面中各个元素的外观、状态甚至运行方式，JavaScript可以根据用户的需要“定制”浏览器，从而使网页更加友好。

（3）支持分布式应用和运算。JavaScript可以使多种任务仅在用户端就可以完成，而不需要网络和服务器开发的参与，从而支持分布式应用 的运算和处理。

**缺点：**

（1）各浏览器厂商对JavaScript支持程度不同。目前在互联网 上有很多浏览器，如firefox、internet explorer、opera等，但每种浏览器支持JavaScript的程度是不一样的，不同的浏览器在浏览一个带有JavaScript脚本的主页时，由于对JavaScript的支持稍有不同，其效果会有一定的差距，有时甚至会显示不出来。

（2） “web安全性”对JavaScript一些功能牺牲。当把JavaScript的一个设计目标设定为“web安全性”时，就需要牺牲JavaScript的一些功能。因此，纯粹的JavaScript将不能打开、读写和保存用户计算机上的文件。其有权访问的唯一信息就是该JavaScript所嵌入开发 的那个web主页中的信息，简言之，JavaScript将只存在于它自己的小小世界—web主页里

### 1.8 Swift

Swift，苹果于2014年WWDC（苹果开发者大会）发布的新开发语言，可与Objective-C\*共同运行于Mac OS和iOS平台，用于搭建基于苹果平台的应用程序。Swift 结合了 C 和 Objective-C 的优点并且不受C兼容性的限制。Swift 采用安全的编程模式并添加了很多新特性，这将使编程更简单，更灵活，也更有趣。Swift 是基于成熟而且倍受喜爱的 Cocoa 和 Cocoa Touch 框架，他的降临将重新定义软件开发，同时它还是第一套具有与脚本语言同样的表现力和趣味性的系统编程语言。Swift的设计以安全为出发点，以避免各种常见的编程错误类别。

**优点：**

（1）简洁的语法

（2）更强的类型安全

（3）函数式编程的支持。Swift 语言本身提供了对函数式编程的支持。

（4）编写 OS X 下的自动化脚本

（5）结合playground，做到真正意义上的所见即所得

（6）便于扩展（能桥接ObjC）

（7）还有直接支持闭包。你想在哪儿定义函数就在哪儿定义。

**缺点：**

（1）App体积变大.使用 Swift 后，App体积大概增加 5-8M左右，对体积大小敏感的慎用。体积变大的原因是因为 Swift 还在变化，所以 Apple 没有在 iOS 系统里放入 Swift 的运行库，反而是每个 App 里都要包含其对应的 Swift 运行库

（2）Xcode 支持不够好。如果你是使用 Xcode经常卡住或者崩溃想必你是肯定碰到过了，这个是目前使用 Swift 最让人头疼的事情，即使是到现在XCode 9， 有时候也会遇到这种问题，所以要看你的承受力了……

（3）第三方库的支持不够多。目前确实 Swift 编写的第三方库确实不多，但可以通过桥接的方式来使用 Objc 的三方库，基本上没有太大问题。

（4）语言版本更新带来的编译问题。Swift 语言本身还在发展，所以每次版本更新后都会出现编译不过的情况（至少到目前为止还是），但是自从 4.0 版本发布后，改动没有 beta 时候那么大了，而且根据 Xcode 提示基本就可以解决语法变动导致的编译问题了。

### 1.9 Scala

Scala是一门多范式的编程语言，一种类似java的编程语言，设计初衷是实现可伸缩的语言、并集成面向对象编程和函数式编程的各种特性。它一方面吸收继承了多种语言中的优秀特性，一方面又没有抛弃 Java 这个强大的平台，它运行在 Java 虚拟机 (Java Virtual Machine) 之上，轻松实现和丰富的 Java 类库互联互通。它既支持面向对象的编程方式，又支持函数式编程。它写出的程序像动态语言一样简洁，但事实上它确是严格意义上的静态语言。Scala 就像一位武林中的集大成者，将过去几十年计算机语言发展历史中的精萃集于一身，化繁为简，为程序员们提供了一种新的选择。

