编译原理

第二章 高级语言及其语法描述

第二章 高级语言及其语法描述

- 程序语言的定义
- 高级语言的一般特性
- 程序语言的语法描述

第二章 高级语言及其语法描述

- 常用的高级语言
  - FORTRAN 数值计算
  - COBOL 事务处理
  - PASCAL 结构程序设计
  - ADA 大型程序、嵌入式实时系
  - LISP 函数式程序设计语言
  - PROLOG 逻辑程序设计
  - ALGOL 算法语言
  - C/C++ 系统程序设计
  - Java Internet 程序设计

A language that doesn't affect the way you think about programming, is not worth knowing.

Alan J. Perlis

ACM 图灵奖


Alan J. Perlis


Edsger W. Dijkstra

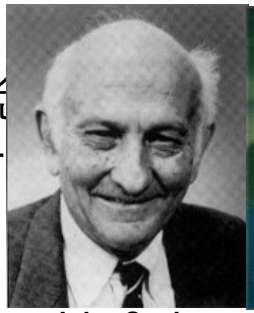

John W. Backus


Kenneth E. Iverson


Niklaus Wirth

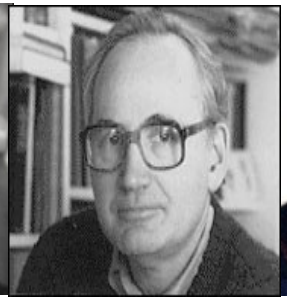
- 程序设计语言
  - Alan J. Perlis (1966) -- ALGOL
  - Edsger Wybe Dijkstra (1972) -- ALGOL
  - Michael O. Rabin & Dana S. Scott (1976) -- 非确定性自动机
  - John W. Backus (1977) -- FORTRAN
  - Kenneth Eugene Iverson (1979) -- APL 程序语言
  - Niklaus Wirth (1984) -- PASCAL
  - John Cocke (1987) -- RISC & 编译优化
  - O. 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
  - .

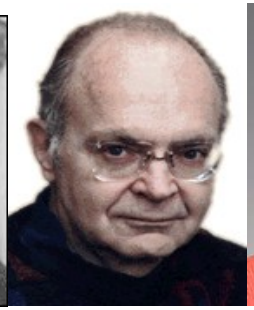

O. Dahl


John Cocke


K.Nygaard


Michael O. Rabin


Dana S. Scott


Donald E. Knuth


Barbara Liskov


Frances E. Allen


Peter Naur


Alan Kay

第二章 高级语言及其语法描述

- 与机器语言或汇编语言比较，高级语言的优点
  - 较接近于数学语言和工程语言，比较直观、自然和易于理解
  - 便于验证其正确性，易于改错
  - 编写效率高
  - 易于移植

