Wuhan University

UNIT 01

绪论

武汉大学计算机学院程序设计课程组
主讲人：常 军

E-MAIL：chunscc@163.com
电 话：18986211771
QQ 群: 1020712774

本讲提纲

- 什么是计算机
- 计算机程序设计语言的发展
- 面向对象的方法
- 面向对象的软件开发
- 信息的表示与存储（自学）
- 程序的开发过程

0. 什么是计算机

程序设计角度看*计算机理论脉络*

图灵模型：计算机理论抽象

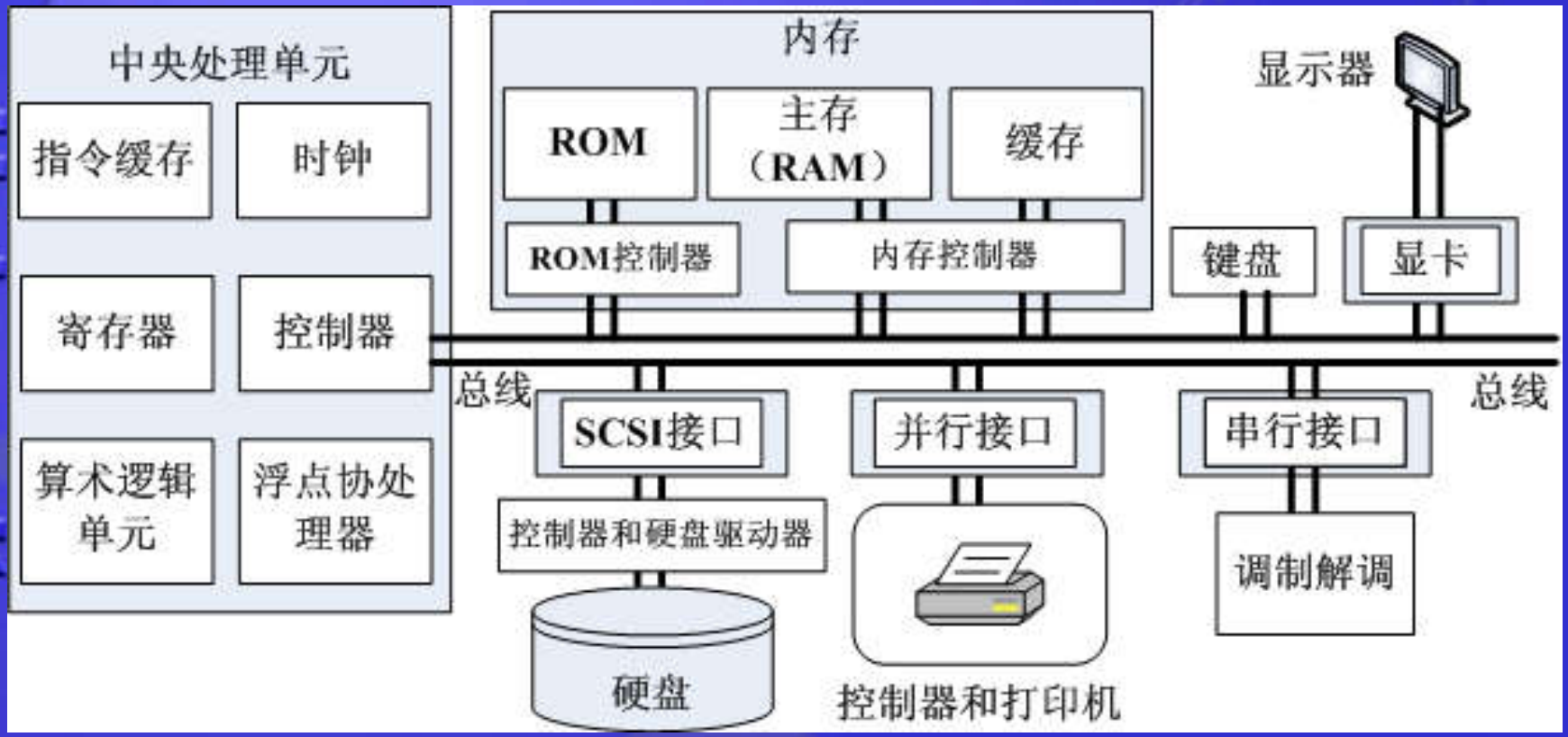
Alan Turing：图灵测试、图灵奖

冯诺依曼模型：电子计算机理论

程序和数据不加区分存储、1946年CRAY I

软件理论：算法、可计算性、逻辑

0.1、硬件是计算机系统的身体，软件是计算机系统的思维中枢

