

编译原理

王 挺

国防科学技术大学

计算机学院 计算机科学与技术系

Email: tingwang@nudt.edu.cn

课程网站 : http://www.icourses.cn/coursestatic/course_6558.html



编译原理

第一章 引论

学习目的

■ 在大学学什么？

□ 重要目标之一：学思维方法

- **Critical thinking**: is the process of thinking that questions assumptions. It is a way of deciding whether a claim is true, false; sometimes true, or partly true.[Wikipedia]

- **批判性思维**：逻辑清晰严密的思考

■ 从计算机科学中学什么？

- 理解计算系统

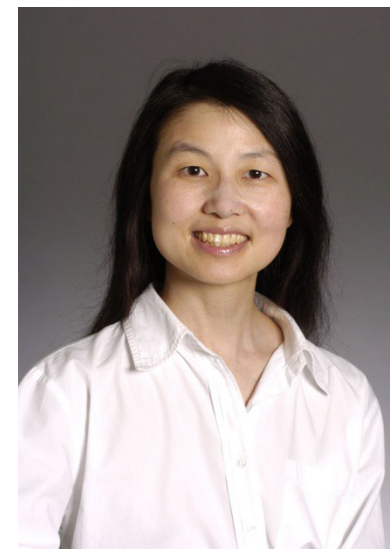
- 设计计算系统

- 训练计算思维 (Computational Thinking)

计算思维

■ 计算思维基本概念

- Jeannette M. Wing, Computational Thinking, Communications of ACM, Vol.49, No.3, 2006, pp.33-35.
- 被认为是近十年来产生的最具有基础性、长期性的学术思想，将成为 21 世纪计算机科学研究热点



Viewpoint | Joannetta M. Wing

Computational Thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.



Computational thinking builds on the power and limits of computing processes, whether they are attained by a human or by a machine. Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of solving alone. Computational thinking confronts the dilemmas of machine intelligence: What can humans do better than computers? And what can computers do better than humans? Most fundamentally it addresses the question: What is computable? Today, we have only part of the

Computational thinking is a fundamental skill in every discipline. It is a way of thinking that involves problem-solving, writing, and analyzing. We should add computational thinking to every child's analytical toolkit. Just as the printing press laid the ground for the spread of ideas, this is a property of information technology that is the computing and computer-based spread of computational thinking.

Computational thinking is a way of solving problems, developing algorithms, and automating them. It is a way of thinking that is based on the concepts of computation. Computational thinking is a way of thinking that is based on the concepts of computation.

Having to solve a particular problem, we might ask: What is the problem? What are the constraints? What are the goals? What are the resources? What are the methods? What are the results? What are the conclusions?

sible. Tackling the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We can consider the machine's instructions as, in essence, common sense, and its operating mechanism.

In solving a problem efficiently, we might seek out whether an approach may be solution in good enough, whether we can use randomization to our advantage, and whether false positions or false expectations are allowed. Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by induction, recursion, combinatorics, or probability.

Consequently, thinking is thinking *in* reality. It is parallel processing. It is far from being made outside and then sent in. It is type checking on the general lines of dimensional analysis. It is recognizing both the virtues and the dangers of analogy, or giving someone an example; more than one reason. It is recognizing both the use and abuse of hidden underlying and procedure call. It is judging a program not just for correctness and efficiency but for wisdom, and a systems design for simplicity and elegance.

Computational thinking is using abstractions and decomposition when analysing a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or model so that relevant aspects of a problem are made tractable. It is using formalisms to describe a system's behaviour in clearly and declaratively. It is having the confidence to use safety, not, reliability, and influence a large complex system without a mathematical background. It is

计算思维

■ 计算思维是什么

- **计算思维**是运用计算机科学的基础概念去求解问题、设计系统和理解人类的行为，它包括了一系列广泛的计算机科学的思维方法
- 计算思维和阅读、写作和算术一样，是 21 世纪每个人的基本技能，而不仅仅属于计算机科学家
- 计算思维在生物、物理、化学、经济学、统计学等其他学科中的影响已经显现

计算思维

- 包括一系列广泛的计算机科学的思维方法
 - 抽象
 - 自动化
 - 问题分解
 - 递归
 - 权衡
 - 保护、冗余、容错、纠错和恢复
 - 利用启发式推理来寻求解答
 - 在不确定情况下的规划、学习和调度
 - ...

关于编译理论与技术

- 课程内容

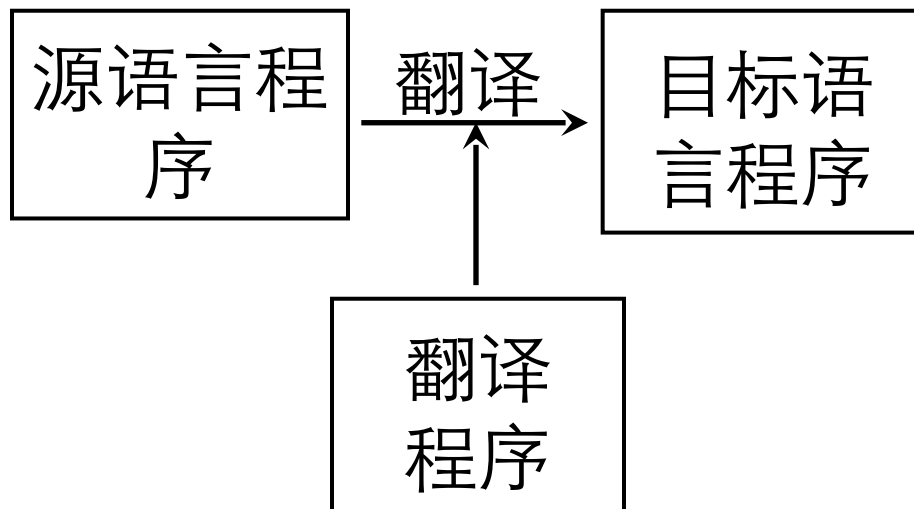
- 介绍程序设计语言 **编译程序** 构造的 **基本原理** 和 **基本实现技术**

- 什么是编译程序？

什么是编译程序

■ 翻译程序

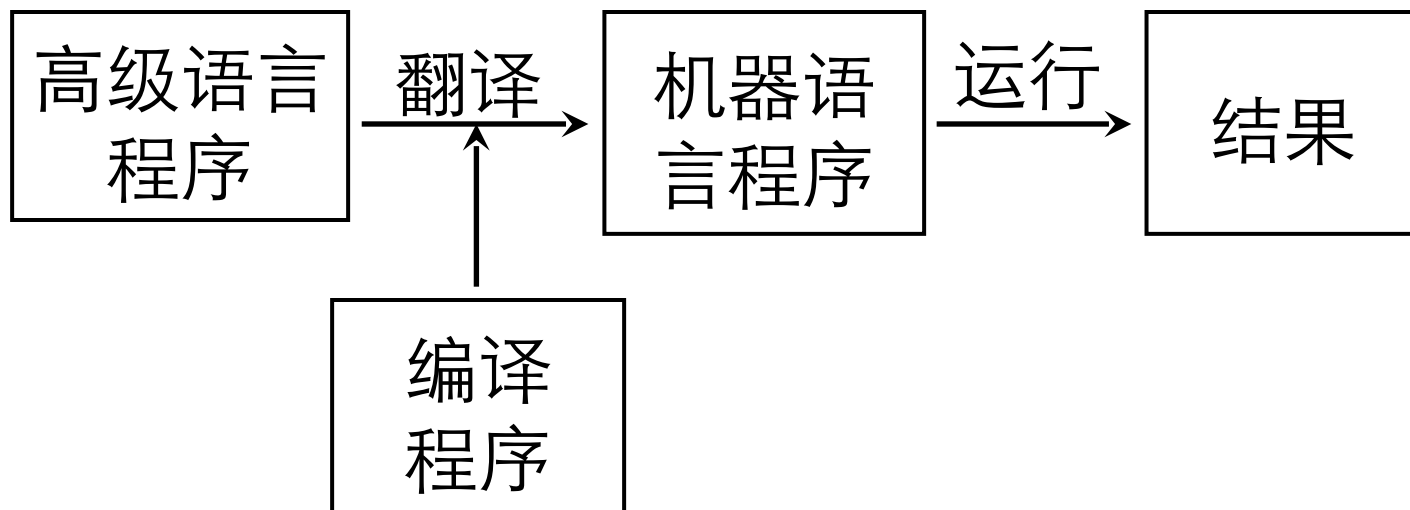
- 把某一种语言程序（称为源语言程序）等价地转换成另一种语言程序（称为目标语言程序）的程序



