软件工程专业导论

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

第3讲 软件与程序思想:
组合-抽象-构造-递归

战德臣

哈尔滨工业大学　教授.博士生导师

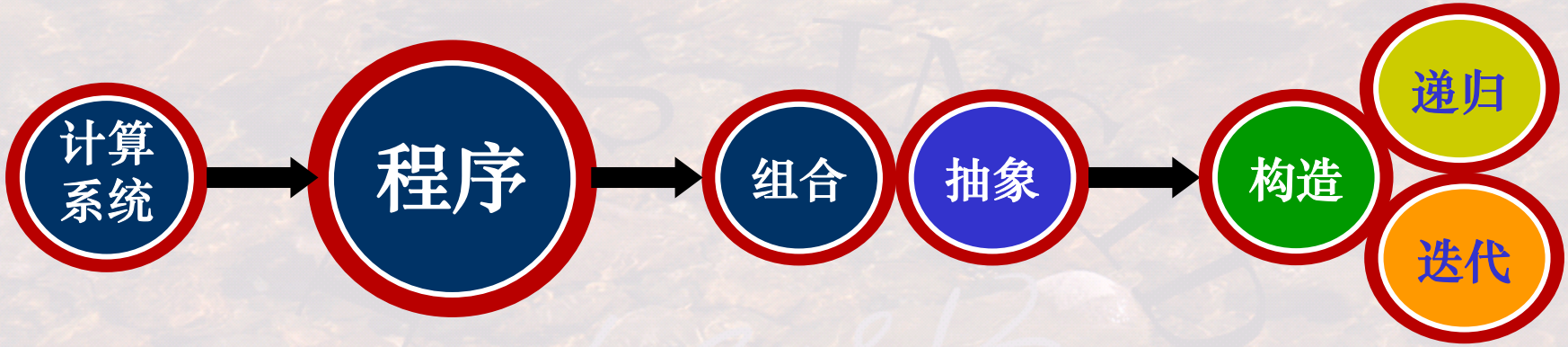
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

软件与程序思想: 组合-抽象-构造-递归

•软件工程学科的基本能力

战德臣 教授



--“程序”是实现计算系统复杂功能的一种重要手段

--程序的本质是组合、抽象与构造

--构造的基本手段是迭代和递归。递归是一种表达相似性对象及动作的重复性无限性构造的重要思想

--(各种)计算机语言仅仅是对软件与程序思想表达的规范化—以便于机器可以识别与执行

为什么需要程序?
----程序的作用和本质

战德臣

哈尔滨工业大学　教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

如何设计实现一个基本计算系统?

首先，设计并实现系统可以执行的基本动作(可实现的)，例如

“与”动作

“或”动作

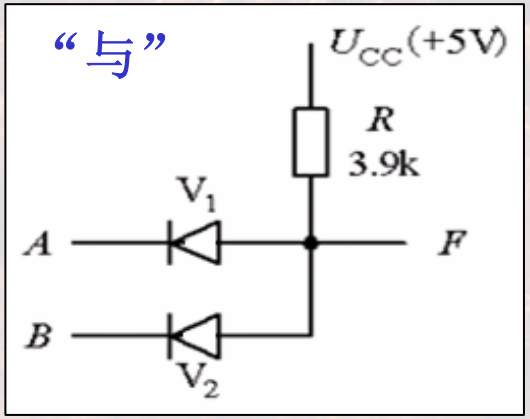
“非”动作

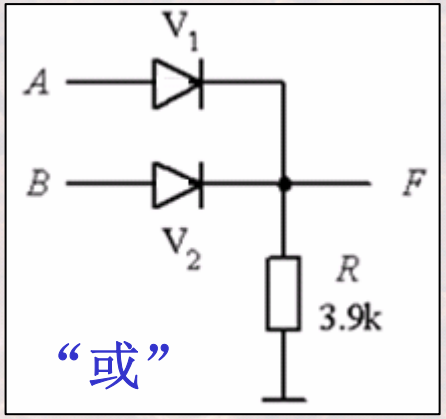
“异或”动作

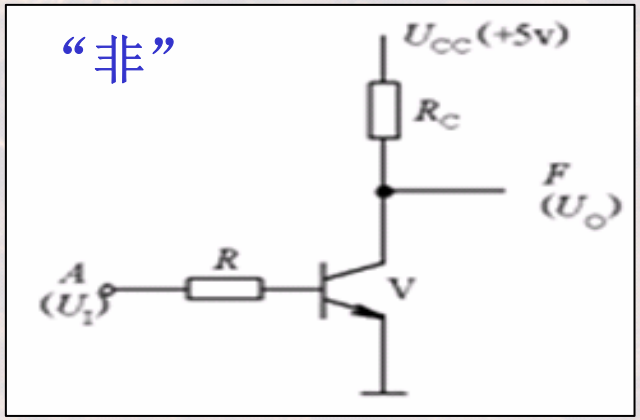
已知的基本事实是:

“加减乘除运算都可转换为加减法运算来实现”

“加减法运算又可以转换为逻辑运算来实现”







战德臣 教授

如何设计实现一个基本计算系统?

首先，设计并实现系统可以执行的基本动作(可实现的)，例如

- “与”动作
- “或”动作
- “非”动作
- “异或”动作

已知的基本事实是:

“加减乘除运算都可转换为加减法运算来实现”

“加减法运算又可以转换为逻辑运算来实现”

那么，复杂的动作呢?

- 系统需要提供复杂的动作
- 复杂的动作千变万化
- 复杂的动作随使用者使用目的的不同而变化

为什么需要程序--程序的作用和本质
(2)什么是程序?

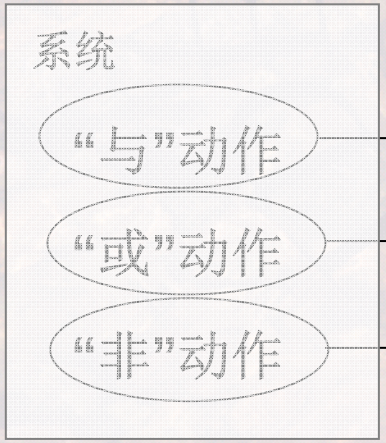
如何设计实现一个基本计算系统?

程序:由基本动作指令构造的，若干指令的一个组合或一个执行序列，用以实现复杂动作

复杂动作

((A AND B) AND C) OR (NOT C))

拆解开



AND

OR

NOT

X= A AND B
X= X AND C
Y= NOT C
X= X OR Y

指令：控制基本动作执行的命令

为什么需要程序--程序的作用和本质
(3)程序能否自动执行?