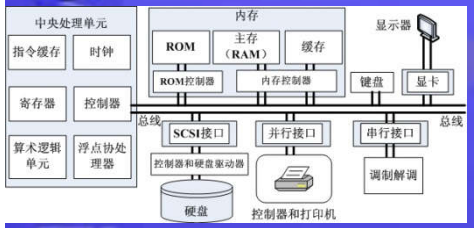


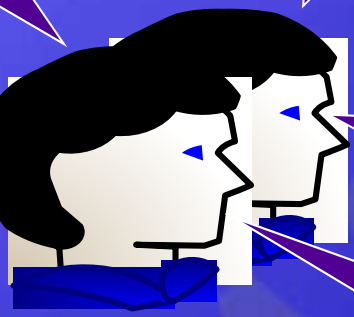
计算机的基本体系结构

0.1、硬件是计算机系统的身体，软件是计算机系统的思维中枢

CPU（中央处理单元）：计算机的大脑。

RAM（随机存储器，内存）：计算机的记忆。



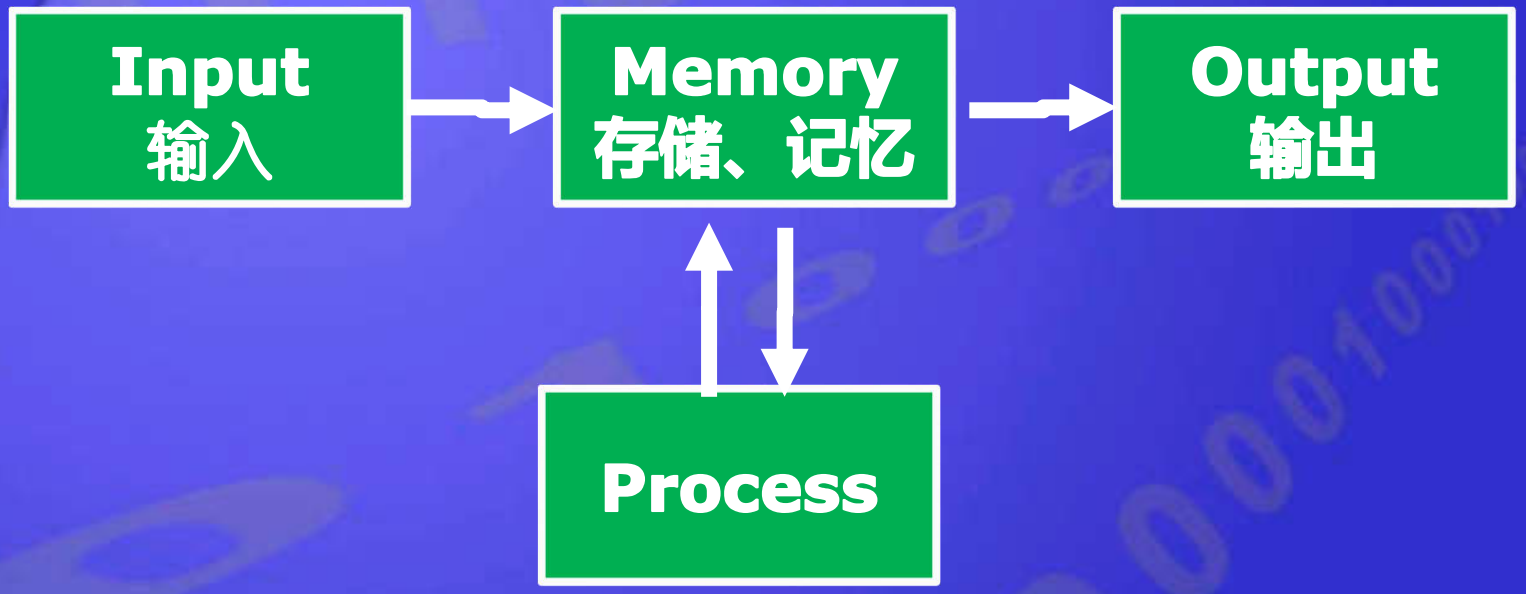


输入（INPUT）设备：计算机的感觉器官（视觉、触觉、味觉等）。

总线（BUS）：计算机的神经系统。

输出（OUTPUT）设备：计算机的操纵设备（手和声音）

功能划分：输入、输出、存储、加工



软件、硬件：无一例外

0.2、计算机的软件

系统软件： 由厂家提供，是用于管理和使用计算机的各种程序以及相应数据、文档的总称。

操作系统