**优点：**

（1）快速实验。Scala有交互式命令行(REPL)，可以在上面快速的试各种语法和代码。这对学习新特性，或者实验新想法非常有用。

（2）一致性。尽管Scala融合了静态类型系统、面向对象、函数式编程等语言特性，但却很少能看出融合的痕迹。

（3）类型安全。Scala编译器和类型系统非常强大，它的目标是尽量把软件错误消灭在编写过程中。 Scala类型系统是图灵完备的，甚至可以在编译期间解决问题。

（4）面向对象。Scala是面向对象的编程语言，所有的变量和方法都封装在对象中，可以把信息封装起来供外部使用。

（5）函数式编程。Scala同时又是函数式编程语言，函数可以独立存在，可以定义一个函数作为另一个函数的返回值，也可以接受函数作为函数的参数。这给组合函数带来了很大的便利。如何把面向对象编程形容成搭积木的话，函数式编程就像拼线条，更灵活和更有创意。

（6）异步编程。由于函数式编程提倡变量不可变，使异步编程变得非常容易。同时Scala提供的Future， 和akka类库，使得异步编程变得非常容易。

（7）基于JVM。Scala会被编译成为jvm bytecode，所以Scala能无缝集成已有的Java类库。你可以非常自然的使用已经存在的非常庞大且稳定的Java类库，比如小巧好用的apache.common.\*,或者Java上的各种工具类库。

**缺点：**

（1）刚刚起步的IDE 支持。正如 Lift 的作者所言，Scala 的 IDE 支持，虽然进行了很多开发，还没有做到 Java 那种地步。重构支持、代码完整以及单元测试整合都很差。更不必说大多数框架支持工具不能很好地与 Scala 兼容。

（2）大多数 IDE 不支持联合编译。同样，当 Scala 更加普及之后，这一点会有所改变。

（3）类的不变性并非真的不变性，因为引用对象自身可能不是不变的。并且目前没有方法可以确保整个对象图谱是不变的。

（4）让 JSR 223 完美地兼容 Scala 非常困难。但另一方面，取得足够好的兼容还是相当容易的。

（5）Scala 不支持元编程（metaprogramming）。通过将其与动态语言结合，如 Ruby，可以绕过这个问题，但如果你是元编程的重度使用者，使用一个完全不同的语言是一种较好的解决办法（Fan 是另一个运行在 JVM 上的静态类型语言，与 Scala 类似，但支持元编程）。

（6）使用 Java 资源的框架，如客户端 GWT，不能很好地兼容 Scala（虽然有人已经在服务器端让 Scala 与 GWT 兼容）。不过，有一个项目正在进行，将能够使 Scala 转化为 Java 资源。

（7）语法和某些概念与 Java 有点不同，比如：颠倒的类型声明顺序、使用下划线而不是通配符、星号和缺省值，太多种类的空概念（nothing）、没有静态方法（你需要使用单例对象 singleton object 作为替代）。文档对这些问题有很好的解释，但是，请留意，这不是 Java 代码到 Scala 代码的自动转换。

### 1.10 GO

Go是一种新的语言，一种并发的、带垃圾回收的、快速编译的语言，于2009年11月正式宣布推出，成为开放源代码项目，并在Linux及Mac OS X平台上进行了实现，后追加Windows系统下的实现。

**优点：**

（1）开发速度很快,代码比较简洁.

（2）标准库比较强大

（3）编译简单,最后就一个可执行文件

**缺点：**

（1）GO运行速度很慢。

（2）GO缺少安全检查,

（3）没有泛型支持

（4）没有一个好的IDE可以点进去看函数定义.

（5）没有一个好的工程版本管理,尤其是github上第三方的.

### 1.11 Rust

Rust是Mozilla开发的注重安全、性能和并发性的编程语言。“Rust”，由web语言的领军人物Brendan Eich（js之父），Dave Herman以及Mozilla公司的Graydon Hoare 合力开发。Rust是针对多核体系提出的语言，并且吸收一些其他动态语言的重要特性，比如不需要管理内存，比如不会出现Null指针等等