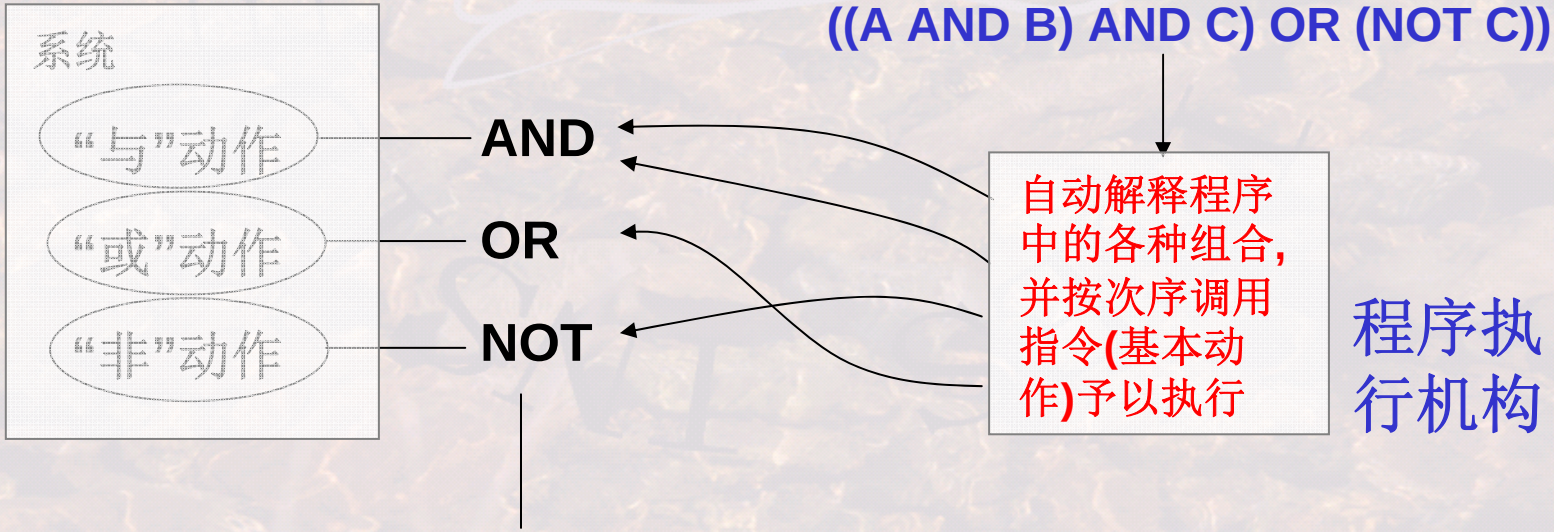
战德臣 教授

如何设计实现一个基本计算系统?

程序:由基本动作指令构造的，若干指令的一个组合或一个执行序列，用以实现复杂动作

复杂动作

((A AND B) AND C) OR (NOT C))

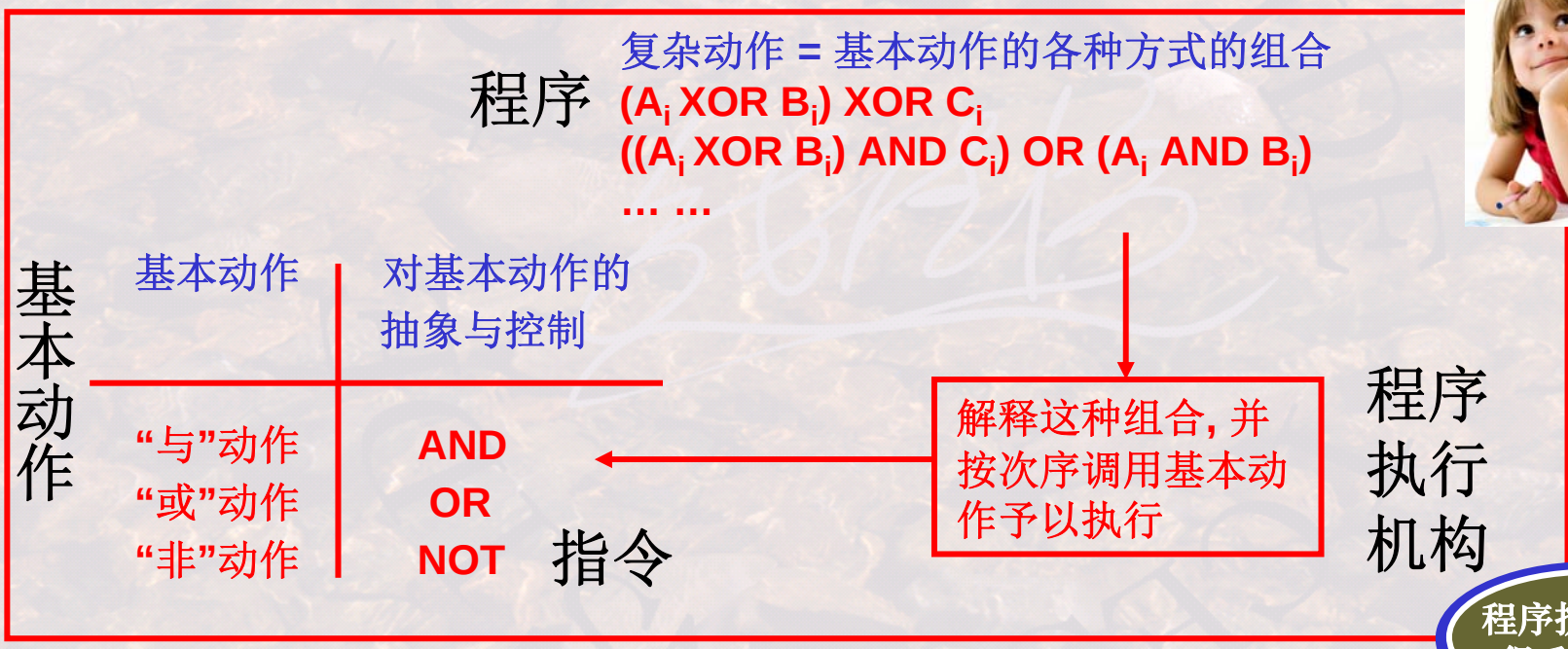


指令：控制基本动作执行的命令

为什么需要程序--程序的作用和本质
(4)计算系统与程序?