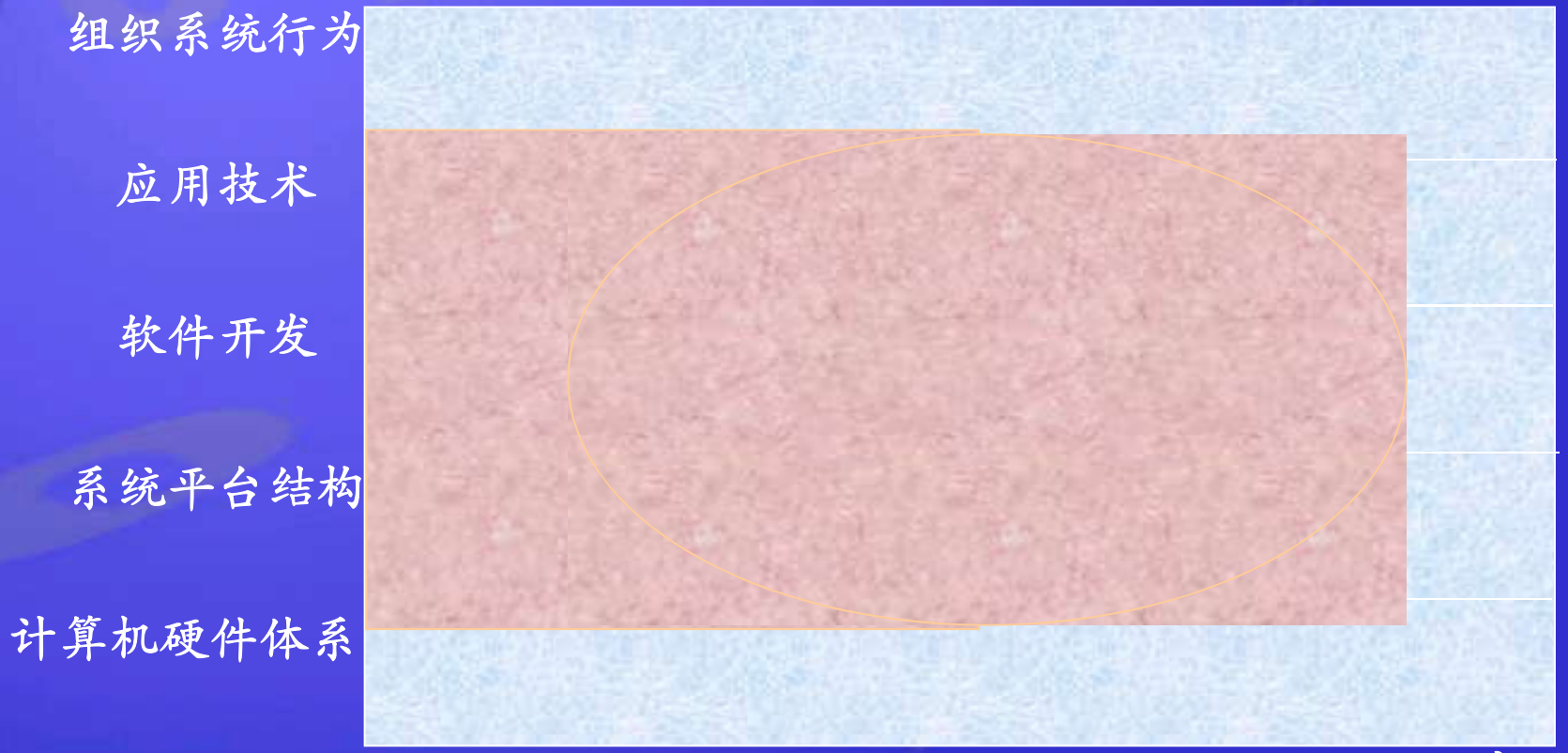
语言处理程序

数据库管理系统

为用户提供各种服务的各种实用程序（如Office图文排版程序、杀毒软件等）

应用软件： 为了使用计算机*处理各种实际问题*而专门研发的具有特定用途的程序。

计算机相关学科领域的问题空间

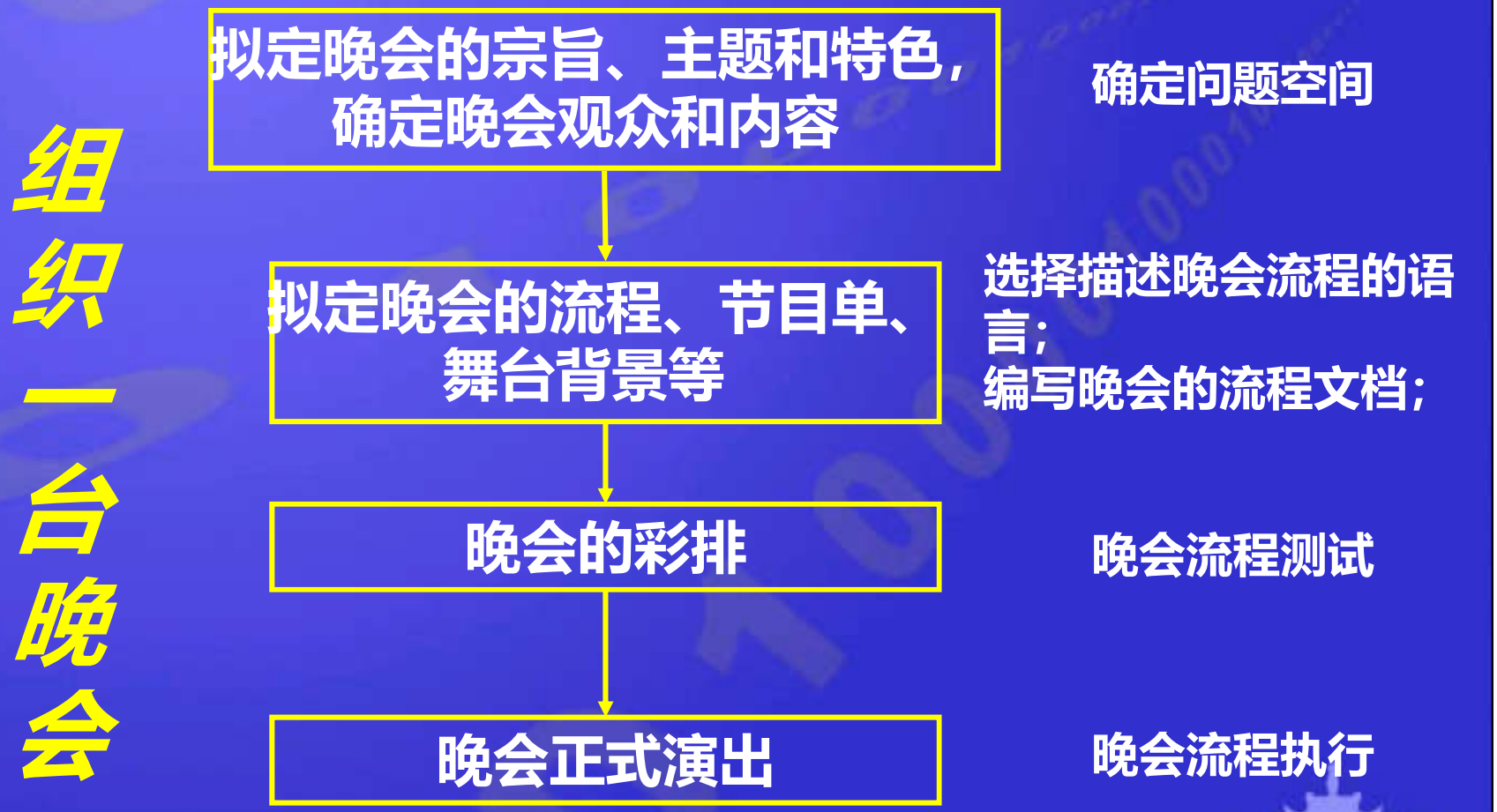


2、计算机学科的知识领域划分

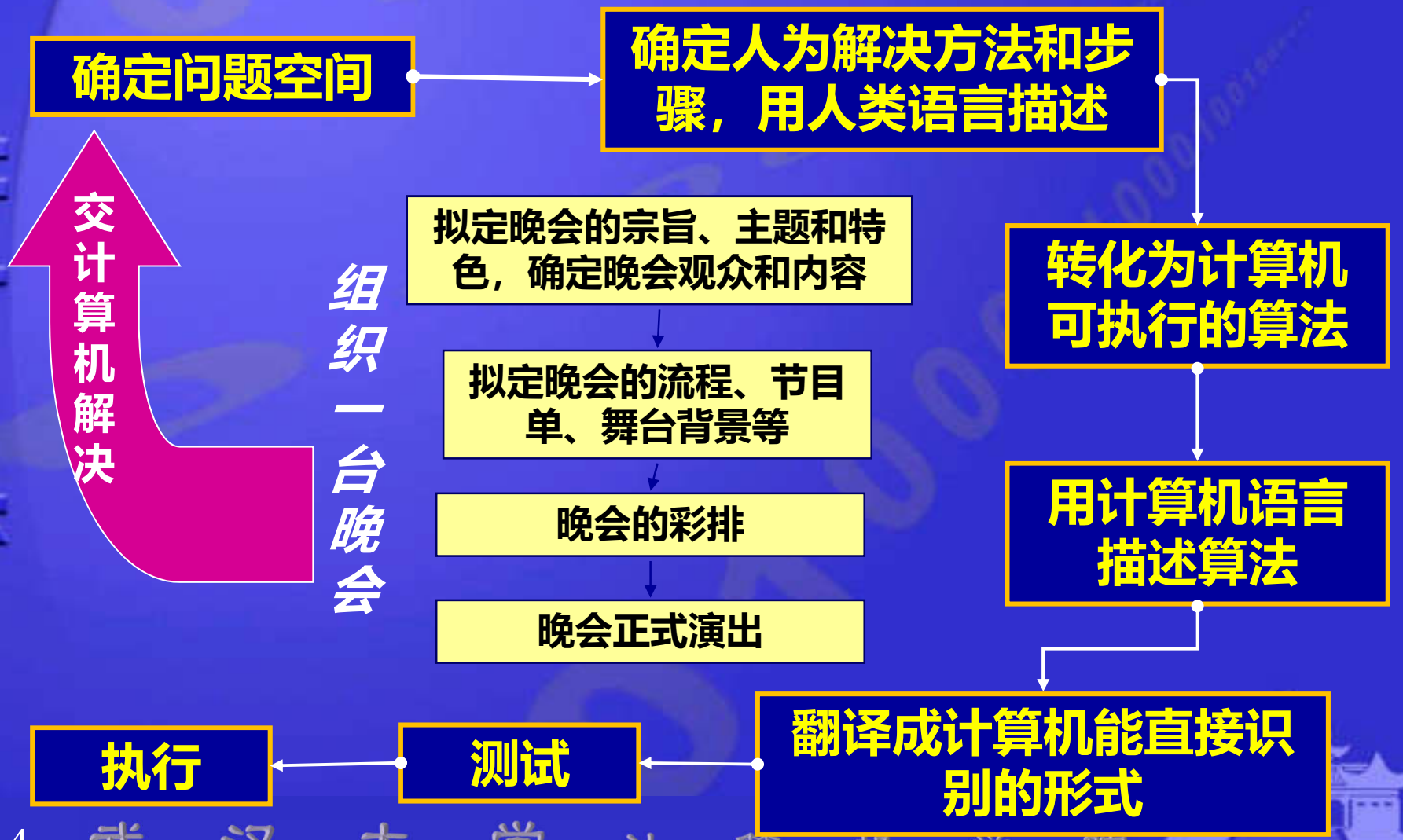
1. 计算机体系结构与组织	9. 计算机网络
2. 程序设计基础	10. 信息管理
3. 程序设计语言	11. 图形学和可视化
4. 算法分析	12. 智能系统
5. 人机交互	13. 数据库管理
6. 操作系统	14. 编译原理
7. 软件工程	15. ……
8. 离散数学	