**优点：**

（1）rust是理论语言，而且是唯一的世界级的，世界级PL理论家的语言中唯一一个存在的。

（2）Rust把安全、精确的内存管理作为一切的中心放在首要的位置。

（3）同时拥有特别强的控制性和特别强的安全性。

（4）有更强的语义，更容易捕获错误的逻辑，编译器直接检查出你代码中的不安全的部分

（5）效率高，速度特别的快

（6）支持范型

（7）社区活跃度很高，更加的强调了社区的作用。

**缺点：**

（1）库不完善，库不全用户少，比较小众

（2）rust编译复杂

### 1.12 HTML5

万维网的核心语言、标准通用标记语言下的一个应用超文本标记语言（HTML）的第五次重大修改

**优点：**

（1）提高可用性和改进用户的友好体验；

（2）有几个新的标签，这将有助于开发人员定义重要的内容；

（3）可以给站点带来更多的多媒体元素(视频和音频)；

（4）可以很好的替代FLASH和Silverlight；

（5）当涉及到网站的抓取和索引的时候，对于SEO很友好；

（6）将被大量应用于移动应用程序和游戏；

（7）可移植性好。

**缺点：**

（1）该标准并未能很好的被浏览器所支持。

（2）因新标签的引入，各浏览器之间将缺少一种统一的数据描述格式，造成用户体验不佳。

## 2 主流的知识库开发工具

一般是指一些被软件工程师用于为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件的特殊软件。这些开发工具都非常具有技术共性，另外还都具有一个共同点，那就是他们既提供开发工具平台，又提供基于该平台生成的各种应用系统，两种产品相辅相成，这种模式也是开发平台销售的主流模式。

### 2.1 Java工具编辑

**2.1.1 Eclipse**

Eclipse是一个开放源代码的、基于Java的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括Java开发工具（Java Development Kit，JDK）。

**2.1.2 MyEclipse**

MyEclipse企业级工作平台（MyEclipse Enterprise Workbench ，简称MyEclipse）是对EclipseIDE的扩展，利用它我们可以在数据库和JavaEE的开发、发布以及应用程序服务器的整合方面极大的提高工作效率。它是功能丰富的JavaEE集成开发环境，包括了完备的编码、调试、测试和发布功能，完整支持HTML, Struts, JSP, CSS, JavaScript, Spring, SQL, Hibernate，可以说MyEclipse是几乎囊括了目前所有主流开源产品的专属eclipse开发工具。

**2.1.3 JBuilder**

JBuilder是Borland公司开发的针对java的开发工具，使用JBuilder将可以快速，有效的开发各类java应用，它使用的JDK与sun公司标准的JDK不同，它经过了较多的修改，以便开发人员能够像开发Delphi应用那样开发java应用。 JBuilder的核心有一部分采用了VCL技术，使得程序的条理非常清晰，就算是初学者，也能完整的看完整个代码。JBuilder另一个特点是简化了团队合作，它采用的互联网工作室技术使不同地区，甚至不同国家的人联合开发一个项目成为了可能。

**2.1.4 NetBeans**

NetBeans 由Sun公司（2009年被甲骨文收购）在2000年创立，它是开放源运动以及开发人员和客户社区的家园，旨在构建世界级的Java IDE。NetBeans当前可以在Solaris、Windows、Linux和Macintosh OS X平台上进行开发，并在SPL(Sun公用许可)范围内使用。

NetBeans是开源软件开发集成环境，是一个开放框架，可扩展的开发平台，可以用于Java，C/C++，PHP等语言的开发，本身是一个开发平台，可以通过扩展插件来扩展功能。

**2.1.5 Jcreator**

Jcreator是Xinox Software公司开发的一个用于Java程序设计的集成开发环境（IDE），具有编辑、调试、运行Java程序的功能。当前最新版本是Jcreator5.00，它又分为LE和Pro版本。LE版本功能上受到一些限制，是免费版本。Pro版本功能最全，但这个版本是一个共享软件。这个软件比较小巧，对硬件要求不是很高，完全用C++写的，速度快、效率高。具有语法着色、代码自动完成、代码参数提示、工程向导、类向导等功能。第一次启动时提示设置JavaJDK主目录及JDKJavaDoc目录，软件自动设置好类路径、编译器及解释器路径，还可以在帮助菜单中使用JDKHelp。

### 2.2 Python开发工具

**2.2.1 github**

Git是一个分布式的版本控制系统，最初由Linus Torvalds编写，用作Linux内核代码的管理。在推出后，Git在其它项目中也取得了很大成功，尤其是在Ruby社区中。目前，包括Rubinius、Merb和Bitcoin在内的很多知名项目都使用了Git。Git同样可以被诸如Capistrano和Vlad the Deployer这样的部署工具所使用。