战德臣 教授

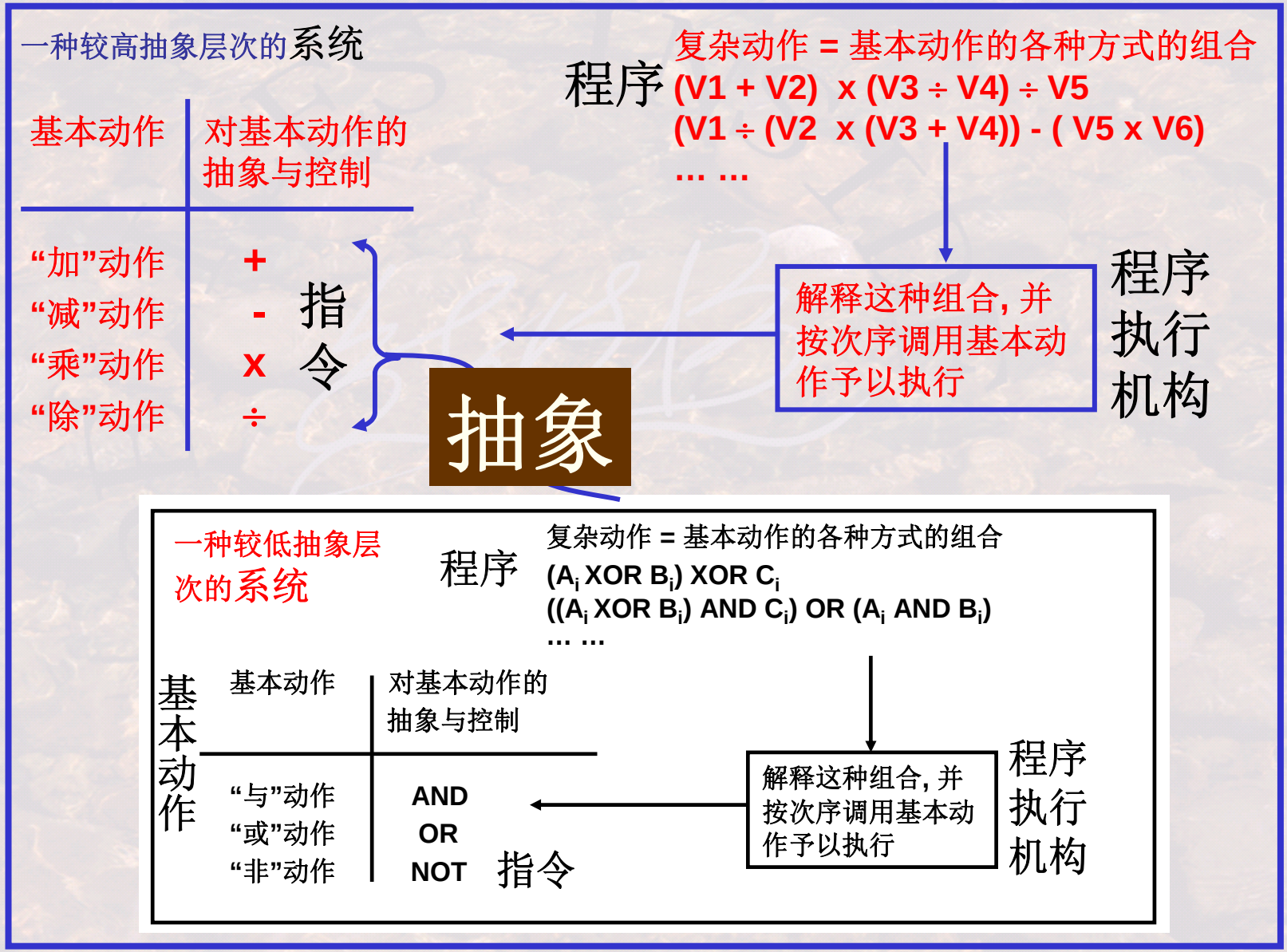
计算系统 = 基本动作 + 指令 + 程序执行机构



为什么需要程序--程序的作用和本质
(5)程序：组合-抽象-构造？

战德臣 教授

抽象：将经常使用的、可由低层次系统实现的一些复杂动作，进行**命名**，以作为高层次系统的指令被使用



组合-抽象-构造示例?
----基于运算组合式的构造

战德臣

哈尔滨工业大学　教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

组合-抽象-构造示例：基于运算组合式的构造
(1)前缀表示法与运算组合式

战德臣 教授

100

205

实际的数值

(100 + 205)

中缀表示法, 用运算符(即前述的指令)
将两个数值组合起来，运算符在中间

(+ 100 205)

前缀表示法, 用运算符(即前述的指令)
将两个数值组合起来，运算符在前面

将运算符表示的操作应用于后面的一组数值上，求出结果

(运算符 操作数1 操作数2)

- ( )括号给出了运算组合式的边界，是运算组合式的一部分
- 一组括号内，只能有一个运算符，但可有多个操作数，其间以空格区分。

组合-抽象-构造示例：基于运算组合式的构造
(1)前缀表示法与运算组合式

ICES 战德臣 教授

(运算符 操作数1 操作数2)

前缀表示法有什么优点吗?

- 基本运算符，如+，-，×，/。

(+ 100 205 300 400 51 304) 一个运算符可以表示连加，连减等情况，

(运算符 操作数1 操作数2 操作数3 操作数4)

(- 500 205 50 100 10 20) 一个运算符可以表示连加，连减等情况，

“指令”是否也是此形式呢?

操作码 地址码

战德臣 教授

组合：将一个运算组合式作为参数代入到另一个运算组合式中

(运算符1 操作数1 操作数2)

(运算符2 操作数a 操作数b)

(运算符1 (运算符2 操作数a 操作数b) 操作数2)

$$\frac{205+\frac{15+3}{90-8\times 8}}{200-10\times 5}$$

(/ 操作数1 操作数2)

(/ (+ 205 操作数a2) (- 200 操作数a4))

(/ (+ 205 (/ 操作数b1 操作数b2)) (- 200 (× 10 5)))

(/ (+ 205 (/ (+ 15 3) (- 90 操作数c1))) (- 200 (× 10 5)))

(/ (+ 205 (/ (+ 15 3) (- 90 (× 8 8)))) (- 200 (× 10 5)))

组合-抽象-构造示例：基于运算组合式的构造

(3)组合构造及构造的计算过程?

战德臣 教授

(+ 100 205)

(+ (+ 60 40) (- 305 100))

((* 3 (+ (* 2 4) (+ 3 5))) (+ (- 10 7) 6))*

计算过程

((* 3 (+ (* 2 4) (+ 3 5))) (+ (- 10 7) 6))*

((* 3 (+ 8 8)) (+ 3 6))*

((* 3 16) 9)*

(48 9)*

432

组合-抽象-构造示例：基于运算组合式的构造
(4)如何用名字简化运算组合式的构造?--计算对象的命名--抽象

ICES 战德臣 教授

命名计算对象和构造中使用名字及计算中以计算对象替换名字

(运算符 操作数1 操作数2)

(define 名字 操作数2)

- 基本运算符，如**define**。

(define height 2) ——— *名字的定义：定义名字**height**与**2**关联，以后可以用**height**来表示**2***

一种类型的名字：数值型的名字

(+ (+ height 40) (- 305 height)) ——— *名字的使用*

(+ (* 50 height) (- 100 height))

*注意：不同类型的对象可以有不同的定义方法。这里统一用**define**来表示，在具体的程序设计语言中是用不同的方法来定义的*

组合-抽象-构造示例：基于运算组合式的构造

(4)如何用名字简化运算组合式的构造?--计算对象的命名--抽象

战德臣 教授

命名计算对象和构造中使用名字及计算中以计算对象替换名字

```
(define pi 3.14159)
(define radius 10)
(*  pi  (* radius radius))
```

名字的定义

名字的使用

```
(*  pi  (*  radius radius))
(*  pi  (* 10  10))
(*  pi  100)
(*  3.14159  100)
314.159
```

名字的替换

计算

```
(define  circumference  (* 2 pi radius))
(*  circmference  20)
```

```
(*  circmference  20)
(*  (*  2  pi  radius) 20)
(*  (*  2  3.14159  10) 20)
(*  62.8318  20)
1256.636
```

计算

组合-抽象-构造示例：基于运算组合式的构造

(5)如何用名字简化运算组合式的构造?—新运算符与新运算组合式--抽象

钱德臣 教授

命名新运算符和构造中使用新运算符及执行中以过程替换新运算符

能定义新运算组合式吗?