什么是编译程序

■ 编译程序 (compiler)

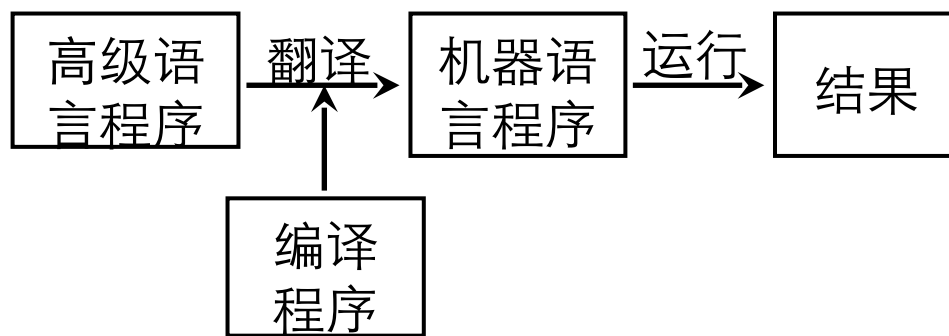
- 把某一种高级语言程序等价地转换成另一种低级语言程序（如汇编语言或机器语言程序）的程序



什么是编译程序

■ 编译程序 (compiler)

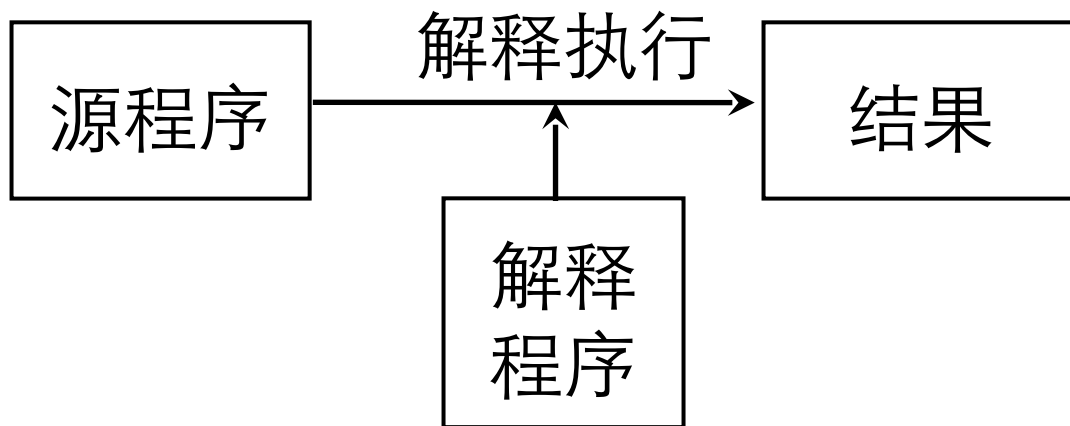
- 把某一种高级语言程序等价地转换成另一种低级语言程序（如汇编语言或机器语言程序）的程序
- 诊断编译程序
- 优化编译程序
- 交叉编译程序
- 可变目标编译程序



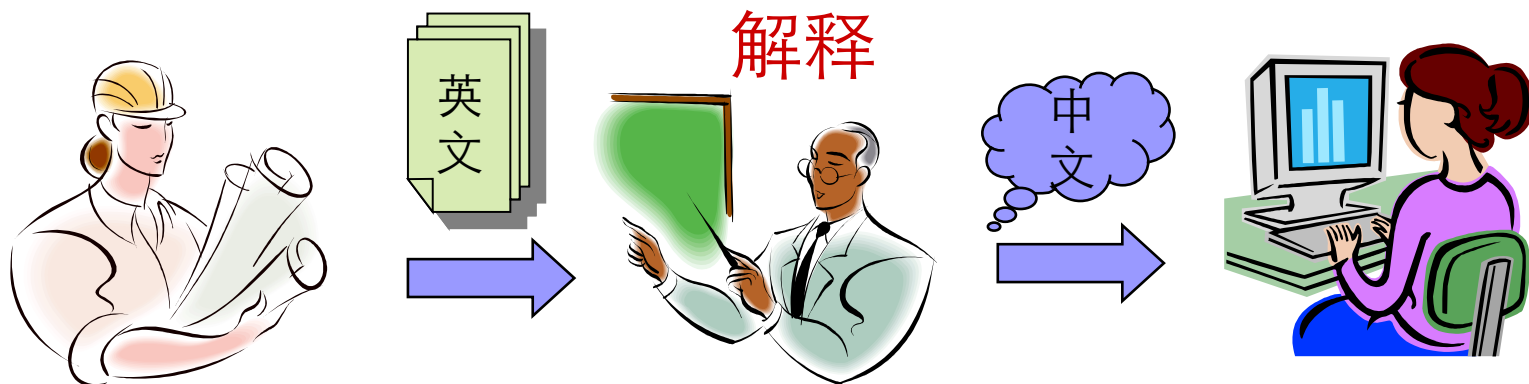
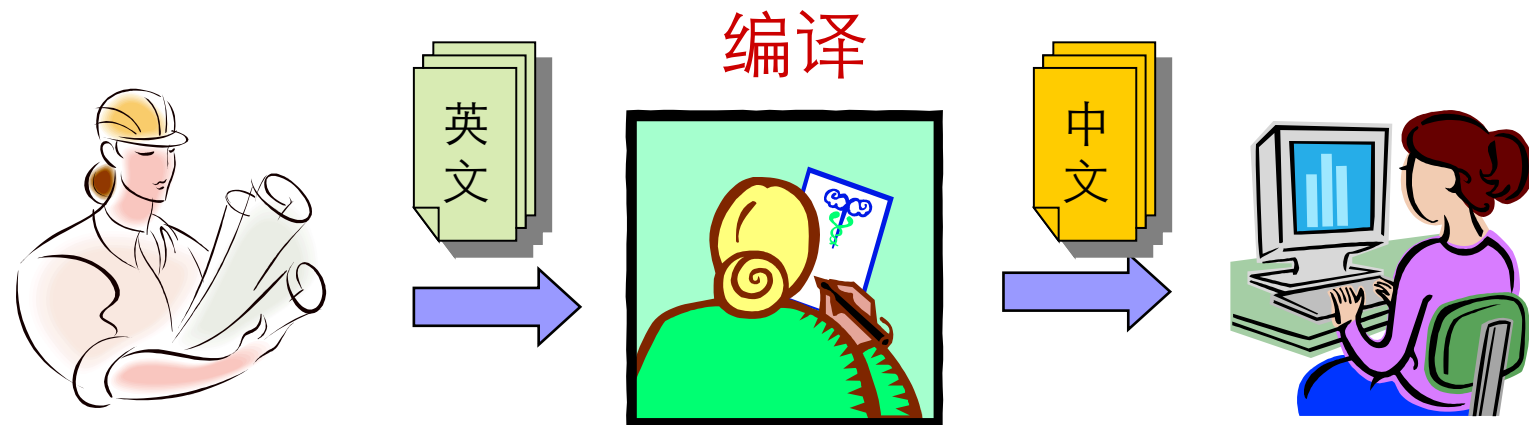
什么是编译程序