第二章 高级语言及其语法描述

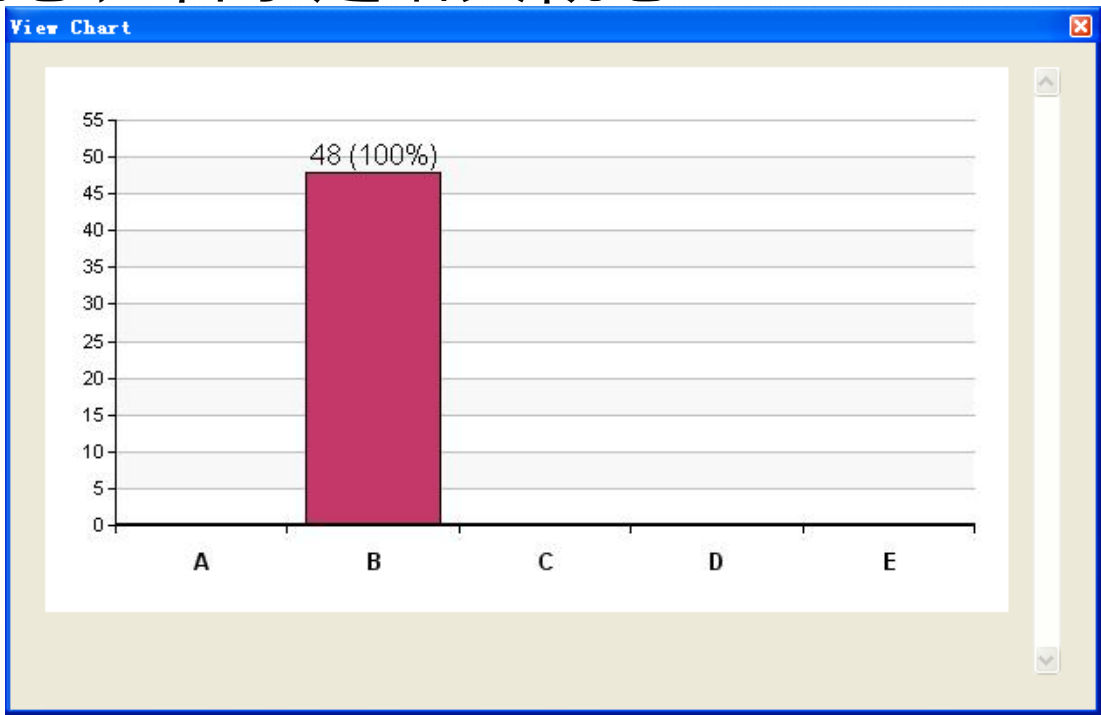
- 程序语言的定义
- 高级语言的一般特性
- 程序语言的语法描述

测试

下面哪种说法正确？（　）

A.　标识符是语义概念，名字是语法概念

B.　标识符是语法概念，名字是语义概念



iclicker

2.1 程序语言的定义

- 程序语言由两方面定义
  - 语法
  - 语义
  - 语用

语法

- 程序本质上是一定字符集上的字符串
- 语法：一组规则，用它可以形成和产生一个合式 (well-formed) 的程序

语法

- 词法规则：单词符号的形成规则
  - 单词符号是语言中具有独立意义的最基本结构
  - 一般包括：常数、标识符、基本字、算符、界符等
  - 描述工具：有限自动机
- 语法规则：语法单位的形成规则
  - 语法单位通常包括：表达式、语句、分程序、过程、函数、程序等；
  - 描述工具：上下文无关文法

语法

- E→i

  E→E+E

  E→E*E

  E→(E)

- 语法规则和词法规则定义了程序的形式结构
- 定义语法单位的意义属于语义问题

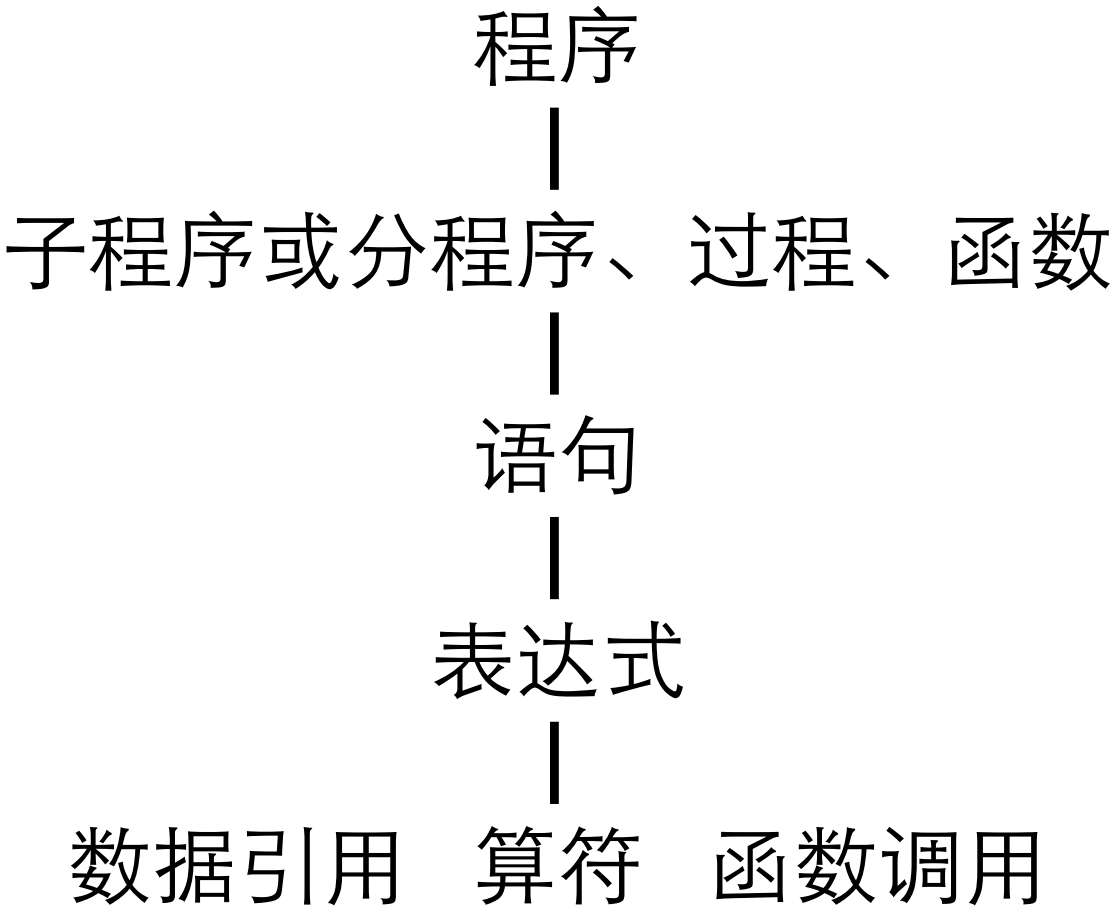
语义

- 语义
  - 一组规则，用它可以定义一个程序的意义
- 描述方法
  - 自然语言描述
    - 隐藏错误、二义性和不完整性
  - 形式描述
    - 操作语义 (PL/1)
    - 指称语义 (ADA)
    - 代数语义 (PASCAL)