(运算符 操作数1 操作数2)

(define 名字 操作数2)

(新运算符 操作数a1 操作数a2 …)

组合式整体作为一个名字

(define (新运算符 操作数a1 操作数a2 …) (运算组合式P))

(运算符 操作数1 操作数2)

(define (新运算符组合式) (运算组合式P))

- 将运算组合式P，定义为以新运算符引导的一个新运算符组合式

注意：不同类型的对象可以有不同的定义方法。这里统一用define来表示，在具体的程序设计语言中是用不同的方法来定义的

组合-抽象-构造示例：基于运算组合式的构造

(5)如何用名字简化运算组合式的构造?—新运算符与新运算组合式--抽象

钱德臣 教授

命名新运算符和构造中使用新运算符及执行中以过程替换新运算符

```
(define (square  x)  (* x  x))
```

x^2

名字的定义：*定义名字square为一个新的运算，即过程或称函数*
另一种类型的名字：运算符型的名字

新运算符，即过程名或函数名

形式参数，使用时将被实际参数所替代

过程体，用于表示新运算符的具体计算规则，其为关于形式参数x的一种计算组合。

```
(square  3)
(square  6)
```

名字的使用

注意：不同类型的对象可以有不同的定义方法。这里统一用define来表示，在具体的程序设计语言中是用不同的方法来定义的

战德臣 教授

命名新运算符和构造中使用新运算符及执行中以过程替换新运算符

(square 10) ——————— *名字的使用*

(square (+ 2 8))

(square (square 3))

(square (square (+ 2 5)))

(define (SumOfSquare x y) (+ (square x) (square y))

x^2+y^2

(SumOfSquare 3 4)

(+ (SumOfSquare 3 4) height)

组合-抽象-构造示例：基于运算组合式的构造
(6)程序构造—组合与抽象

战德臣 教授

命名新运算符和构造中使用新运算符及执行中以过程替换新运算符

```
(define (NewProc  a)  (SumOfSquare (+ a  1) (*  a 2)))
```

名字的定义

$(a+1)^2+(a*2)^2$

名字的使用

```
(NewProc  3)
(NewProc (+  3   1))
```

怎样执行呢？

组合-抽象-构造示例：基于运算组合式的构造
(7)构造程序的执行—求值、代入与计算

战德臣 教授

命名新运算符和构造中使用新运算符及执行中以过程替换新运算符

含名字的运算组合式的计算方法：求值、代入、计算

(NewProc (+ 3 1))的两种计算过程示意

(NewProc (+ 3 1))

(NewProc 4)

名字被替换

(SumOfSquare (+ 4 1) (* 4 2))

(SumOfSquare 5 8)

名字被替换

(+ (Square 5) (Square 8))

(+ (* 5 5) (* 8 8))

名字被替换

(+ 25 64)

89

先求值，再代入

组合-抽象-构造示例：基于运算组合式的构造
(7)构造程序的执行—求值、代入与计算

战德臣 教授

命名新运算符和构造中使用新运算符及执行中以过程替换新运算符

含名字的运算组合式的计算方法：代入、求值、计算

(NewProc (+ 3 1))的两种计算过程示意

(NewProc (+ 3 1))	*先代入，后求值*
(SumOfSquare (+ (+ 3 1) 1) (* (+ 3 1) 2))	
(+ (Square (+ (+ 3 1) 1) (Square (* (+ 3 1) 2)))	
(+ (* (+ (+ 3 1) 1) (+ (+ 3 1) 1)) (* (* (+ 3 1) 2) (* (+ 3 1) 2)))	*代入阶段*
(+ (* (+ 4 1) (+ 4 1)) (* (* 4 2) (* 4 2)))	*求值阶段*
(+ (* 5 5) (* 8 8))	
(+ 25 64)	
89	

组合-抽象-构造示例：基于运算组合式的构造
(8)你能表达与构造程序吗?

战德臣 教授

◆问题1：用前缀表示法书写下述表达式

$$\frac{10 + 4 + (8- (12 - (6 + 4\div 5)))}{3*(6-2)(12-7)}$$

◆问题2：请定义一个过程，求某一数值的立方

a^3

◆问题3：进一步以问题2定义的过程，再定义一个过程，求某两个数值的立方和。a^3+b^3 进一步求 5^3+8^3 ，并模拟给出计算过程。

组合-抽象-构造示例：基于运算组合式的构造
(8)你能表达与构造程序吗?

战德臣 教授

◆问题1：用前缀表示法书写下述表达式

$$\frac{10 + 4 + (8- (12 - (6 + 4\div5)))}{3*(6-2)(12-7)}$$

(/ 操作数1 操作数2)

(/ (+ 10 4 操作数a2) 操作数2)

(/ (+ 10 4 (- 8 操作数a3)) 操作数2)

(/ (+ 10 4 (- 8 (- 12 操作数a4))) 操作数2)

(/ (+ 10 4 (- 8 (- 12 (+ 6 操作数a5)))) 操作数2)

(/ (+ 10 4 (- 8 (- 12 (+ 6 (/ 4 5))))) 操作数2)

(/ (+ 10 4 (- 8 (- 12 (+ 6 (/ 4 5))))) (* 3 操作数b2 操作数b3))

(/ (+ 10 4 (- 8 (- 12 (+ 6 (/ 4 5))))) (* 3 (- 6 2) (- 12 7)))

组合-抽象-构造示例：基于运算组合式的构造
(8)你能表达与构造程序吗?

ICES 战德臣 教授

◆问题2：请定义一个过程，求某一数值的立方 a^3

◆问题3：进一步以问题2定义的过程，再定义一个过程，求某两个数值的立方和。a^3+b^3 进一步求 5^3+8^3 ，并模拟给出计算过程。

```
(define (新运算符 操作数a1 操作数a2 …) ( 运算组合式P ) )
----
(define (cube x) ( 运算组合式P ))
(define (cube x) (* x x x))
----
(cube a)
----
(define (sumofcube x y) ( 运算组合式P ))
(define (sumofcube x y) (+ (cube x) (cube y)))
----
(sumofcube a b)
(sumofcube 5 8)
```

重复性无限性构造的思维--递归与迭代?

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

重复性无限性构造的思维--递归与迭代?
(1)递归的视觉形式

战德臣 教授

自相似性事物的无限重复性构造

重复性无限性构造的思维--递归与迭代?
(2)为什么需要递归?

战德臣 教授

(* … (* (* (* 1 1) 2) 3)…n)

怎样在表达中既去掉省略号，而又能表达近乎无限的内容?

(运算符1 操作数1 操作数2)

(运算符2 操作数a 操作数b)

(运算符1 (运算符2 操作数a 操作数b) 操作数2)

(/ (+ 205 (/ (+ 15 3) (- 90 (× 8 8)))) (- 200 (× 10 5)))

(/ (+ 205 (/ (+ 15 3 - 90 (× 8 8)))) (- 200 (× 10 5)))

这个式子是有问题的哟

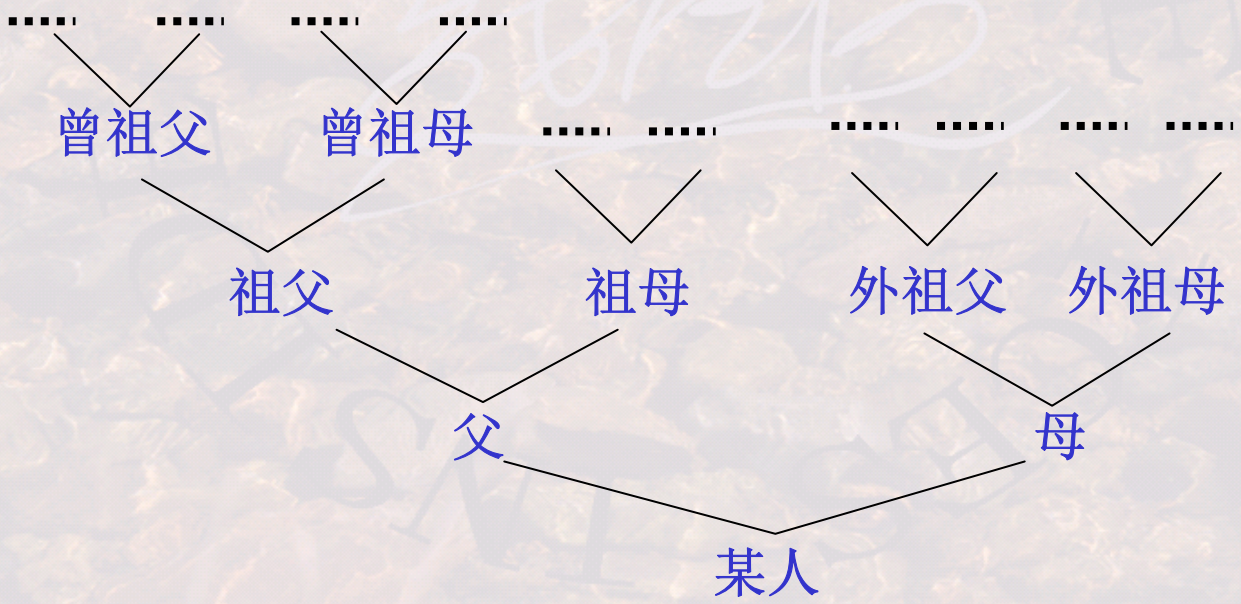
怎样表达运算组合式的层层构造--构造规则?