■ 解释程序

- 把源语言写的源程序作为输入，但不产生目标程序，而是边解释边执行源程序本身



编译程序 vs. 解释程序



关于编译理论与技术

■ 编译理论与技术

- 计算机科学与技术中理论和实践相结合的最好典范

■ ACM 图灵奖

- 授予在计算机技术领域作出突出贡献的科学家
- 程序设计语言、编译理论与方法约占 1/3

ACM 图灵奖



Alan J. Perlis



Edsger W. Dijkstra



John W. Backus



Kenneth E. Iverson



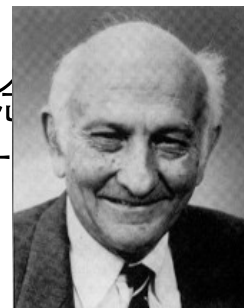
Niklaus Wirth

■ 程序设计语言

- Alan J. Perlis (1966) -- ALGOL
- Edsger Wybe Dijkstra (1972) -- ALGOL
- Michael O. Rabin & Dana S. Scott (1976) -- 非确定性自动机
- John W. Backus (1977) -- FORTRAN
- Kenneth Eugene Iverson (1979) -- APL 程序语言
- Niklaus Wirth (1984) -- PASCAL
- John Cocke (1987) -- RISC & 编译优化
- O. Dahl , K.Nygaard (2001) -- Simula 语言和 OO 概念
- Alan Kay(2003) -- SmallTalk 语言和面向对象程序设计
- Peter Naur(2005) -- ALGOL60 以及编译设计
- Frances E. Allen(2006)-- 优化编译器
- Barbara Liskov(2008)-- 编程语言和系统设计的实践与理论



O. Dahl



John Cocke



K.Nygaard



Michael O. Rabin



Dana S. Scott



Donald E. Knuth



Barbara Liskov



Frances E. Allen



Peter Naur



Alan Kay

关于编译理论与技术

■ 编译理论与技术

- 计算机科学与技术中理论和实践相结合的最好典范

■ ACM 图灵奖

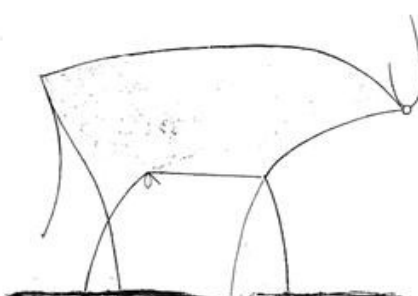
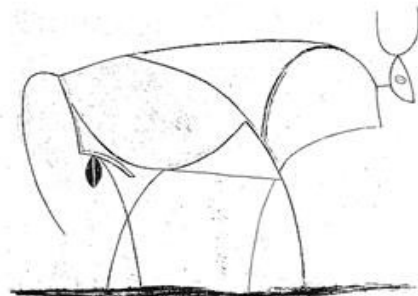
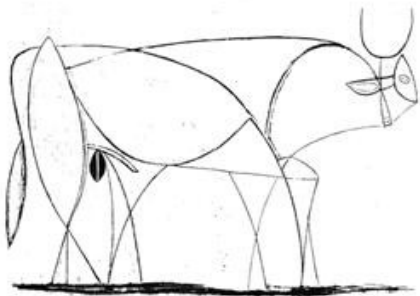
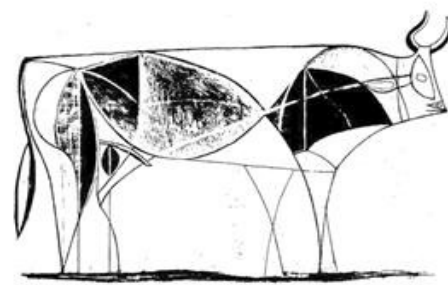
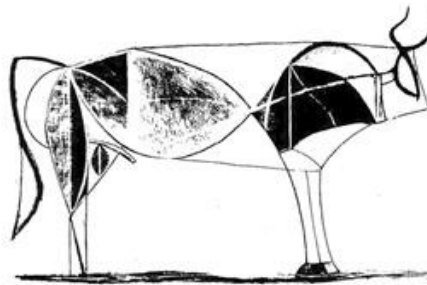
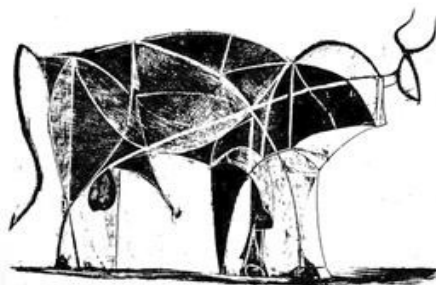
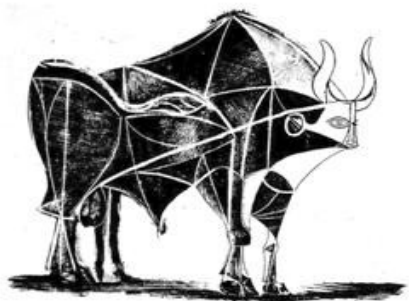
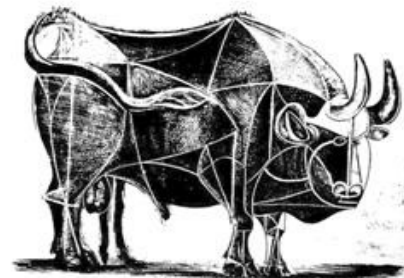
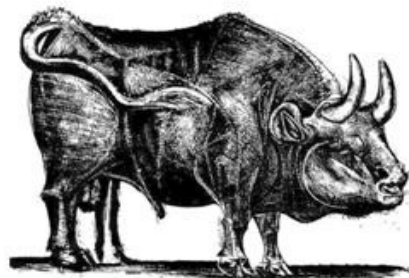
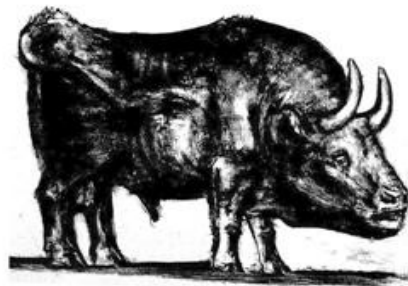
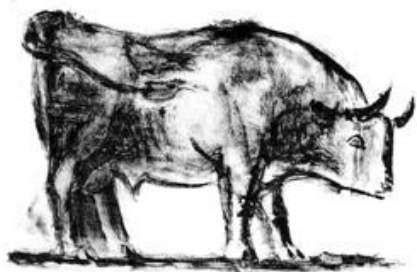
- 授予在计算机技术领域作出突出贡献的科学家
- 程序设计语言、编译理论与方法约占 1/3