1. 计算机程序设计语言的发展

1.1.1、什么是程序和程序设计



1.1.1、什么是程序和程序设计



1.1.1、什么是程序和程序设计

程序(program)：指示计算机如何运行的指令集合。

源程序： 程序的描述，采用各种计算机语言编写而成，给程序员看的。源程序不能直接被计算机执行。

目标程序： 编译后的程序。

机器代码： 从源程序翻译而成的计算机能够直接识别的机器语言代码程序。

1.1.1、什么是程序和程序设计

程序(program)：指示计算机如何运作的指令集合。

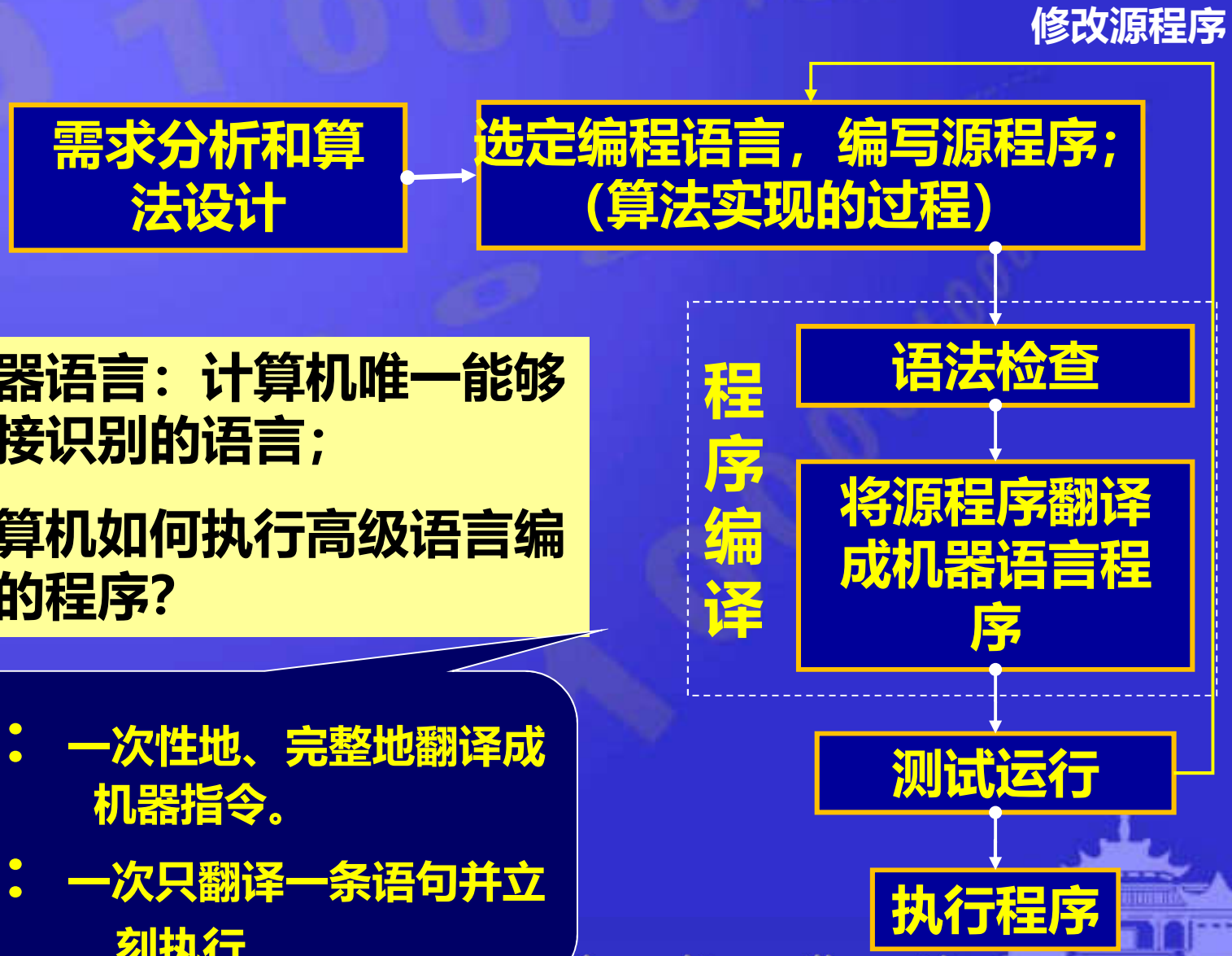
程序 = 数据结构 + 算法

程序 = 数据结构 + 算法 + 程序设计方法 + 语言工具

1.1.2、计算机语言

低级语言	第一代语言	机器语言	二进制语言
	第二代语言	汇编语言	符号化的机器指令
高级语言	第三代语言	结构化高级语言	自然语言，强调代码清晰度；代表：BASIC、Fortran、Pascal、C……
	第四代语言	非结构化高级语言	面向对象、可视化编程环境，如Visual C++、Power Builder、Delphi、Java等；描述型语言，例如Prolog、LISP等

1.1.2、计算机语言



机器语言：计算机唯一能够直接识别的语言；

计算机如何执行高级语言编写的程序?

编译： 一次性地、完整地翻译成机器指令。

解释： 一次只翻译一条语句并立刻执行。

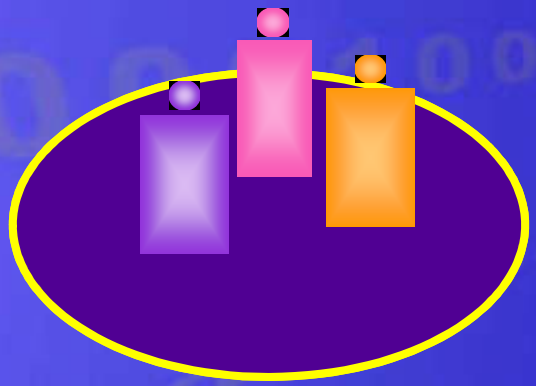
1.1.3 面向对象的语言

- **出发点：** 更直接地描述客观世界中存在的事物(对象)以及它们之间的关系。

- **特点：**
  - 是高级语言。
  - 将客观事物看作具有属性和行为的对象。
  - 通过抽象找出同一类对象的共同属性和行为，形成类。
  - 通过类的继承与多态实现代码重用

1.1.3 面向对象的语言（续）

- **优点：**使程序能够比较直接地反映问题域的本来面目，软件开发人员能够利用人类认识事物所采用的一般思维方法来进行软件开发。



2. 面向对象的方法

1.2.1 面向对象方法的由来
——面向过程的程序设计方法

最早的程序

- 目的：用于数学计算
- 主要工作：设计求解问题的过程
- 缺点：对于庞大、复杂的程序难以开发和维护

面向过程的结构化程序设计方法

设计思路

- 自顶向下、逐步求精。采用模块分解与功能抽象，自顶向下、分而治之。

程序结构：

- 按功能划分为若干个基本模块，形成一个树状结构。
- 各模块间的关系尽可能简单，功能上相对独立；每一模块内部均是由顺序、选择和循环三种基本结构组成。
- 其模块化实现的具体方法是使用子程序（函数）。

面向过程的结构化程序设计方法（续）

优点：有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子任务，便于开发和维护。

面向过程的结构化程序设计方法（续）

缺点：可重用性差、数据安全性差、难以开发大型软件和图形界面的应用软件

- 把数据和处理数据的过程分离为相互独立的实体。
- 当数据结构改变时，所有相关的处理过程都要进行相应的修改。
- 每一种相对于老问题的新方法都要带来额外的开销。
- 图形用户界面的应用程序，很难用过程来描述和实现，开发和维护也都很困难。

面向过程的结构化程序设计方法（续）

面向过程的结构化程序设计方法

解决问题的步骤

程序=算法+数据结构

优点：

有效地分解较复杂的任务；

可读性强；

便于开发和维护。

缺点：数据与操作分离

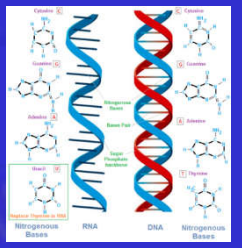
“过程+操作”的不稳定性：不能直接反映人求解问题的思路；

程序代码可重用性差；

代码一致性维护难度大。

面向过程的结构化程序设计方法（续）

- 面向过程的方法
  - 我们的世界是由一个个相互关联的小系统组成
  - 首先找到过程的起点
  - 然后分析过程中每一个部分，直到过程的终点
  - 过程中的每一个部分都是过程链上不可分割的一环



将世界视为过程这个方法本身蕴含着一个前提假设：

这个过程是稳定的！

可惜这个世界唯一不变的就是变化！

面向过程的结构化程序设计方法（续）

面向过程的困难

- ✓ 面向过程的方法并非不正确
- ✓ 只是构成一个系统的因素太多
- ✓ 要考虑所有可能的因素及因果关系很难
- ✓ 把这个过程模拟出来更难！

我们精力有限，计算能力有限，只能放弃对整个过程的了解！

重新寻找一个新方法，将复杂的系统转化成一个个我们可以控制的小单元！

1.2.2 面向过程还是面向对象

- 面向对象兴起之前，编程以过程为中心
- 面向过程构建的系统已经到达超越其处理能力的复杂性极点！
- 对象能提升抽象级别构建更大更复杂的系统

本质上面向过程和面向对象是一个古而有之的认识论的问题：

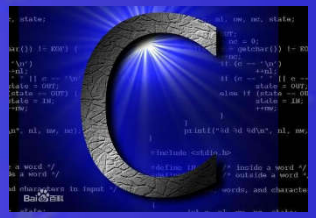
面向对象认识论帮助我们构建更为复杂的系统来解释越来越复杂的世界！

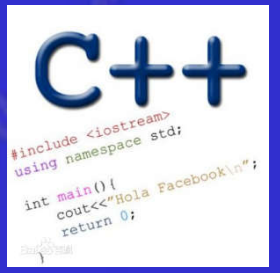
重要的是掌握认识论所采用的方法和分析过程，而不是技术本身！

“我们开发软件并不是为了让它面向对象化，或遵循其它的模式。

我们开发软件是为了解决问题！

1.2.2 面向过程还是面向对象





- 面向过程：结构化程序设计
- 面向对象：封装、继承、多态
- 技术只是认知世界的工具，只是表征不是本征

UML创始人Grady Booch：

“我对面向对象编程的目标从来都就不是复用。相反，对我来说，对象提供了一种处理复杂性问题的方式。”

亚里士多德：“您把这个世界视为过程还是对象？”