怎样判断一个运算组合式是否符合构造规则?

重复性无限性构造的思维--递归与迭代?
(2)为什么需要递归?

战德臣 教授

怎样定义你的祖先?



重复性无限性构造的思维--递归与迭代?
(3)数学上的递推式?

战德臣 教授

◆一个数列的第n项a_n与该数列的其他一项或多项之间存在某种对应关系，被表达为一种公式，称为递推式

等差数列递推公式

$a_0=5$

$a_n=a_{n-1}+3$　当n>=1时

- 第1项(或前K项)的值是已知的—递推基础;
- 由第n项或前n项计算第n+1项—递推规则/递推步骤;
- 由前向后，可依次计算每一项

等差数列的产生

$a_0=5$

$a_1=a_0+3 = 8$

$a_2=a_1+3 = 11$

$a_3=a_2+3 = 14$

$a_4=a_3+3 = 17$

… …

重复性无限性构造的思维--递归与迭代?

(4)类比递推式进行定义: 祖先

战德臣 教授

(1)某人的双亲(父母)是他的祖先（递归基础）

--- h(1)直接给出

(2)某人**祖先**的双亲(父母)同样是某人的**祖先**（递归步骤）

--- h(n+1) = g(h(n) , n)

--- n: 第几代； h(n)是第几代祖先； g是h(n+1)与h(n)和n的关系

重复性无限性构造的思维--递归与迭代?

(4)类比递推式进行定义 :运算组合式

战德臣 教授

(1)单一数值是运算组合式

--- h(0)直接给出，递归基础

(2)如果X, Y是运算组合式，则(+ X Y), (* X Y), (- X Y), (/ X Y)也是运算组合式

--- h(n+1) = g(h(n) , n)，即递归步骤

--- n: 第几层；h(n)是第几层表达式；g是(运算符 操作数1 操作数2)，指出h(n+1)与h(n)和n的关系

(/ (+ 10 4 (- 8 (- 12 (+ 6 (/ 4 5))))) (- 15 8))

是运算组合式吗?

说明：这里不允许连加、连减情况发生

h(n+1)	(/ 操作数1 操作数2)
h(n)	(/ (+ 10 4 操作数a2) 操作数2)
h(n-1)	(/ (+ 10 4 (- 8 操作数a3)) 操作数2)
⋮	(/ (+ 10 4 (- 8 (- 12 操作数a4))) 操作数2)
h(1)	(/ (+ 10 4 (- 8 (- 12 (+ 6 操作数a5)))) 操作数2)
h(0)	(/ (+ 10 4 (- 8 (- 12 (+ 6 (/ 4 5))))) 操作数2)

---

h(0)	(/ 4 5)
h(1)	(+ 6 (/ 4 5))
h(2)	(- 12 (+ 6 (/ 4 5)))
⋮	(- 8 (- 12 (+ 6 (/ 4 5))))
h(n)	(+ 10 4 (- 8 (- 12 (+ 6 (/ 4 5)))))
h(n+1)	(/ (+ 10 4 (- 8 (- 12 (+ 6 (/ 4 5))))) 操作数2)

战德臣 教授

递归是表达相似性对象及动作的无限性重复性构造的方法。

■递归基础：定义、构造和计算的起点，直接给出。即h(0)。

■递归步骤：由前n项或第n项定义第n+1项；由低阶f(k)且k<n，来构造高阶f(n+1)；即: h(n+1) = g (h(n), n)，或者 h(n+1) = g(h(k), n, k)形式，g是需要明确给出的，以说明h(n+1)怎样由h(k), k和n构造出来。

用递归定义

用递归构造

递归计算/执行

◆递归是一种关于抽象的表达方法---用递归定义无限的相似事物

◆递归是一种算法或程序的构造技术---自身调用自身，高阶调用低阶，构造无限的计算步骤

◆递归是一种典型的计算/执行过程---由后向前代入，直至代入到递归基础，再由递归基础向后计算直至计算出最终结果，即由前向后计算

重复性无限性构造的思维--递归与迭代?

(6)两种不同的递归函数--递归与递推(迭代)的差别?

战德臣 教授

Fibonacci数列： 无穷数列1，1，2，3，5，8，13，21，34，55，……

$$F(n)=\begin{cases}1 & n=0\\ 1 & n=1\\ F(n-1)+F(n-2) & n>1\end{cases}$$

递归定义

F(0)=1;

F(1)=1;

F(2)=F(1)+F(0)=2;

F(3)=F(2)+F(1)= 3;

F(4)=F(3)+F(2)= 3+2=5;… …

递推计算/迭代计算/迭代执行

定义是递归的, 但执行可以是递归的也可是迭代/递推的

重复性无限性构造的思维--递归与迭代?
(6)两种不同的递归函数--递归与递推(迭代)的差别?

战德臣 教授

阿克曼递归函数:

$$A(m,n)=\begin{cases} n+1 & \text{若} m=0 \\ A((m-1),1) & \text{若} n=0 \\ A(m-1,A(m,n-1)) & \text{若} m,n>0 \end{cases}$$

递归定义

函数本身是递归的，
函数的变量也是递归的

A(1,2) = A(0,A(1,1)) --A(1,2)按m,n>0公式代入
= A(0, A(0,A(1,0))) --A(1,1)按m,n>0公式代入
= A(0, A(0,A(0,1))) --A(1,0)按n=0公式代入
=A(0,A(0,2)) --A(0,1)按m=0公式代入, A(0,1)=2
=A(0,3)=4。 --A(0,2)按m=0公式代入, A(0,2)=3; A(0,3)按m=0代入。

A(1,3) =A(0, A(1,2))
=A(0. <代入前式A(1,2)的计算过程>)
=A(0, 4) = 4+1=5。

…

A(1,n) =A(0, A(1,n-1))
=A(0. <代入前式…计算过程>)
=A(0,n+1)=n+2。

递归计算/递归执行

由后向前代入，再由前向后计算

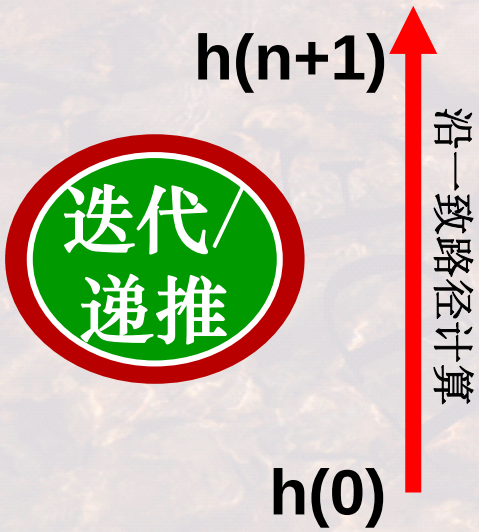
重复性无限性构造的思维--递归与迭代?