■ 体现了很多典型的计算思维方法

计算思维 vs. 编译

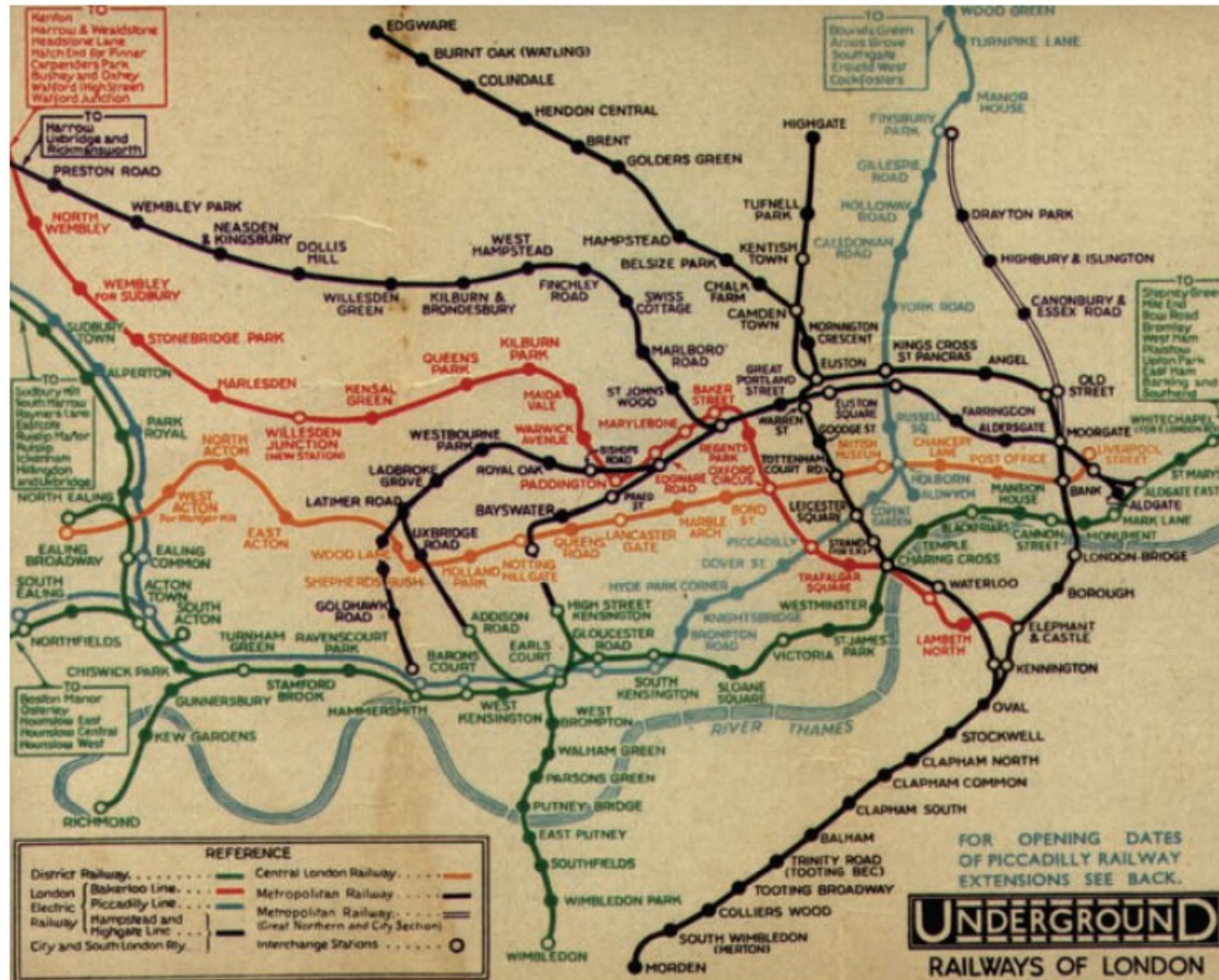
■ 抽象 (Abstraction)

- **抽象**就是忽略一个主题中与当前问题 (或目标) 无关的那些方面, 以便更充分地注意与当前问题 (或目标) 有关的方面
- **抽象**是从众多的事物中抽取出共同的、本质性的特征, 舍弃其非本质的特征
- **抽象**是一种从个体把握一般、从现象把握本质的认知过程和思维方法

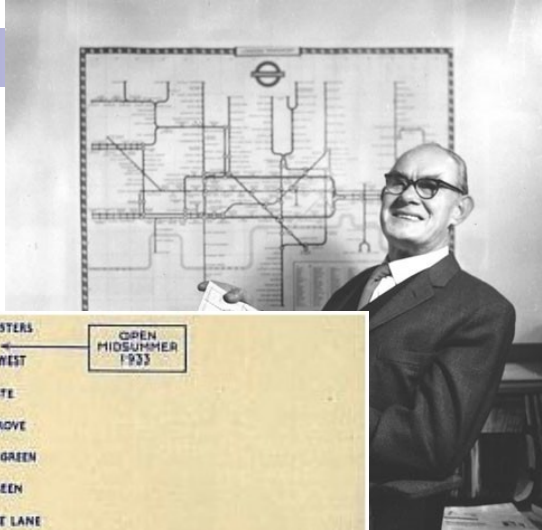


毕加索 《牛》

The London Underground Map (1928)

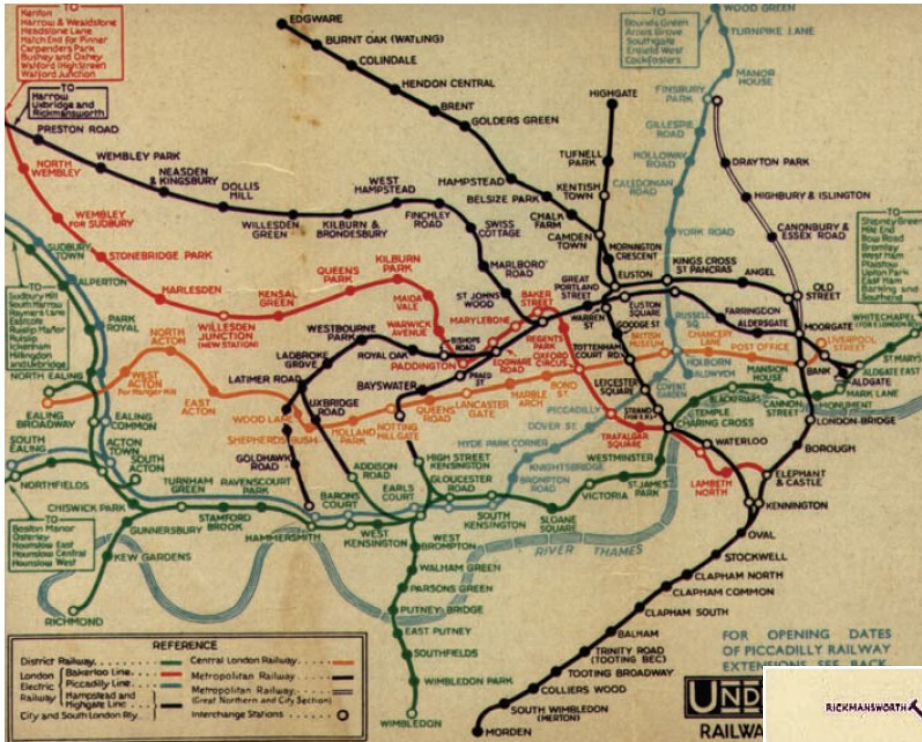


The London Underground Map (1933)



ck





The London Underground Map (1928)