1.2.3 面向对象的程序设计

面向对象的程序设计方法

20世纪60～70年代萌芽 80年代完善 90年代以来繁荣	**代表语言：** C++、JAVA……	**设计思路：** 从实际世界中存在的事物出发去构建系统，尽量使用人的自然思维，如抽象分类等

行为模式：功能划分

对象＝算法＋数据结构

程序＝对象＋对象＋对象＋……

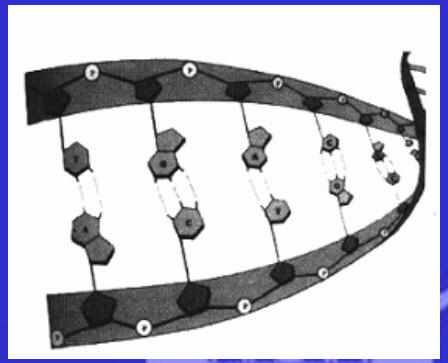
优点：

程序模块的独立性、数据的安全性就有了良好的保障。
通过继承与多态性，可以大大提高程序的可重用性，使得软件的开发和维护都更为方便。

1.2.3 面向对象的程序设计

面向对象的方法

- ✓ 我们的世界是由一个个相互独立的对象组成
- ✓ 对象之间无因果关系
- ✓ 平时：“鸡犬之声相闻，老死不相往来”
- ✓ 没有外力，对象保持着“静止”状态
- ✓ 外部力量驱动下，对象之间才会依据某种规律相关传递消息



1.2.3 面向对象的程序设计

什么是面向对象?

- 所谓面向对象的方法，就是使我们分析、设计和实现一个系统的方法尽可能自然地使用我们在生活中用到的以对象为中心的思想，分析、认识、设计一个系统的方法

Everything is an object

1.2.3 面向对象的程序设计

面向对象特性

- ✓ **封装：** 对象有坚硬的外壳，对象内部是黑匣子
- ✓ **继承：** 对象可以繁殖，孩子有父辈的全部本领
- ✓ **多态：** many objects has a same face!
- ✓ **聚合：** 对象可以结合一起形成新的对象
- ✓ **接口：** 对象都是多面派，根据要求展现一个面
- ✓ **依赖：** 对象知道与它有联系的一小群对象
- ✓ **耦合：** 与小伙伴间保持信息交流

1.2.4 面向对象的基本概念

对象

一般意义上的对象：

现实世界中实际存在的事物。

有形的（比如一辆汽车），或者无形的（比如一项计划）。

构成世界的一个独立单位，具有：

静态特征：可以用某种数据来描述

动态特征：对象所表现的行为或具有的功能

面向对象方法中的对象：

系统中用来描述客观事物的一个实体，它是用来构成系统的一个基本单位。对象由**一组属性和一组行为（或方法）**构成。

属性：用来描述对象静态特征的数据项。

方法：用来描述对象动态特征的操作序列。

1.2.4 面向对象的基本概念

对象

现实世界中对象的特征：

- 每一个对象必须有一个**名字**以区别于其他对象
- 用**属性**来描述对象的某些特征
- 有一组**操作**，每一个操作决定对象的一种行为

对象的操作可分为两类：

自身所承受的操作

施加于其他对象的操作

对象向外界提供的接口形式：使用对象唯一需要知道的信息。

1.2.4 面向对象的基本概念

对象

例如有一个人名叫小明，性别男，身高1.80m，体重68kg，可以修电器，可以教计算机课，下面来描述这个对象

对象名：小明

对象的状态：

- 性别：男
- 身高：1.80m
- 体重：68kg

对象的功能（可做的操作）：

回答身高
回答体重
回答性别

→ 均属于自身所承受的操作

修理电器
教计算机课

→ 属于施加与其他对象的操作

1.2.4 面向对象的基本概念

类

类的定义：

类是具有相同属性和行为的一组对象的集合，它为属于它的全部对象提供了统一的抽象描述，其内部包括属性和行为两个主要部分。

类是对象集合的再抽象。

例如，台式计算机是类，小明的台式计算机是对象。

类与实例的关系：

组成类的对象均为此类的实例（Instance）。

类是多个实例的综合抽象，实例是类的个体实物。

1.2.4 面向对象的基本概念

类

人类的定义：

```
class  HUMANKIND{
          char   *name;
          double  weight;
public:
          void       eat( );
          void      wear( );
          void      sleep( );
          void      travel( );
};
```

属性、特征、数据成员

行为、操作、函数成员

人类的实例：HUMANKIND XiaoMing ;

1.2.4 面向对象的基本概念

消息与事件

消息与事件的定义：

- 消息（Message）是描述事件发生的信息，它是对象之间发出的行为请求。
- 多个消息构成一个事件。
- 消息：是一个对象向另一个对象发出的执行某种操作的请求
- 消息的响应：对象执行操作被称为对消息的响应，也就是对一个对象的类成员函数的一次调用
- 发送者：发送消息的对象
- 接收者：接收消息的对象

1.2.4 面向对象的基本概念

消息与事件

消息的三个性质：

- 同一对象可接收不同形式的多个消息，产生不同的响应；
- 相同形式的消息可以送给不同的对象，所作出的响应可以是截然不同的；
- 消息的发送可以不考虑具体的接收者，对象可以响应消息，也可以对消息不予理会，对消息的响应并不是必须的。

1.2.4 面向对象的基本概念

方法

方法就是对象能执行的操作：

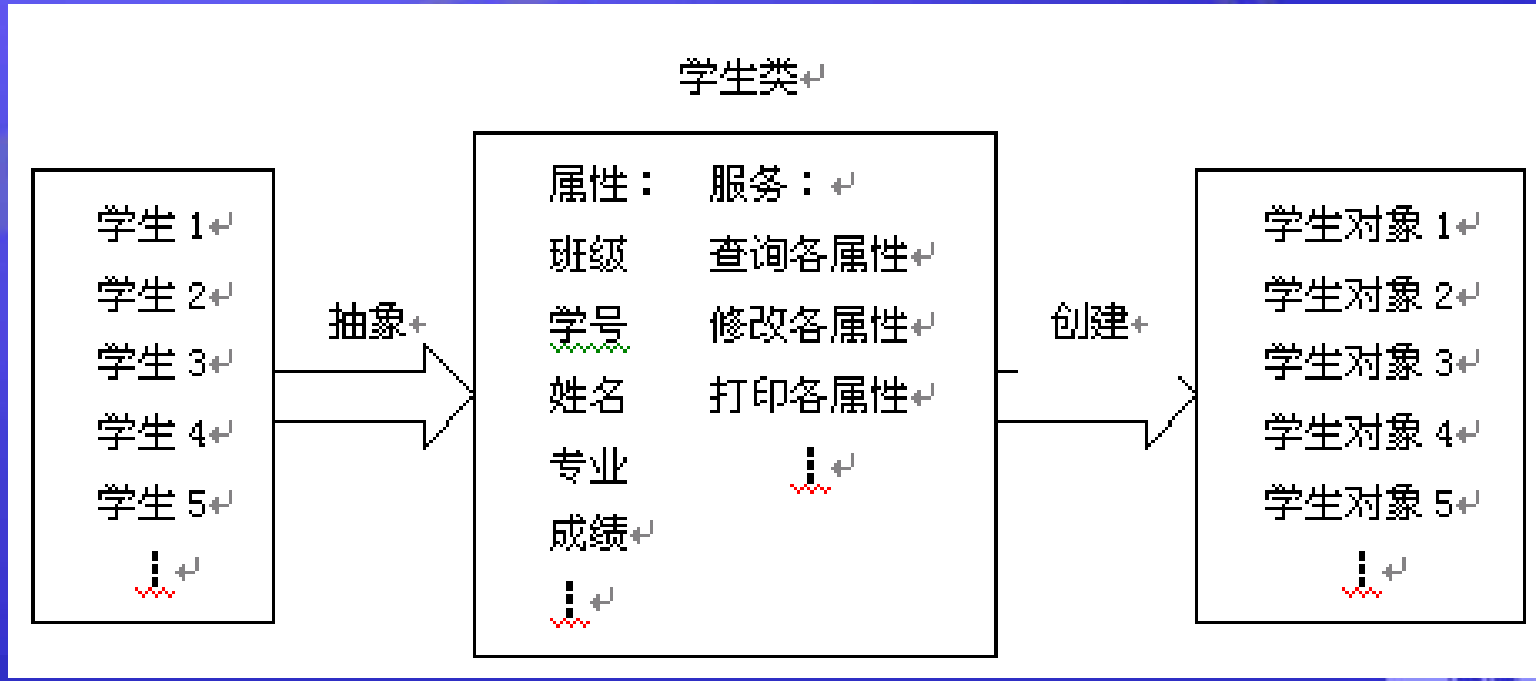
- 方法包括接口和方法体两个部分；
- 方法的接口：消息的模式，说明了方法的调用协议；
  （C++中成员函数的原型）
- 方法的方法体：某种操作的一系列计算步骤，即一段代码。
  （C++ 中成员函数的函数体）