**2.2.2 Vim**

Vim 是高级文本编辑器，旨在提供实际的 Unix 编辑器‘Vi’功能，支持更多更完善的特性集。Vim 不需要花费太多的学习时间，一旦你需要一个无缝的编程体验，那么就会把 Vim 集成到你的工作流中。

**2.2.3 Eclipse with PyDev**

Eclipse 是非常流行的 IDE，而且已经有了很久的历史。Eclipse with Pydev 允许开发者创建有用和交互式的 Web 应用。PyDev 是 Eclipse 开发 Python 的 IDE，支持 Python，Jython和 IronPython 的开发。

**2.2.4 Sublime Text**

Sublime Text 是开发者中最流行的编辑器之一，多功能，支持多种语言，而且在开发者社区非常受欢迎。Sublime 有自己的包管理器，开发者可以使用TA来安装组件，插件和额外的样式，所有这些都能提升你的编码体验。

**2.2.5 Emacs**

Emacs是一种强大的文本编辑器，在程序员和其他以技术工作为主的计算机用户中广受欢迎。EMACS，即Editor MACroS（宏编辑器）的缩写，最初由Richard Stallman于1975年在MIT协同Guy Steele共同完成。

**2.2.6 PyCharm**

PyCharm是由JetBrains打造的一款Python IDE。我们知道，VS2010的重构插件Resharper就是出自JetBrains之手。那么，PyCharm吸引人的特点主要在两方面。首先，PyCharm用于一般IDE具备的功能，比如， 调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制。另外，PyCharm还提供了一些很好的功能用于Django开发，同时支持Google App Engine，更酷的是，PyCharm支持IronPython！

**2.2.7 PyScripter**

PyScripter 是一个开源的 Python 语言集成开发环境。兼容 Python 2.6 和 3.0 版本。PyScripter只能使用 32位的 Python版本。

### 2.3 常用工具

**2.3.1 Visual Studio**

Visual Studio是微软公司推出的开发环境，是最流行的Windows平台应用程序开发环境。它是一套完整的开发工具，用于生成 ASP NET Web 应用程序、XML Web services、桌面应用程序和移动应用程序。Visual Basic、Visual C# 和 Visual C++ 都使用相同的集成开发环境 (IDE)，这样就能够进行工具共享，并能够轻松地创建混合语言解决方案。

**2.3.2 FLASH**

FLASH是Macromedia公司出品(现已被Adobe收购)的，用在互联网上动态的、可互动的shockwave。它的优点是体积小，可边下载边播放，这样就避免了用户长时间的等待。FLASH可以用其生成动画，还可在网页中加入声音。这样你就能生成多媒体的图形和界面，而使文件的体积却很小。FLASH虽然不可以象一门语言一样进行编程，但用其内置的语句并结合JAVASCRIPE，您也可作出互动性很强的主页来。FLASH另外一个特点就是必须安装插件PLUG-IN，才能被浏览器所接受！当然这也避免了浏览器之间的差异，使之一视同仁！有人曾经说过：下个世纪的网络设计人不会用FLASH，必将被淘汰出局！我相信这句话没错！（因为我们学它，所以我多讲一些了）

**2.3.3 FrontPage**

应用开发工具可大大提高编制网络课件的效率. 目前,国际上比较流行的Web页制作软件大致可以分为两类:代码型和所见即所得型.代码型的制作软件要编程,对于初学者来说想要立即上手比较困难. Microsoft公司的FrontPage就不同了,它作为一种所见即所得型的代表软件,可以使使用者的工作效率得以很大提高. FrontPage 2000 是Office 2000 家族的一员,沿袭着Office 的风格.所以会用 Word 的人很容易学会FrontPage。Office 2003之后FrontPage已经并入微软Expression Studio系列产品。

**2.3.4 DW**

Dreamweaver是美国MACROMEDIA公司开发的集网页制作和管理网站于一身的所见即所得网页编辑器，它是第一套针对专业网页设计师特别发展的视觉化网页开发工具，利用它可以轻而易举地制作出跨越平台限制和跨越浏览器限制的充满动感的网页。Micromedia公司后被Adobe公司收购，Dreamweaver也随Adobe软件一同发布。