The London Underground Map (1933) by Harry Beck



北京地铁线路图

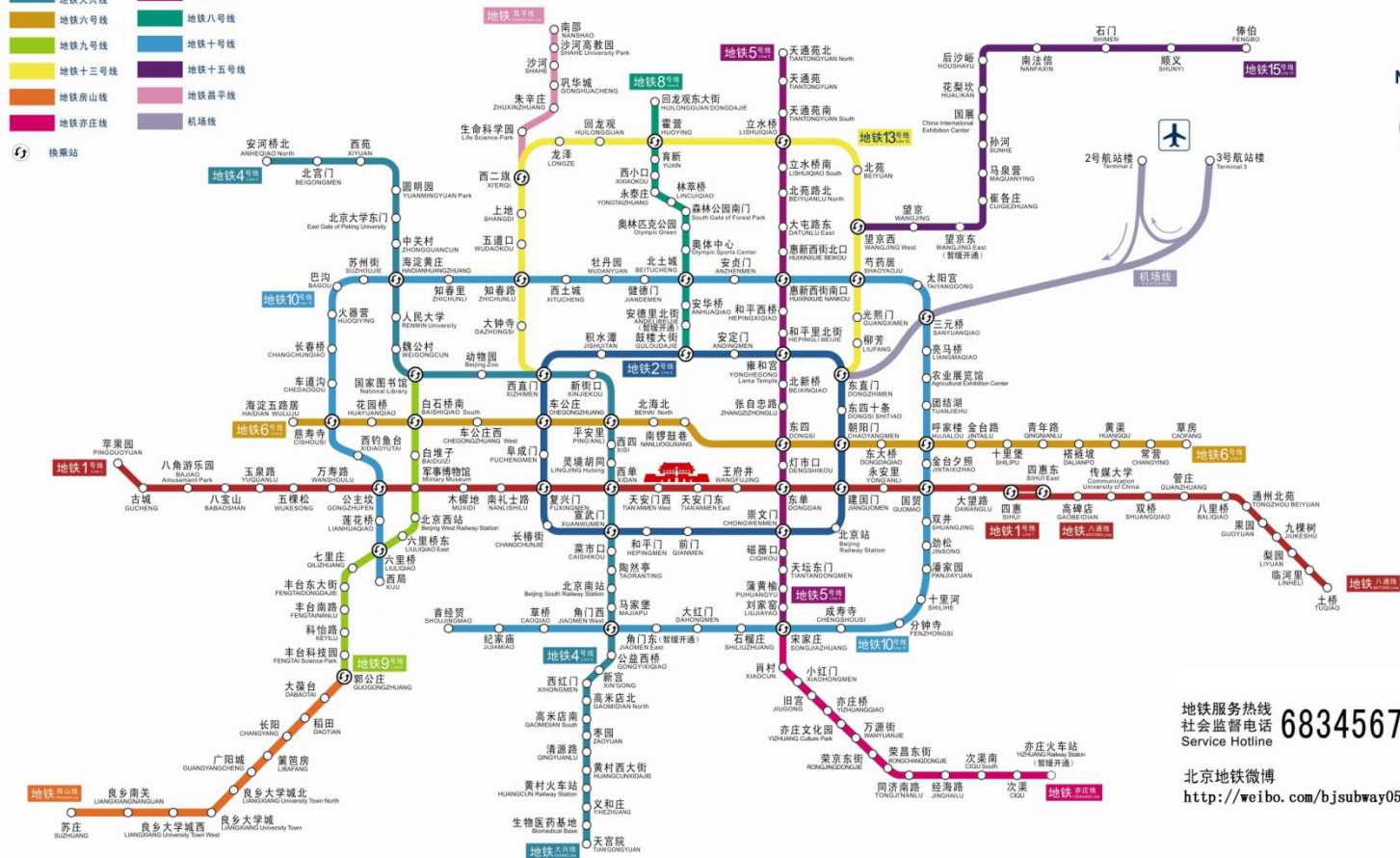
北京地铁线路图 Beijing Subway Map

图例:

Legend

- 地铁一号线
- 地铁八号线
- 地铁四号线
- 地铁六号线
- 地铁九号线
- 地铁十三号线
- 地铁房山线
- 地铁亦庄线
- 地铁二号线
- 地铁五号线
- 地铁八号线
- 地铁十号线
- 地铁十五号线
- 地铁昌平线
- 机场线

换乘站



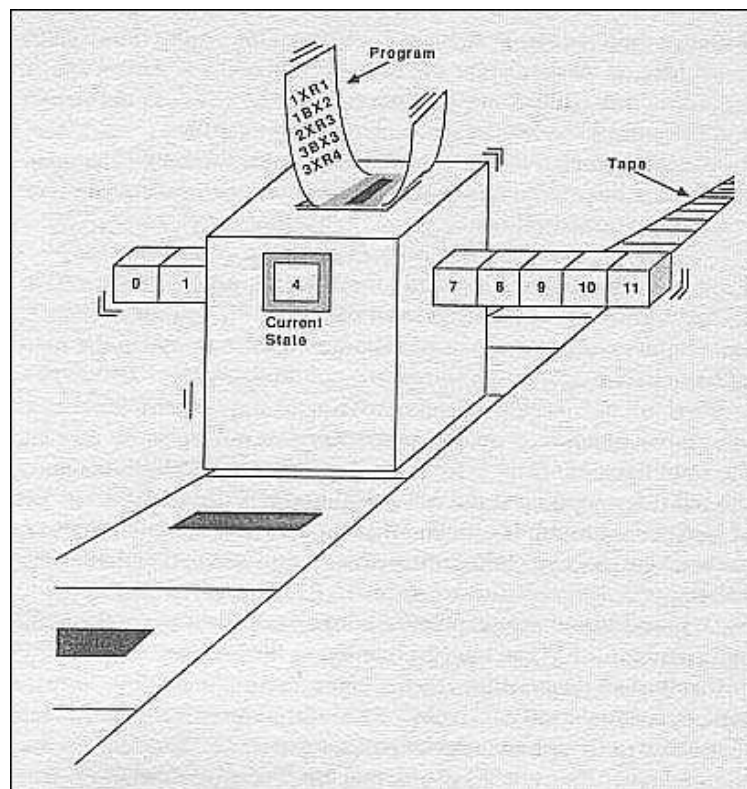
地铁服务热线
Service Hotline
68345678

北京地铁微博
http://weibo.com/bjsubway0528

计算思维 vs. 编译

■ 抽象 (Abstraction)

□ 图灵机



计算思维 vs. 编译

■ 抽象 (Abstraction)

- 图灵机

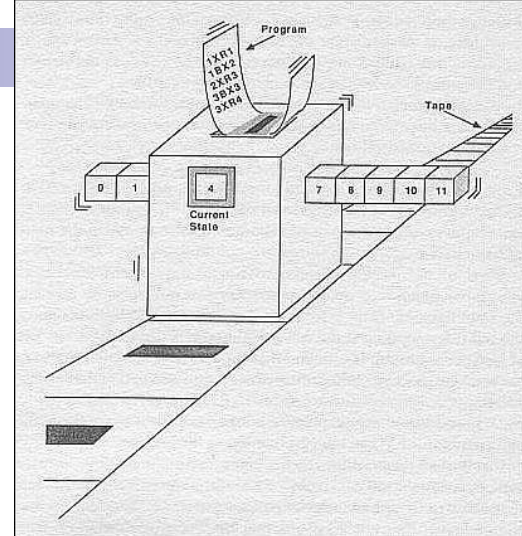
- 组成

- 一条无限长的纸带
- 一个读写头
- 一套控制读写头工作的规则
- 一个状态寄存器

- 邱奇 - 图灵论题 (The Church-Turing thesis)

- 所有计算或算法都可以由一台图灵机来执行

- 可计算 = 图灵可计算



计算思维 vs. 编译

■ 编译原理中的 "抽象"

- 有限自动机
- 形式文法
- ...

计算思维 vs. 编译

■ 自动化

- 将抽象思维的结果在计算机上进行实现，是一个将计算思维成果物化的过程，也是将理论成果应用于技术的实践
- 自动化的思维方法不仅体现在编译程序本身的工作机制上，更体现在了编译程序的生成工具的研究和设计上

计算思维 vs. 编译

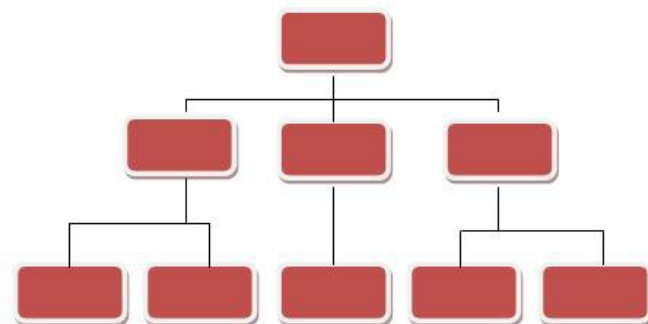
■ 编译原理中的 " 自动化 "

- 有限自动机
- 预测分析程序
- 算符优先分析
- LR 分析
- ...