1.2.5 面向对象的基本特征（抽象性）

抽象性(Abstract)

抽象就是忽略事物中与当前目标无关的非本质特征，更充分地注意与当前目标有关的本质特征。

数据抽象和行为抽象

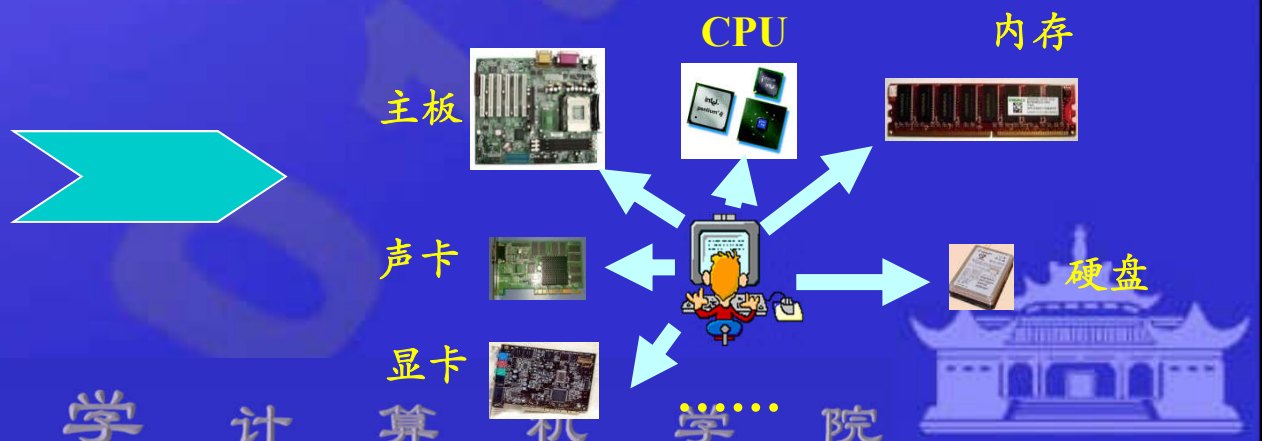


1.2.5 面向对象的基本特征（封装性）

封装性(Encapsulation)

- ✓ **封装**就是把对象的属性和行为结合成一个独立的单位，并尽可能隐蔽对象的内部细节。
- ✓ 其有两个含义：一是**封装性**，另一个是**“信息隐蔽”**。
  其一，是把对象的全部属性和行为结合在一起，形成一个不可分割的独立单位。
  其二，尽可能隐蔽对象的内部细节，对外形成一道屏障，与外部的联系只能通过外部接口实现。

信息隐藏原则



1.2.5 面向对象的基本特征（封装性）

封装性的几个条件：

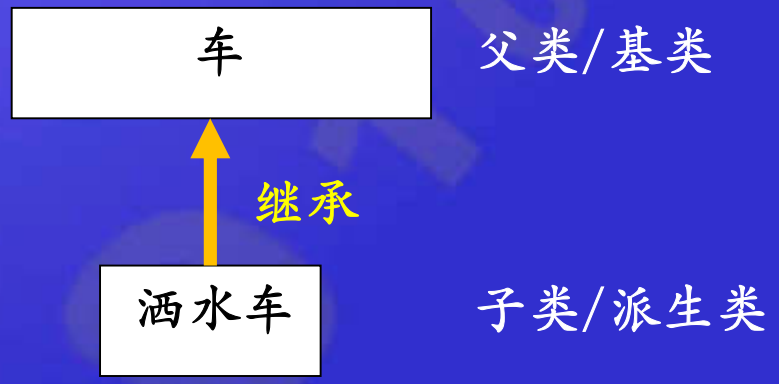
- ✓ 对象必须有一个清楚的边界：对象的私有数据和实现操作的代码都被封装在内；
- ✓ 具有一个描述对象与其他对象如何相互作用的接口，该接口必须说明消息传递的使用方法；
- ✓ 对象内部的代码和数据应收到保护，其他对象不能直接修改。

好处：对象的使用者和设计者分开，提供代码的复用性，减轻开发软件系统的难度。

1.2.5 面向对象的基本特征（继承性）

继承性（Encapsulation）

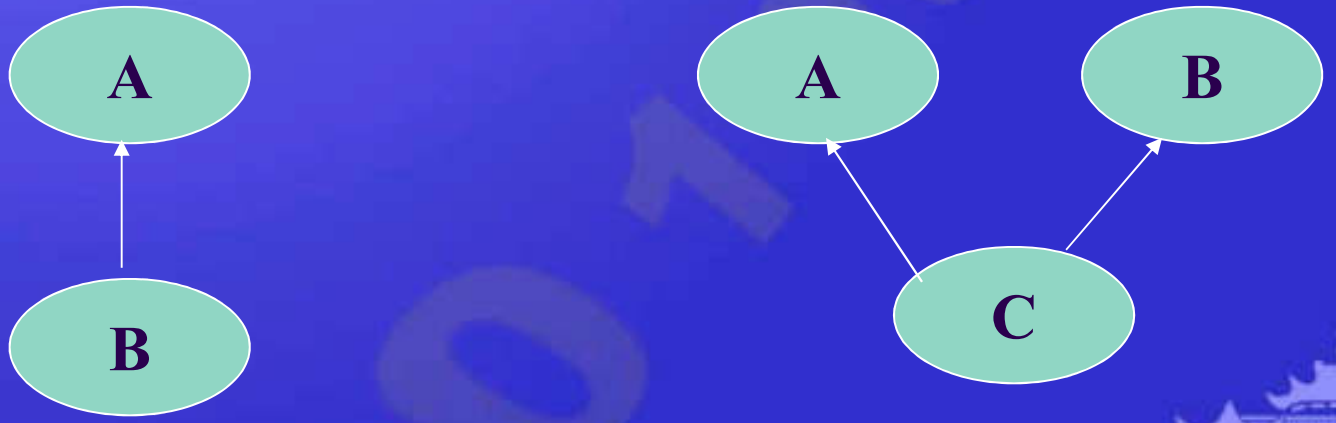
- ✓ **继承**是一种联结类与类的层次模型。**继承性**是指特殊类的对象拥有其一般类的属性和行为的特性。
- ✓ **继承意味着“自动地拥有”**，即特殊类中不必重新定义已在一般类中定义过的属性和行为，而它却自动地、隐含地拥有其一般类的属性与行为



1.2.5 面向对象的基本特征（继承性）

继承的分类（按继承来源划分）：

- ✓ **单继承：** 每个派生类只直接继承了一个基类的特征；
- ✓ **多继承：** 指多个基类派生出一个派生类的继承关系，多继承的派生类直接继承了不止一个基类的特征。