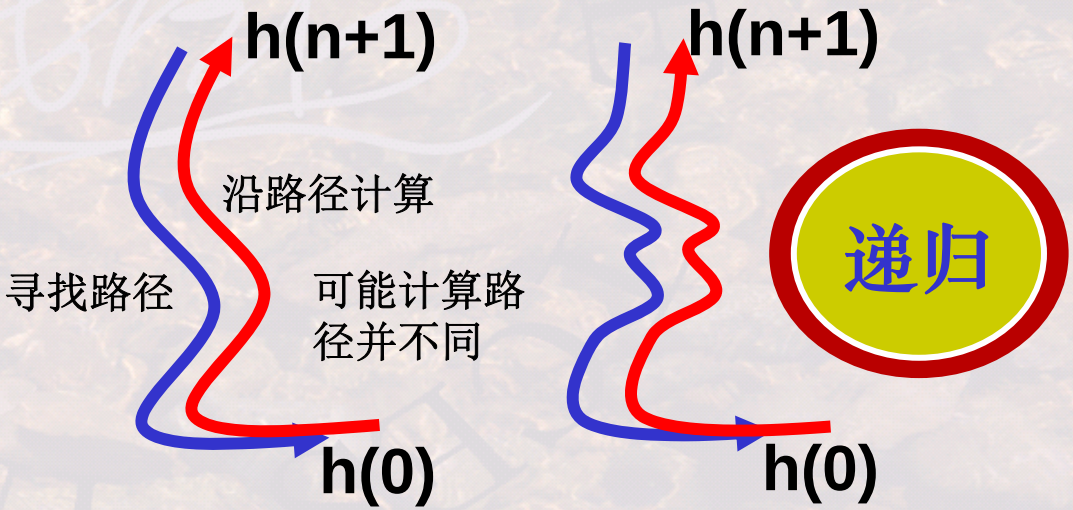
(6)两种不同的递归函数--递归与递推(迭代)的差别?

战德臣 教授

$$h(n)=\begin{cases}1 & n=0\\ 1 & n=1\\ h(n-1)+h(n-2) & n>1\end{cases}$$

$$h(m,n)=\begin{cases}n+1 & 若m=0\\ h((m-1),1) & 若n=0\\ h(m-1,h(m,n-1)) & 若m,n>0\end{cases}$$





在前次结果基础上进行计算
可采用循环结构实现

只能由函数形式实现

重复性无限性构造的思维--递归与迭代?
(6)两种不同的递归函数--递归与递推(迭代)的差别?

战德臣 教授

阿克曼递归函数:

$$A(m,n)=\begin{cases} n+1 & \text{若} m=0 \\ A((m-1),1) & \text{若} n=0 \\ A(m-1,A(m,n-1)) & \text{若} m,n>0 \end{cases}$$

递归定义

函数本身是递归的，
函数的变量也是递归的

A(1,n) =A(0, A(1,n-1))=A(0. <...代入前式计算过程>)=A(0,n+1)=n+2。

A(2,1)= A(1， A(2， 0)) --A(2,1)按m,n>0公式代入
= A(1， A(1， 1)) --A(2,0)按n=0公式代入
=A(1， A(0， A(1， 0))) --A(1,1)按m,n>0公式代入
= A(1， A(0， A(0， 1))) --A(1,0)按n=0公式代入
= A(1， A(0， 2)) --A(0,1)按m=0公式代入
= A(1， 3) --A(0,2)按m=0公式代入
=A(0， A(1， 2)) --A(1,3)按m,n>0公式代入
= A(0， A(0， A(1， 1))) --A(1,2)按m,n>0公式代入
= A(0， A(0， A(0， A(1， 0)))) --A(1,1)按m,n>0公式代入
= A(0， A(0， A(0， A(0,1))) --A(1,0)按n=0公式代入
= A(0， A(0， A(0， 2))) --A(0,1)按m=0公式代入
= A(0， A(0， 3)) = A(0， 4) = 5。 --依次按m=0公式代入

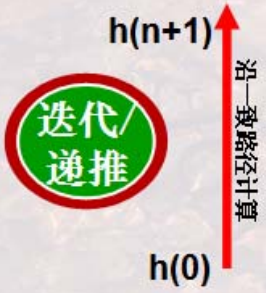
递归计算/递归执行

由后向前代入，
再由前向后计算

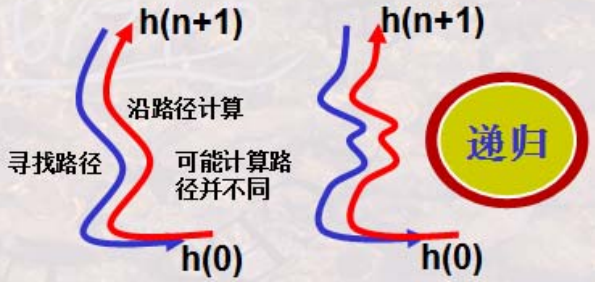
重复性无限性构造的思维--递归与迭代?
(7)小结

战德臣 教授

这里没有多少数学...但有的是很重要的很深刻的计算思维... ...



在前次结果基础上进行计算
可采用循环结构实现



只能由函数形式实现

组合-抽象-构造示例?
----基于典型计算机语言的迭代构造

战德臣

哈尔滨工业大学　教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

组合-抽象-构造示例--基于典型计算机语言的迭代构造
(1)初步认识计算机语言程序

战德臣 教授

变量　　常量

保留字

```
K = 0;
For  I =1 to 100 Step 1
{    If  I<=50 && I > 30
     { K =  K+I; }
}
```

语句

表达式

“保留字”为编译器识别和处理语句而确定的一些特定符号

组合-抽象-构造示例--基于典型计算机语言的迭代构造
(1)初步认识计算机语言程序

ICES 战德臣 教授

分支结构

```
If 条件表达式
{  (条件为真时运行的)程序语句序列1 }
Else
{  (条件为假时运行的)程序语句序列2 }
```

循环结构

```
For 计数器变量 = 起始值 To  结束值 [Step 增量表达式]
{ 循环体的程序语句序列 }
```

```
K = 0;
For  I =1 to 100 Step 1
{    If  I<=50 && I > 30
     { K =  K+I; }
}
```

组合-抽象-构造示例--基于典型计算机语言的迭代构造
(2)考虑一个计算多项式值的问题

战德臣 教授

给定n, 求n^2, 如何计算? ---只用加减法来进行乘方运算

n	$K_n=n^2$	一阶差分 $\alpha_n = K_n-K_{n-1}$	二阶差分 $\beta_n = \alpha_n - \alpha_{n-1}$
0	0		
1	1	1	
2	4	3	2
3	9	5	2
4	16	7	2
5	25	9	2

- 仅需能够进行加法运算和减法运算；
- 其他运算可通过组合加法与减法运算来实现；
- 例如: 乘方运算?

$$K_{n+1} = K_n + \alpha_n + \beta_n$$
$$\alpha_n = K_n - K_{n-1}$$
$$\beta_n = \alpha_n - \alpha_{n-1}$$

组合-抽象-构造示例--基于典型计算机语言的迭代构造

(2)考虑一个计算多项式值的问题

战德臣 教授

n=x	$K_n = x^2+2x+3$	一阶差分 $\alpha_n = K_n - K_{n-1}$	二阶差分 $\beta_n = \alpha_n - \alpha_{n-1}$
0	3		
1	6	3	
2	11	5	2
3	18	7	2
4	27	9	2
5	38	11	2
6	51	13	2