计算思维 vs. 编译

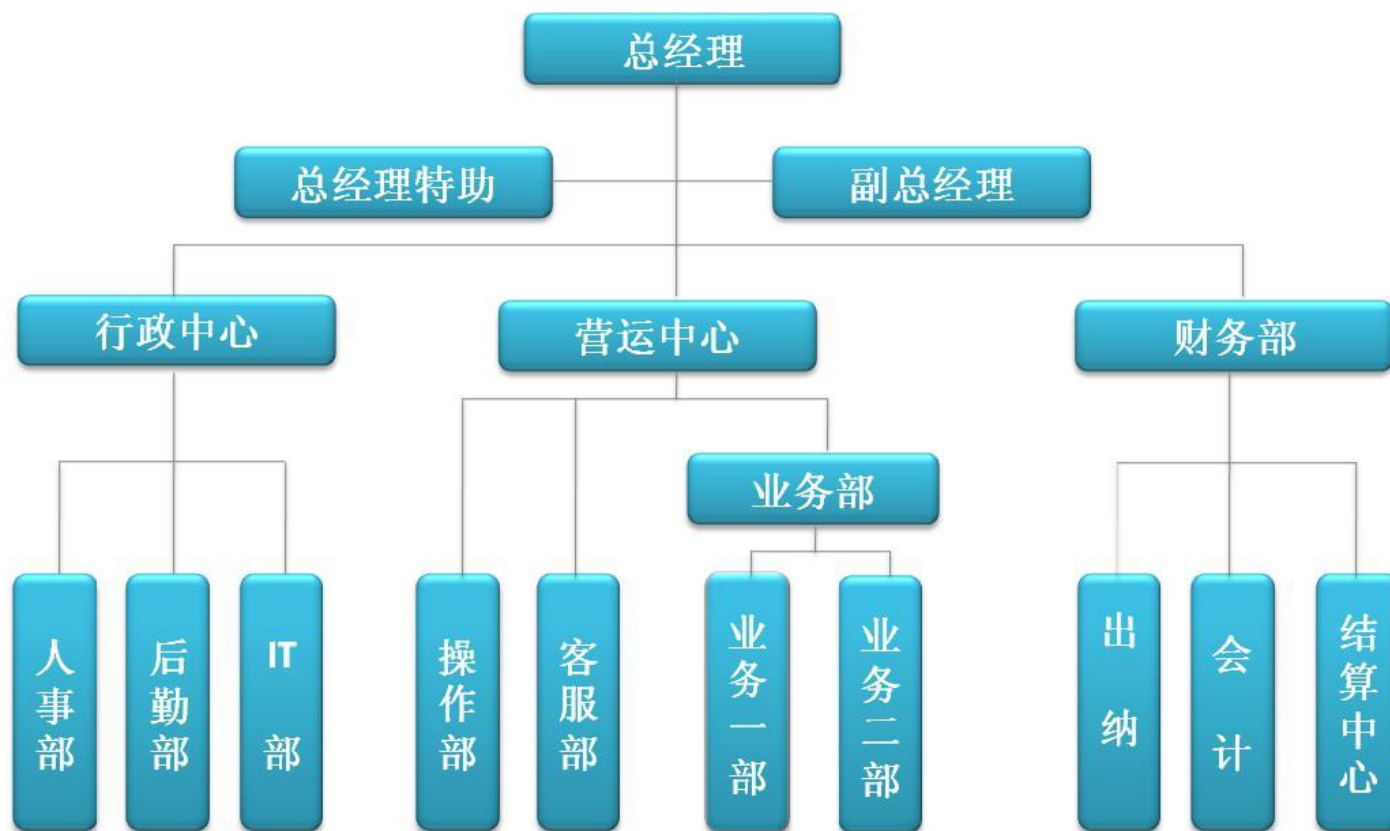
■ 分解 (Decomposition)

- 将大规模的复杂问题分解成若干个较小规模的、更简单的问题加以解决
 - 明确描述问题，并对问题解法作出全局性决策
 - 把问题分解成相对独立的子问题
 - 再以同样的方式对每个子问题进一步精确化，直到获得对问题的明确的解答



计算思维 vs. 编译

■ 分解 (Decomposition)



计算思维 vs. 编译

- 编译原理中的 " 问题分解 "
- ☐ 为什么编译程序引入中间语言?
- ☐ 为什么编译分成多个阶段?
- ☐ ...

计算思维 vs. 编译

■ 递归

- 问题的解决又依赖于类似问题的解决，只不过后者的复杂程度或规模较原来的问题更小
- 一旦将问题的复杂程度和规模化简到足够小时，问题的解法其实非常简单

计算思维 vs. 编译

- 编译原理中的 " 递归 "
 - 递归下降分析
 - 基于树遍历的属性计算
 - 语法制导翻译
 - ...

计算思维 vs. 编译

■ 权衡（折中）

- 理论可实现 vs. 实际可实现
- 理论研究重在探寻问题求解的方法，对于理论成果的研究运用又需要在能力和运用中作出权衡

计算思维 vs. 编译

■ 编译原理中的 " 权衡 "

- 用上下文无关文法来描述和处理高级程序设计语言
- 优化措施的选择
- ...

计算思维 vs. 编译

- 计算思维包括一系列广泛的计算机科学的思维方法
 - 抽象
 - 自动化
 - 问题分解
 - 递归
 - 权衡
 - ...
- 请大家在学习过程告诉我还有哪些方法和例子
- 学习编译原理
- 训练计算思维
- 享受计算之美

学习编译原理的意义

- 学习编译程序构造原理，技术
 - 其中的计算思维方法
- 更好地理解高级语言
- 编译的原理和方法有助于构造一些实用的工具

编译原理和方法的应用

■ Html/XML 分析

HTML 语言教程：目录

[新的特色](#) *** [HTML 标记\(Tag\)的索引\(Index\)](#)

[页面\(Page\)](#) [字体\(Font\)](#) [文字布局\(Text Style\)](#) [图象\(Image\)](#) [表单\(Form\)](#)

[表格\(Table\)](#) [表格进阶\(Advanced\)](#) [多窗口页面\(Frames\)](#)

[会移动的文字\(Marquee\)](#) [多媒体页面\(Alternative Inline Elements\)](#)

详细目录

- [页面](#)
 - [文件结构](#)
 - [语言字符集信息](#)
 - [背景色彩和文字色彩](#)
 - [页面空白](#)
 - [链接](#)
 - [开新窗口](#)
 - [标尺线](#)
- [字体](#)
 - [标题](#)
 - [字号](#)
 - [物理字体](#)
 - [逻辑字体](#)