1.2.5 面向对象的基本特征（继承性）

继承的分类（按继承内容划分）：

- ✓ **取代继承：** 派生类对象**完整地**继承了基类的所有“属性”和“操作”，且**没有修改或增加**新的“属性”和“操作”。
- ✓ **包含继承：** 派生类对象**完整地**继承了所有基类的所有“属性”和“操作”，并增加了新的“属性”和“操作”。
- ✓ **受限继承：** 派生类对象**部分地**继承了基类的“属性”和“操作”，并且没有增加新的“属性”和“操作”。
- ✓ **特化继承：** 派生类对象继承了基类的“属性”和“操作”，并对原有“属性”和“操作”进行了**修改**。

1.2.5 面向对象的基本特征（多态性）

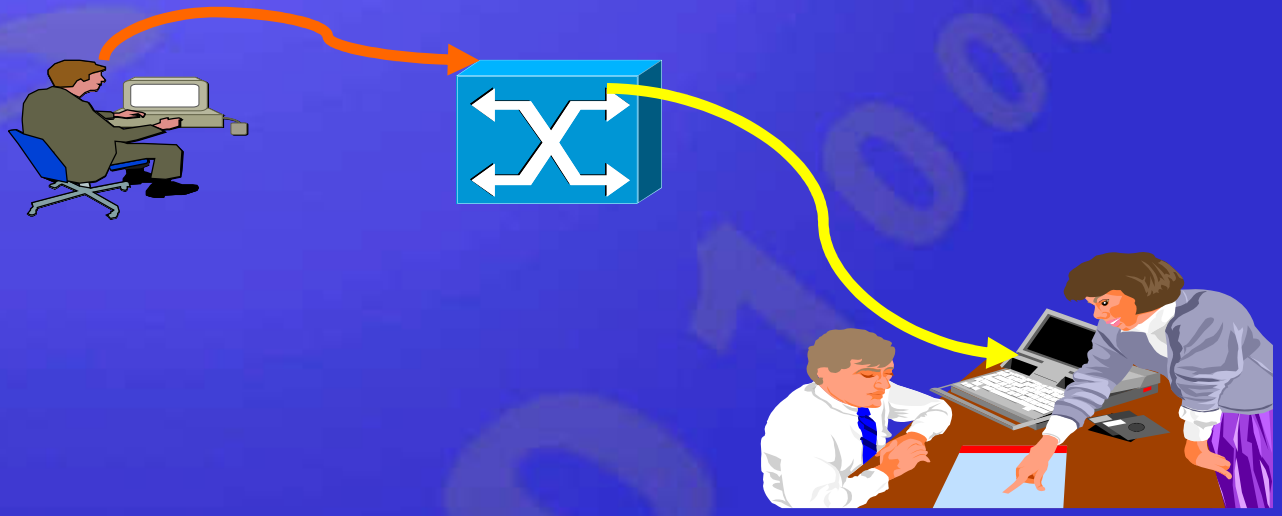
多态性（Polymorphism）是指类中同一函数名对应多个具有相似功能的不同函数，可以使用相同的调用方式来调用这些具有不同功能的同名函数的特性。

C++支持两种多态性：编译时的多态性和运行时的多态性。

- ✓ **重载：**编译时多态性的实现，多个函数具有相同的名字但具有不同的作用。

  函数重载　操作符重载
- ✓ **虚函数：**运行时多态性的实现，虚函数使用户在一个类等级中可以使用相同函数的多个版本。

3. 面向对象的软件开发



面向对象的软件工程

“软件危机”的出现促成软件工程学的形成和发展：

面向对象的软件工程是面向对象方法在软件工程领域的全面应用。它包括:

- ✓ 面向对象的分析（OOA）
- ✓ 面向对象的设计（OOD）
- ✓ 面向对象的编程（OOP）
- ✓ 面向对象的测试（OOT）
- ✓ 面向对象的软件维护（OOSM）

1.3.1 分析

系统分析阶段应该扼要精确地抽象出系统必须做什么，但是不关心如何去实现。

面向对象的系统分析，直接用问题域中客观存在的事物建立模型中的对象，对单个事物及事物之间的关系，都保留他们的原貌，不做转换，也不打破原有界限而重新组合，因此能够很好地映射客观事物。

1.3.2 设计

针对系统的一个具体实现运用面向对象的方法。其中包括两方面的工作：

- ✓ 把OOA模型直接搬到OOD，作为OOD的一部分
- ✓ 针对具体实现中的人机界面、数据存储、任务管理等因素补充一些与实现有关的部分。

1.3.3 编程

OOP工作就是用一种面向对象的编程语言把OOD模型中的每个成分书写出来，是面向对象的软件开发最终落实的重要阶段。

1.3.4 测试

测试的任务是发现软件中的错误。

在面向对象的软件测试中继续运用面向对象的概念与原则来组织测试，以对象的类作为基本测试单位，可以更准确地发现程序错误并提高测试效率。

1.3.5 维护

将软件交付使用后，工作并没有完结，还要根据软件的运行情况和用户的需求，不断改进系统。

使用面向对象的方法开发的软件，其程序与问题域是一致的，因此，在维护阶段运用面向对象的方法可以大大提高软件维护的效率。

基本术语

源程序：

用源语言写的，有待翻译的程序

目标程序：

也称为"结果程序"，是源程序通过翻译程序加工以后所生成的程序。

翻译程序：

是指一个把源程序翻译成等价的目标程序的程序。

S

三种不同类型的翻译程序

汇编程序：

其任务是把用汇编语言写成的源程序，翻译成机器语言形式的目标程序。

编译程序：

若源程序是用高级程序设计语言所写，经翻译程序加工生成目标程序，那么，该翻译程序就称为"编译程序"。

解释程序：

这也是一种翻译程序，同样是将高级语言源程序翻译成机器指令。它与编译程序不同点就在于：它是边翻译边执行的，即输入一句、翻译一句、执行一句，直至将整个源程序翻译并执行完毕。