- 多项式运算?
- 初始值不一样，计算的多项式也是不一样的；

$$x^2 + 2x + 3$$

组合-抽象-构造示例--基于典型计算机语言的迭代构造
(3)试构造一个程序

战德臣 教授

```
/*类C语言表达的计算规则—程序
Main()
{
   int k, n, square[ ], alpha[ ], beta[ ];
   input k;
   square[0]=0;
   square[1]=1;
   square[2]=4;
   alpha[1] = 1;
   for n=2 to k-1
   {
        alpha[n] = square[n] - square[n-1];
        beta[n] = alpha[n] - alpha[n-1];
        square[n+1] = square[n] + alpha[n] + beta[n];
   }
   output square[k];
}
```

输入不同的初始值便可计算不同的一元二次多项式的值

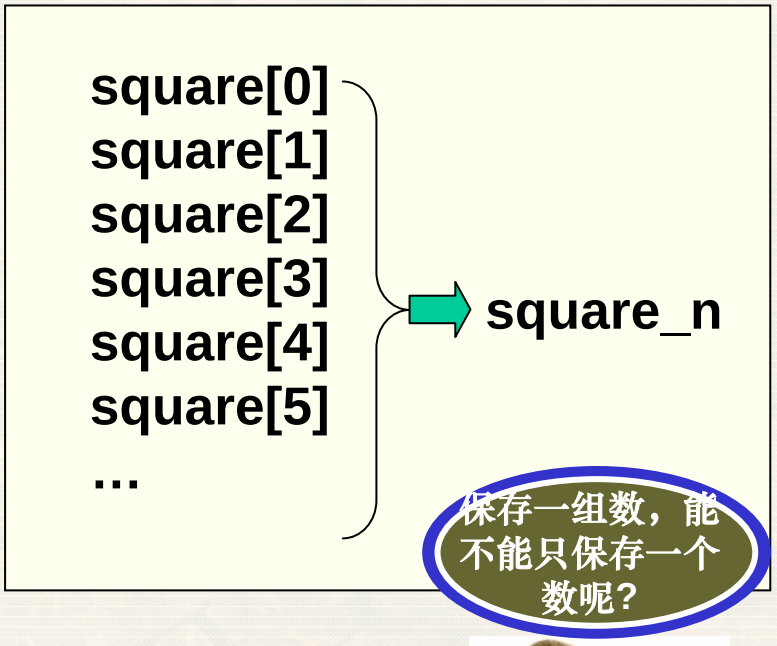
n	n^2	一阶差分 $\alpha_n = n^2-(n-1)^2$	二阶差分 $\beta_n = \alpha_n - \alpha_{n-1}$
0	0		
1	1	1	
2	4	3	2
3	9	5	2
4	16	7	2
5	25	9	2

$$(n+1)^2 = n^2 + \alpha_n + \beta_n$$

组合-抽象-构造示例--基于典型计算机语言的迭代构造
(3)试构造一个程序

战德臣 教授

```
/*类C语言表达的计算规则—程序
Main()
{
   int k, n, square[ ], alpha[ ], beta[ ];
   input k;
   square[0]=0;
   square[1]=1;
   square[2]=4;
   alpha[1] = 1;
   for n=2 to k-1
   {
        alpha[n] = square[n] - square[n-1];
        beta[n] = alpha[n] - alpha[n-1];
        square[n+1] = square[n] + alpha[n] + beta[n];
   }
   output square[k];
}
```



组合-抽象-构造示例--基于典型计算机语言的迭代构造
(3)试构造一个程序

战德臣 教授

这些 都只是一个变量，虽然名字长了一点


```
/*类C语言表达的计算规则—程序
Main()
{
  int k, n, square_nminus1,square_n, alpha_nminus1, alpha_n, beta_n;
  input k;
  square_nminus1=1; square_n=4; alpha_nminus1=1;
  for n=2 to k-1
  {
      alpha_n = square_n – square_nminus1;
      beta_n = alpha_n – alpha_nminus1;
      square_nplus1 = square_n + alpha_n + beta_n;
      square_nminus1 = square_n;
      square_n = square_nplus1;
      alpha_nminus1 = alpha_n;
  }
  output square_n;
}
```

重复

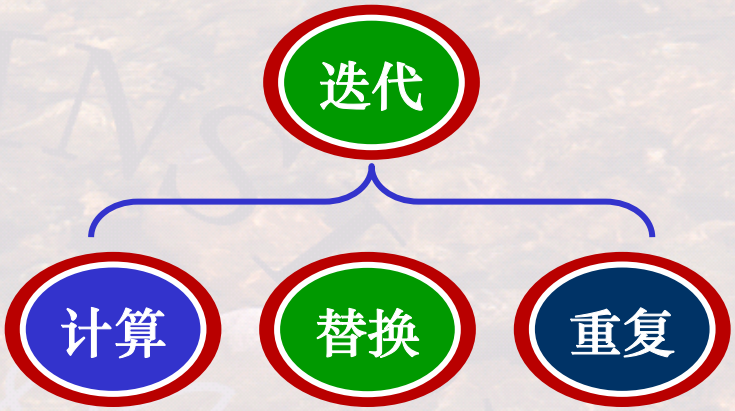
计算

替换

组合-抽象-构造示例--基于典型计算机语言的迭代构造
(4)试构造一个程序—程序的执行过程模拟

战德臣 教授

```
/*类C语言表达的计算规则—程序
Main()
{
  int k, n, square_nminus1,square_n, alpha_nminus1, alpha_n, beta_n;
  input k;
  square_nminus1=1; square_n=4; alpha_nminus1=1;
  for n=2 to k-1
  {
     alpha_n = square_n – square_nminus1;
     beta_n = alpha_n – alpha_nminus1;
     square_nplus1 = square_n + alpha_n + beta_n;
     square_nminus1 = square_n;
     square_n = square_nplus1;
     alpha_nminus1 = alpha_n;
  }
  output square_n;
}
```



n	square_nplus1	square_n	square_nminus1	alpha_n	alpha_nminus1	beta_n
2	9	4	1	3	1	2
3	16	9	4	5	3	2
4	25	16	9	7	5	2
5	36	25	16	9	7	2
		36	25		9	

组合-抽象-构造示例?
----基于典型计算机语言的递归构造

战德臣

哈尔滨工业大学　教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

组合-抽象-构造示例--基于典型计算机语言的递归构造
(1)你知道函数是一种抽象吗?

战德臣 教授

函数是一种抽象，用一个名字代表一个程序段落

函数的定义

形式参数

```
int Sum(int m, int n)
    //函数名, 设置形式参数
{
    //函数体,函数的语句序列
    S = m + n;
    Return S;
}
```

函数体，实现函数功能的程序语句序列
以形式参数作为需要处理的对象。
当被调用时，用实际参数替换相应的形式参数进行程序执行。

与“运算组合式”的对比

(define (新运算符组合式) (运算组合式P))

(define (sum m n) (+ m n))

函数体

形式参数

新运算组合式，即“函数”

组合-抽象-构造示例--基于典型计算机语言的递归构造

(1)你知道函数是一种抽象吗?

战德臣 教授

主函数

```
Main()
{
  Printf("请输入被加数");
  Scanf("%d",&x);
  Printf("请输入加数");
  Scanf("%d",&y);
  z = Sum(x,y);
  Printf("求和结果为%d", z);
}
```

函数调用,
给定实际参数

函数的使用

实际参数

```
OtherFunc()
{
  …
  z = Sum(6,10);
  …
  …
  y = Sum(20,30);
}
```

函数的使用

实际参数

与“运算组合式”的对比

```
(sum  5  4)
( +  ( sum  20  30)  (sum  15  20))
```

组合-抽象-构造示例--基于典型计算机语言的递归构造

(2)试着构造一个程序?