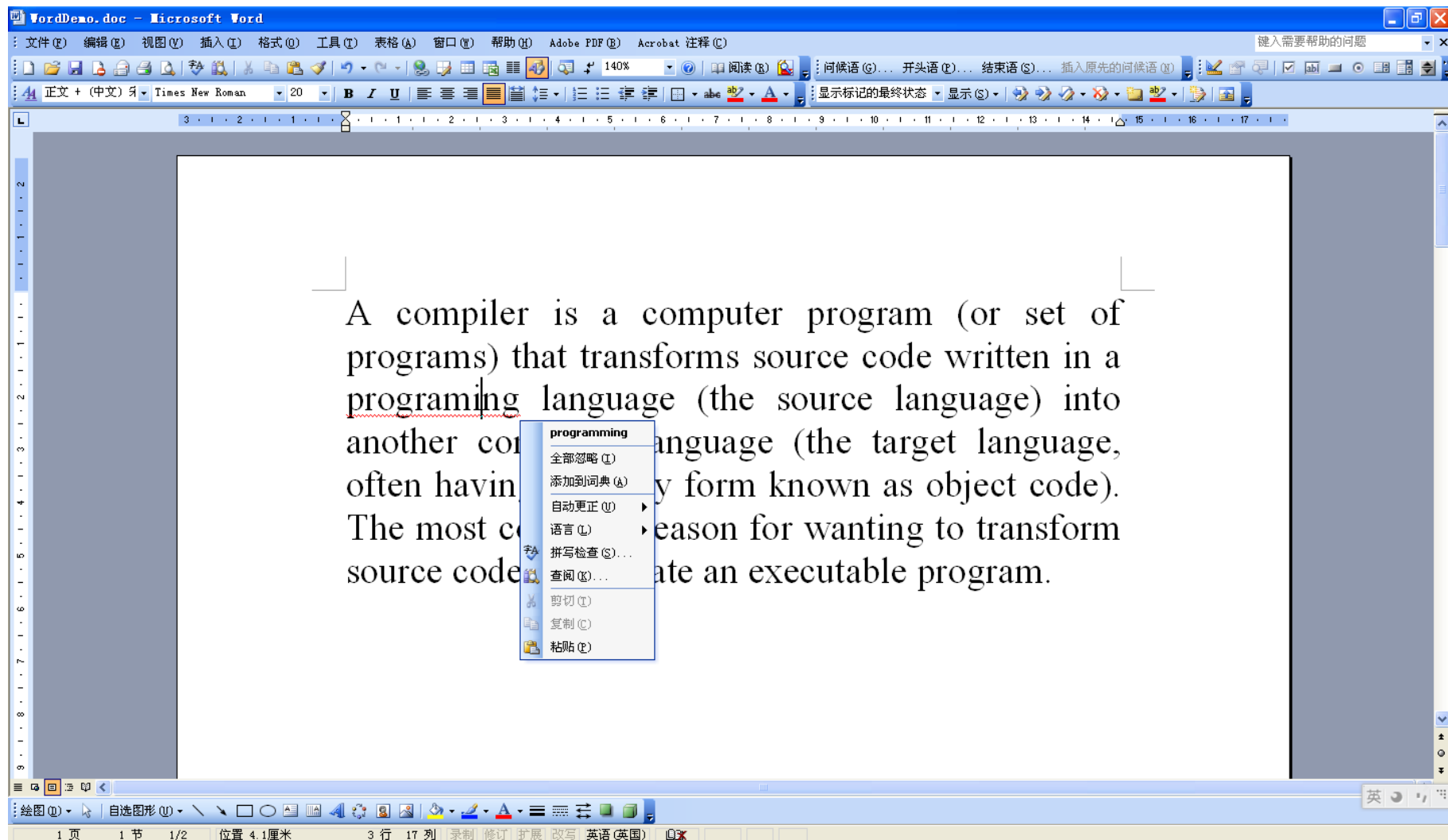
编译原理和方法的应用

■ Html/XML 分析

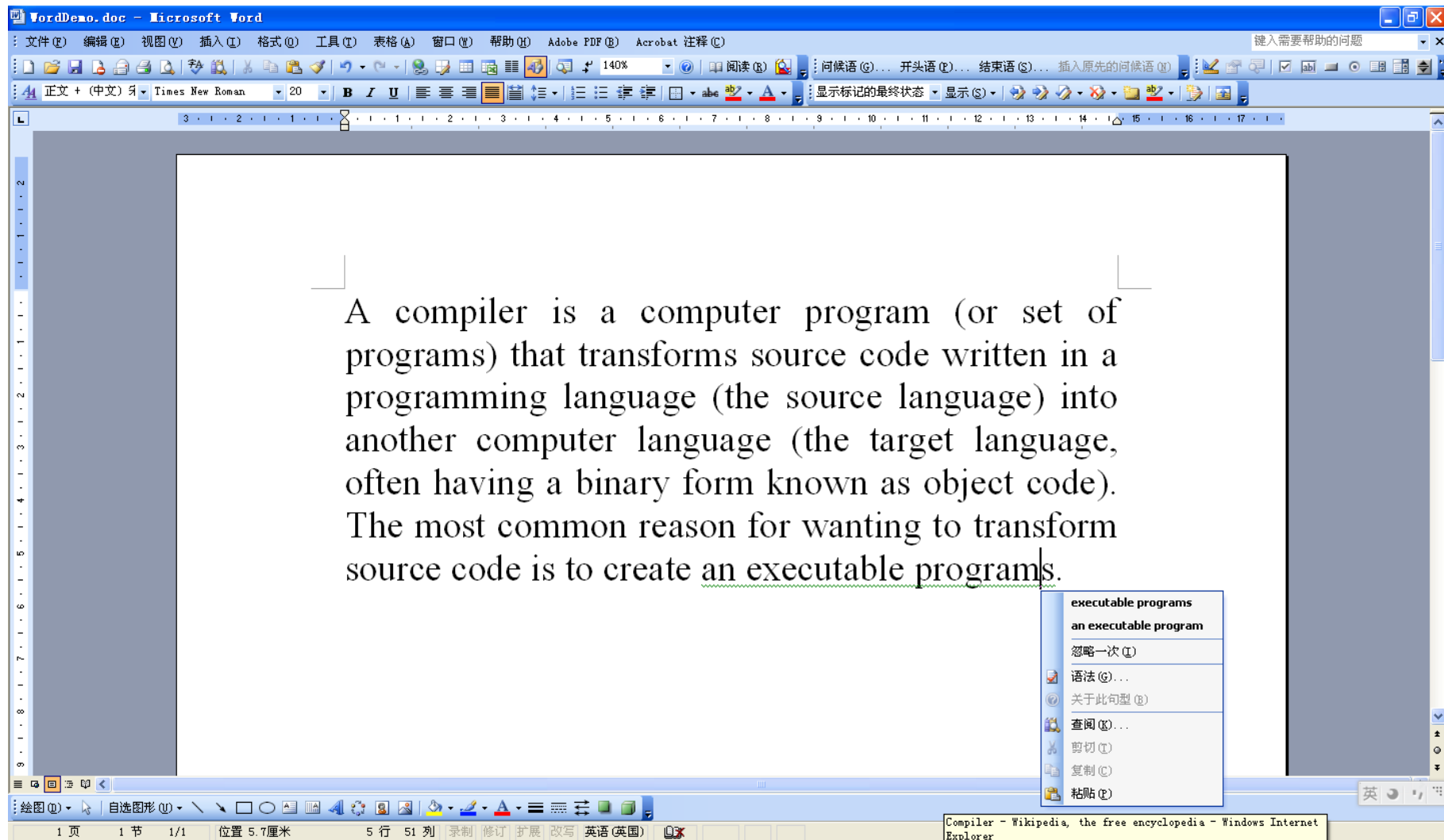
```
<HTML><HEAD><TITLE>HTML 语言教程</TITLE>
<META http-equiv=Content-Type content="text/html; charset=gb2312">
<META content="MSHTML 6.00.6000.16705" name=GENERATOR></HEAD>
<BODY text=#000000 bgColor=#ffffff>
<H1 align=center><FONT size=+3>HTML 语言教程: 目录</FONT></H1>
<P>
<HR>

<P></P>
<CENTER>
<P><A href="http://www.gzsums.edu.cn/webclass/html/new.html">新的特色</A> *** <A
href="http://www.gzsums.edu.cn/webclass/html/tag-index.html">HTML
标记(Tag)的索引(Index)</A></P></CENTER>
<CENTER><A
href="http://www.gzsums.edu.cn/webclass/html/page.html">页面(Page)</A> <A
href="http://www.gzsums.edu.cn/webclass/html/font.html">字体(Font)</A> <A
href="http://www.gzsums.edu.cn/webclass/html/text_style.html">文字布局(Text
Style)</A> <A
href="http://www.gzsums.edu.cn/webclass/html/image.html">图象(Image)</A> <A
href="http://www.gzsums.edu.cn/webclass/html/form.html">表单(Form)</A>
<P><A href="http://www.gzsums.edu.cn/webclass/html/table.html">表格(Table)</A> <A
href="http://www.gzsums.edu.cn/webclass/html/table02.html">表格进阶(Advanced)</A> <A
href="http://www.gzsums.edu.cn/webclass/html/frame.html">多窗口页面(Frames)</A>
<P><A
href="http://www.gzsums.edu.cn/webclass/html/marquee.html">会移动的文字(Marquee)</A> <A
href="http://www.gzsums.edu.cn/webclass/html/inline.html">多媒体页面(Alternative
Inline Elements)</A></CENTER>
<P>
```