程序的开发过程

编辑

将源程序输入到计算机中，生成后缀为cpp的磁盘文件。

编译

将程序的源代码转换为机器语言代码。

连接

将多个源程序文件以及库中的某些文件连在一起，生成一个后缀为exe的可执行文件。

运行调试

5. 程序的开发过程

程序设计的基本步骤

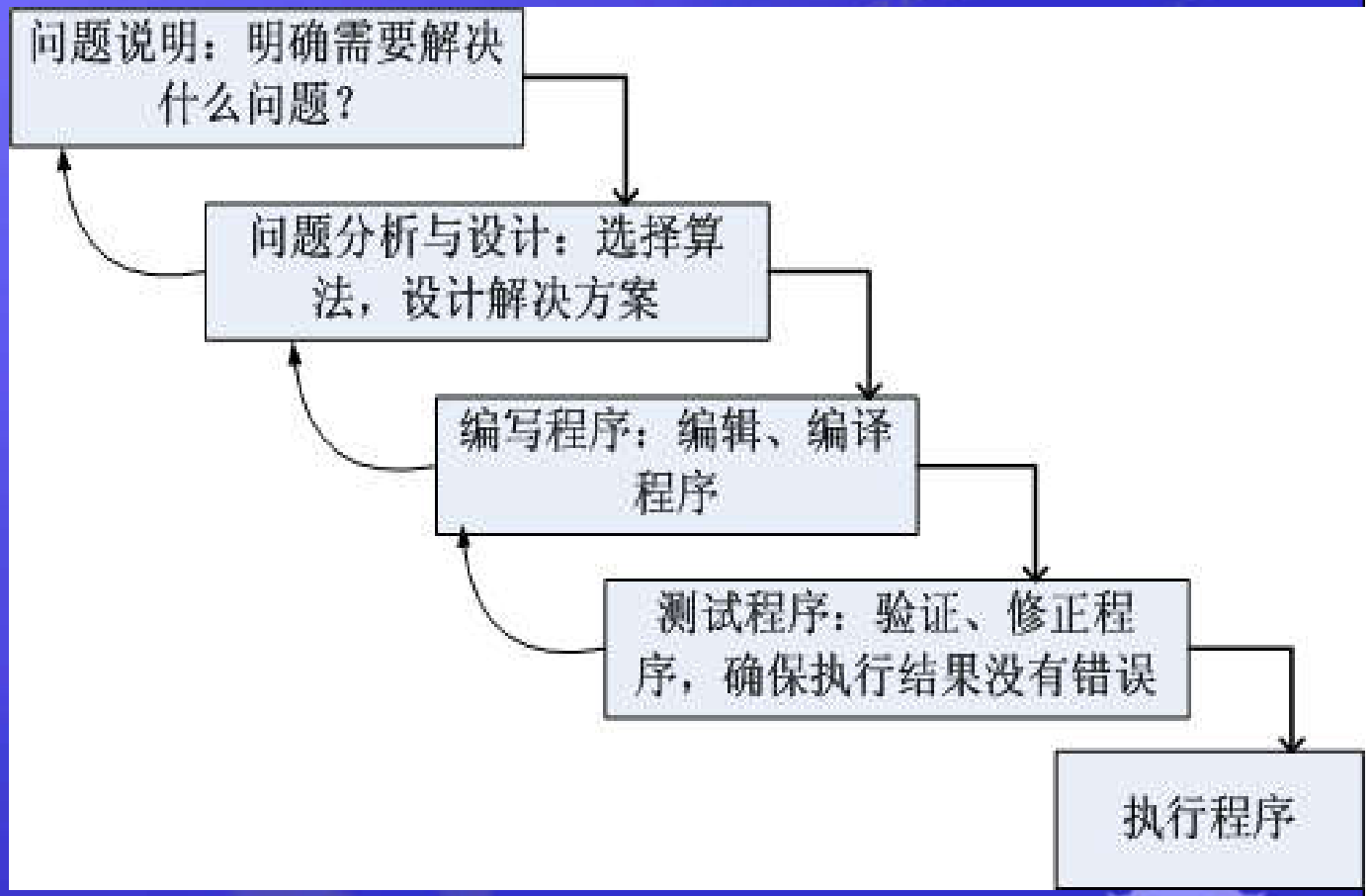
需求分析：选题

设计：组稿

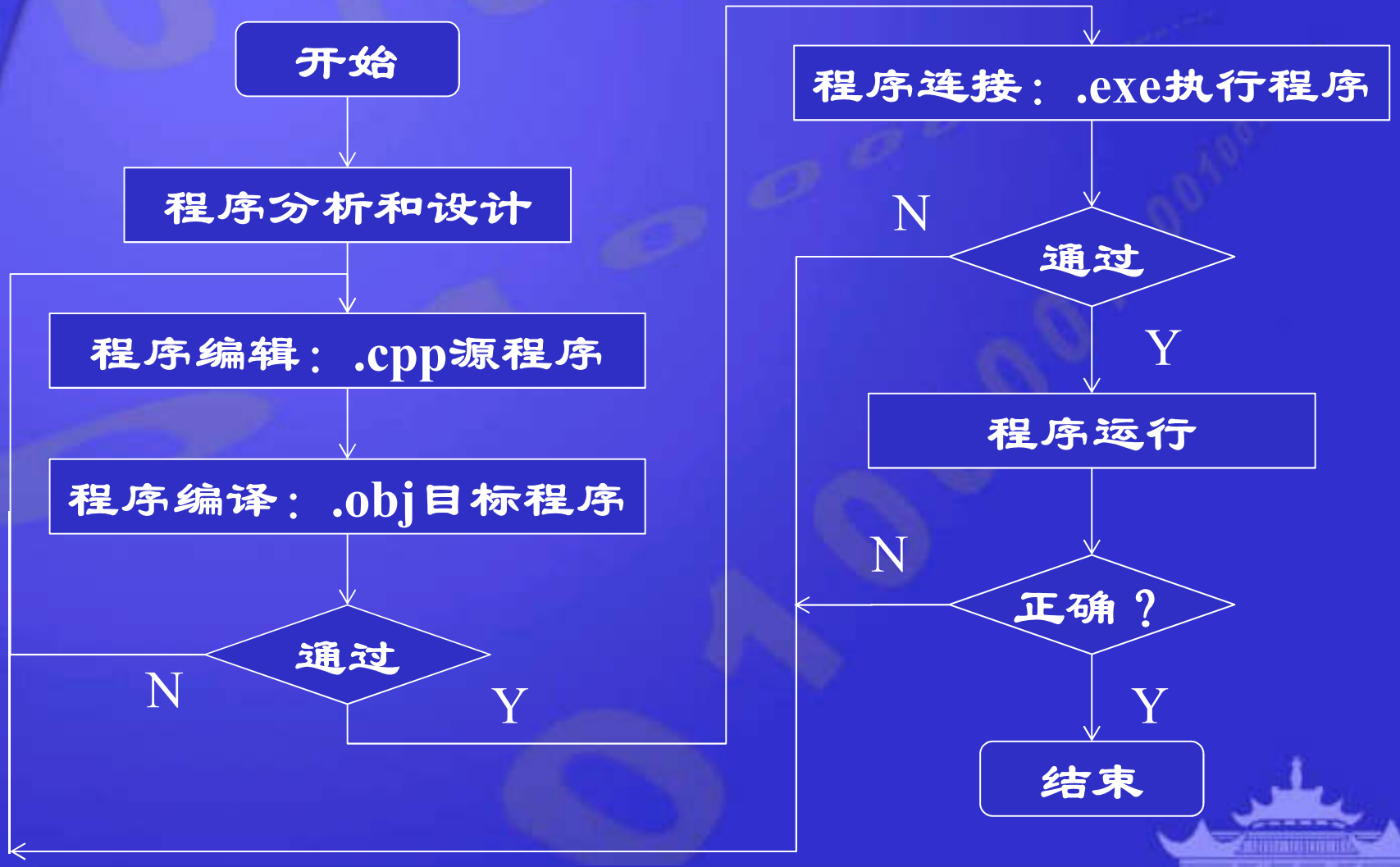
编辑：编著或编译

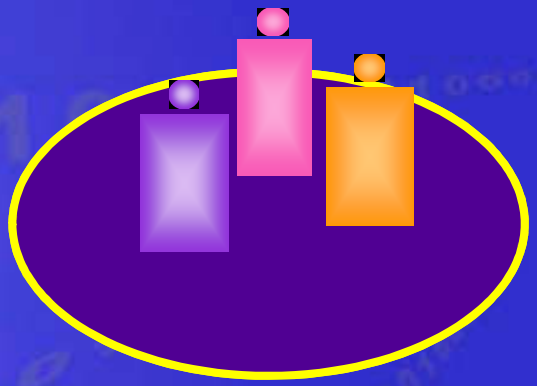
测试：编辑审稿

执行：加工出版



程序开发过程


开始
程序分析和设计
程序编辑：.cpp源程序
程序编译：.obj目标程序
通过
N
Y
程序连接：.exe执行程序
通过
N
Y
程序运行
正确？
N
Y
结束



本讲小结

本讲主要知识点

1. 计算机程序设计语言是计算机可以识别的语言，用于描述解决问题的方法，供计算机阅读和执行。计算机语言可以分为：低级语言（机器语言、汇编语言）、高级语言。面向对象的语言属于高级语言的一种。
2. 面向对象的基本概念：类与对象，封装性、继承性和多态性等。
3. 面向对象的软件工程是面向对象方法在软件工程领域的全面应用。
4. 计算机加工的对象是数据信息，而指挥计算机操作的是控制信息。所有的信息在计算机内部都是用二进制数表示的，具体的表示方式根据信息的类型有所不同。
5. 程序开发的过程

第1讲上机练习

学生用书：实验1

1. Visual Studio开发环境入门
2. Eclipse开发环境入门

第1讲　作业和课后练习

作业：没有

自学的课后练习，建议：

- 常用编译环境安装和使用
- 开始学习Linux
- 习题1-9、1-10、1-11

本讲结束