战德臣 教授

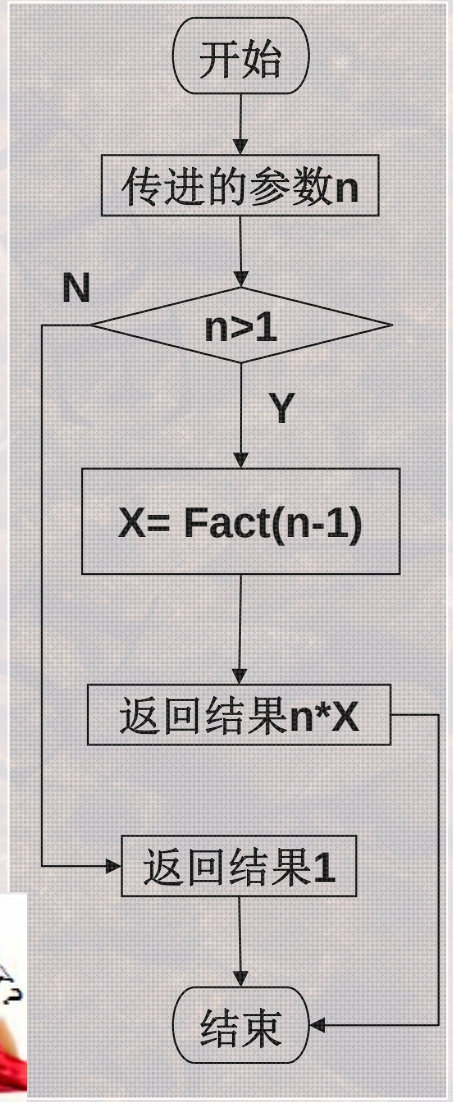
具有无限的自相似性步骤的表达，自身调用自身，高阶调用递阶

示例：求n!的算法或程序 *--用递归方法构造*

$$n! = \begin{cases} 1 & \text{当} n \le 1 \text{时} \\ n \times (n-1)! & \text{当} n > 1 \text{时} \end{cases}$$

```
long int Fact(int n)
{    long int x;
     If (n > 1)
     { x = Fact(n-1);
  /*递归调用*/
        return n*x;   }
     else return 1;
  /*递归基础*/
}
```

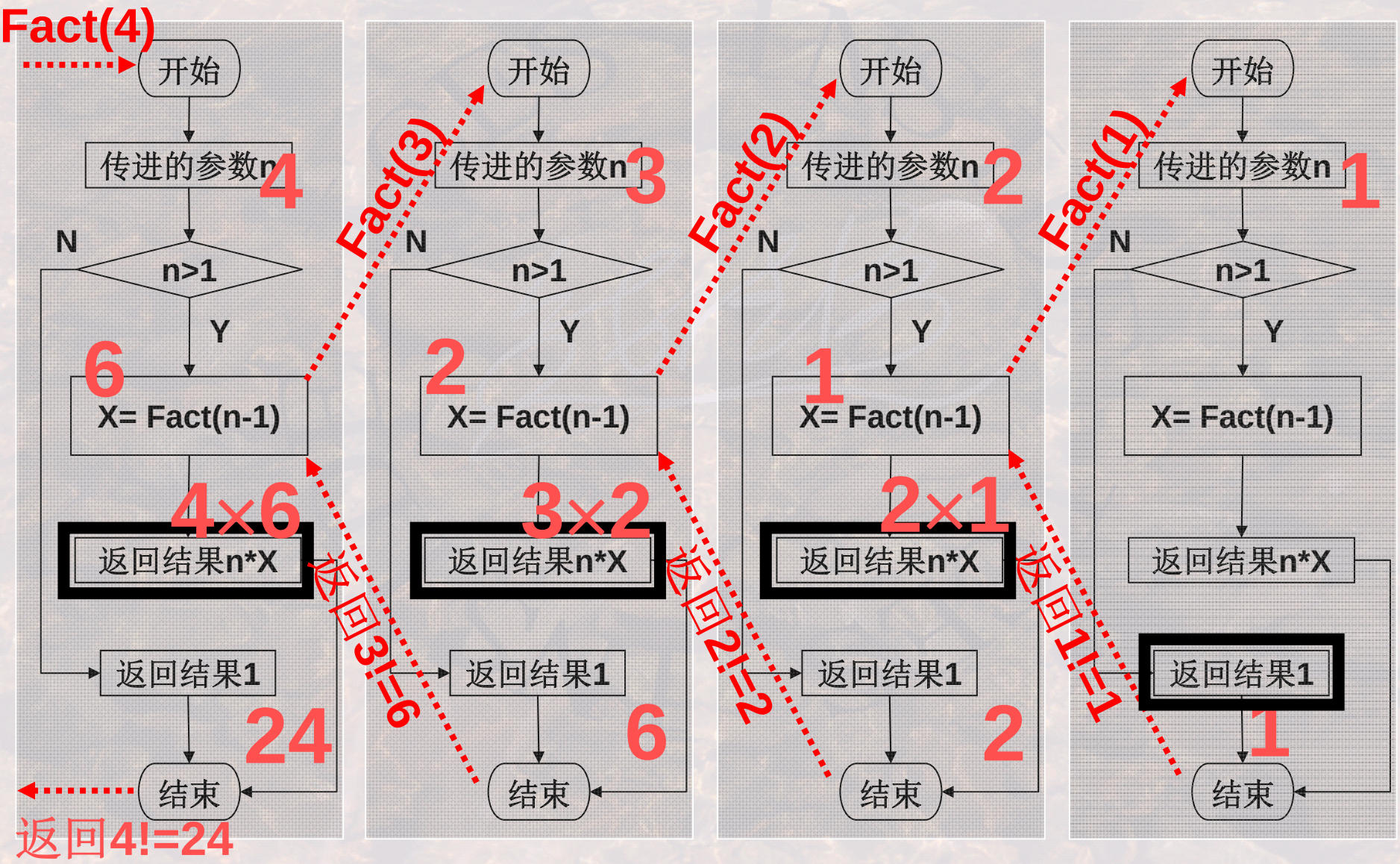
注意这里的n是形式参数，实际执行时将被实际参数所替换… …



组合-抽象-构造示例--基于典型计算机语言的递归构造

(3)递归程序的执行过程?

战德臣 教授



组合-抽象-构造示例--基于典型计算机语言的递归构造

(4)试着用迭代方法构造程序?

战德臣 教授

示例：求n!的算法或程序 *--用迭代方法构造*

$$n!=\begin{cases}1 & \text{当} n\le 1\text{时}\\ 1\times 2\times\ldots(n-1)\times n & \text{当} n>1\text{时}\end{cases}$$

```
long int Fact(int n)
{   int  counter；
    long product=1;
    for  counter=1 to n step 1
    { product = product * counter；  }
  /*迭代*/
return product;
}
```

Fact(5)的执行过程

循环次数	Counter	Product
初始值	-	1
循环第1次	1	1
循环第2次	2	2
循环第3次	3	6
循环第4次	4	24
循环第5次	5	120
退出循环	6	

第3讲 软件与程序思想:
组合-抽象-构造-递归

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent
Computing for Enterprises & Services,
Harbin Institute of Technology

软件与程序思想: 组合-抽象-构造-递归

•回顾本讲学习了什么