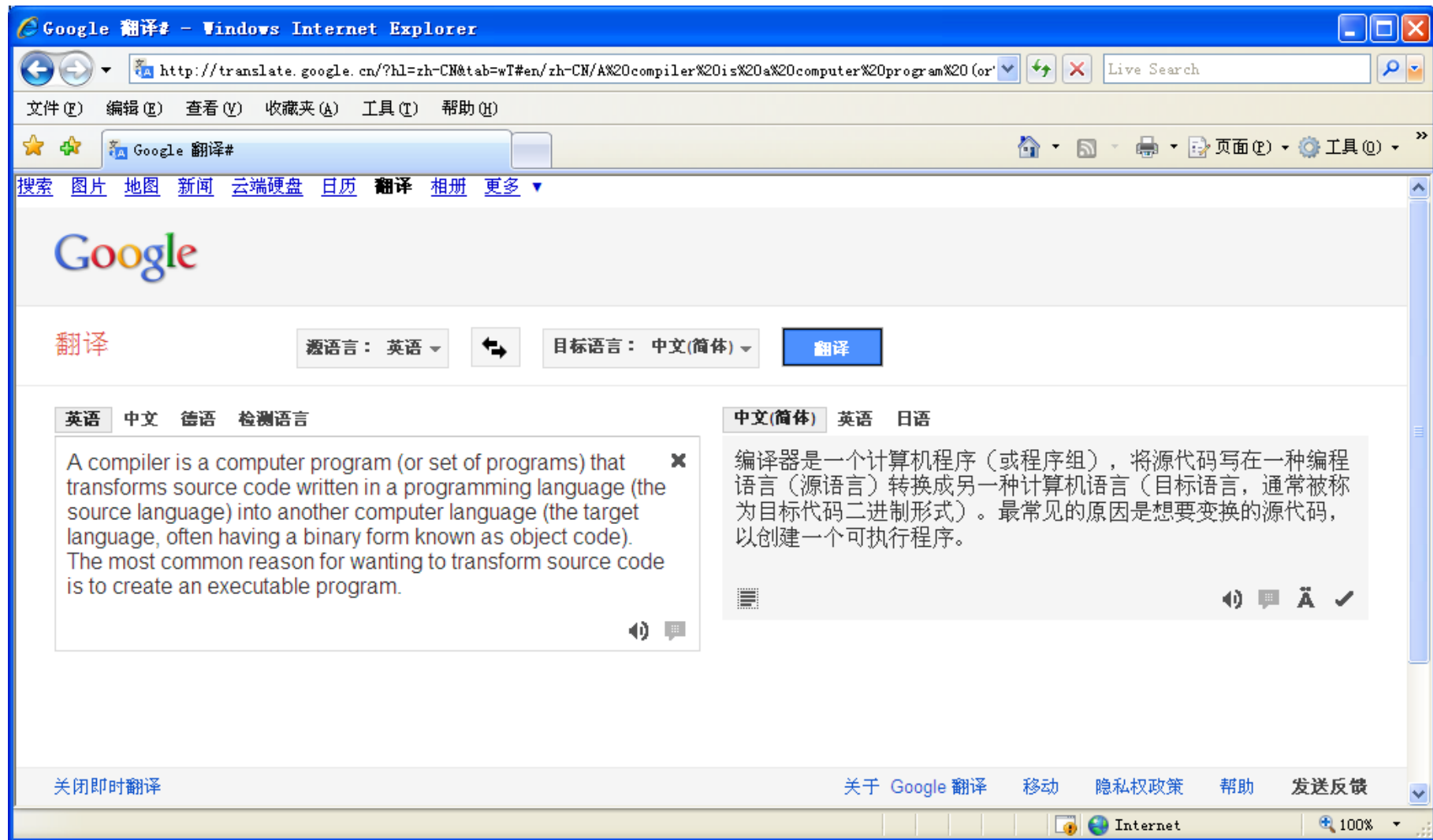
编译原理和方法的应用



编译原理和方法的应用



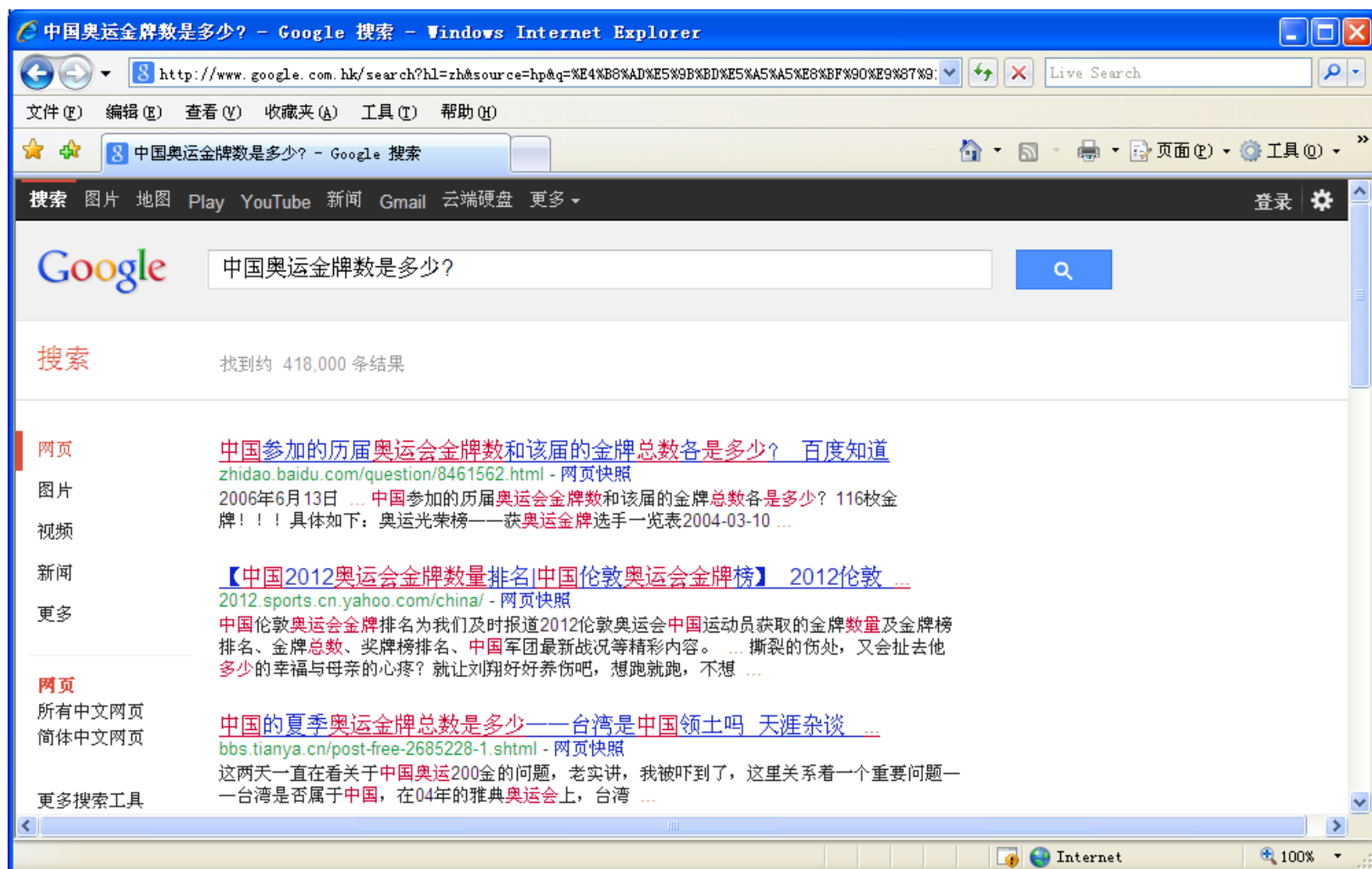
编译原理和方法的应用



编译原理和方法的应用



编译原理和方法的应用



编译原理和方法的应用

■ 更多有意思的应用

- 用户接口： Shell 命令解释器， ...
- 查询语言： SQL， XQuery， ...
- 网络协议： HTTP， SOAP， ...
- ...

But I Can't Make You Love Me if you don't

课程安排

■ 理论和实践相结合

□ 理论学习

- 理解：预习、听课
- 巩固：复习、作业

□ 实践环节

- 单元实验：词法分析 (LEX)、语法分析 (JavaCC, YACC)
- 综合实验：PL 语言编译器扩展

课程安排

■ 考核

- 闭卷笔试： 70%

- 上机实习： 25%

- 书面作业： 5%

■ 总成绩 = 笔试成绩 * 70% + 上机实习成绩 * 25% + 书面作业成绩 * 5%

课程安排

■ 参考书

- Compilers: Principles, Techniques, and Tools (2nd Edition). Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Addison Wesley; 2006
 - 中文版(本科教学版): 编译原理: Principles, Techniques, and Tools, 译者: 赵建华、郑滔、戴新宇, 机械工业出版社, 2009
- 《编译原理——学习指导与典型题解析》
 - 刘春林、王挺、周会平等, 2004 年, 国防工业出版社

小结

- 计算思维
- 什么是编译程序
 - 翻译、编译、解释
- 计算思维 vs. 编译
- 学习编译原理的意义
 - 编译原理和方法的应用
- 课程安排