程序语言的基本功能和层次结构

- 程序语言的基本功能
  - 描述数据和对数据的运算
- 所谓程序，本质上说是描述一定数据的处理过程

程序的层次结构



程序语言成分的逻辑和实现意义

- 抽象的逻辑的意义
  - 数学意义
- 计算机实现的意义
  - 具体实现

计算思维与数学思维的不同

第二章 高级语言及其语法描述

- 程序语言的定义
- 高级语言的一般特性
- 程序语言的语法描述

2.2 高级语言的一般特性

- 高级语言的分类
- 程序结构
- 数据结构与操作
- 语句与控制结构

2.2 高级语言的一般特性

- 高级语言的分类
- 程序结构
- 数据结构与操作
- 语句与控制结构

2.2 高级语言的一般特性

- 高级语言的分类
  - 强制式语言 (Imperative Languge)/ 过程式语言
    - 命令驱动，面向语句
    - FORTRAN 、 C 、 Pascal ， Ada
  - 应用式语言 (Applicative Language)
    - 注重程序所表示的功能，而不是一个语句接一个语句地执行
    - LISP 、 ML

2.2 高级语言的一般特性

- 高级语言的分类
  - 强制式语言 (Imperative Languge)/ 过程式语言
  - 应用式语言 (Applicative Language)
  - 基于规则的语言 (Rule-based Language)
    - 检查一定的条件，当它满足值，则执行适当的动作
    - Prolog

2.2 高级语言的一般特性

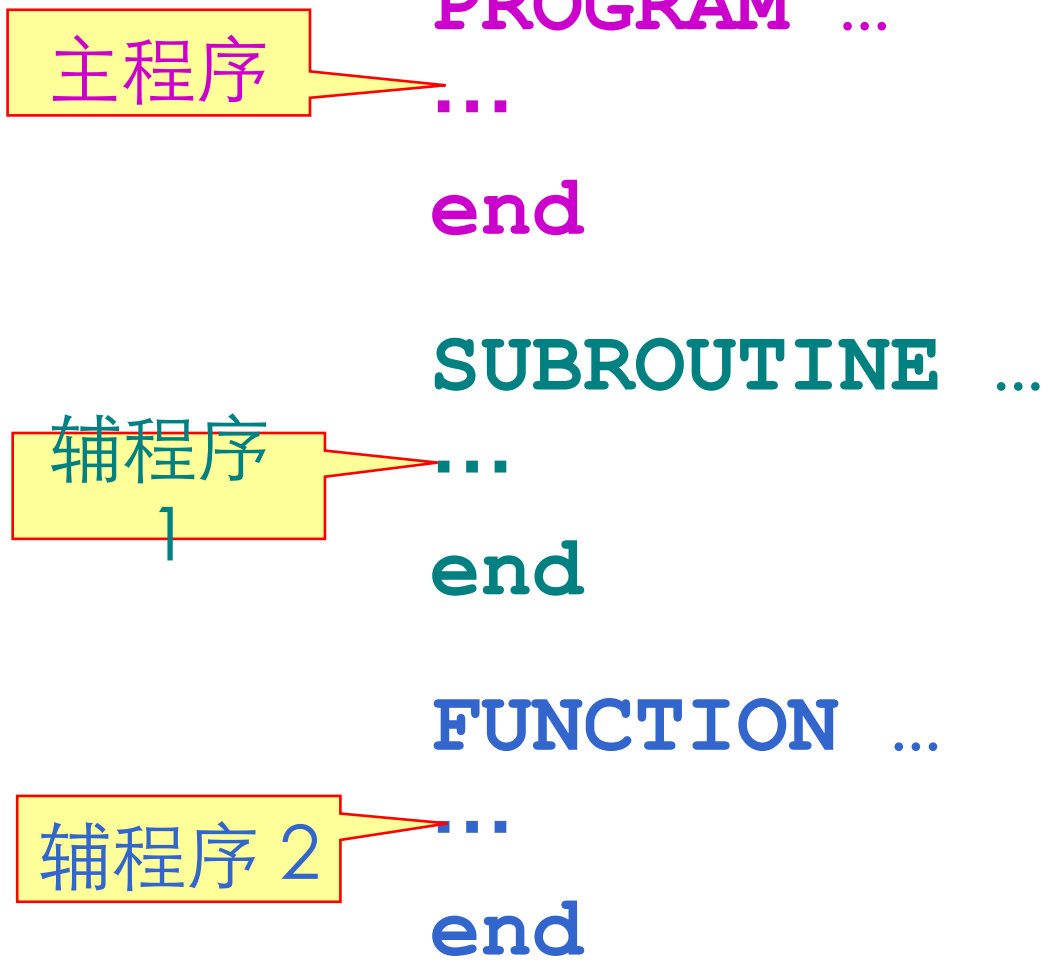
- 高级语言的分类
  - 强制式语言 (Imperative Languge)/ 过程式语言
  - 应用式语言 (Applicative Language)
  - 基于规则的语言 (Rule-based Language)
  - 面向对象语言 (Object-Oriented Language)
    - 封装性、继承性和多态性
    - Smalltalk， C++， Java

2.2 高级语言的一般特性

- 高级语言的分类
- 程序结构
- 数据结构与操作
- 语句与控制结构

2.2 高级语言的一般特性

- 程序结构
  - FORTRAN



- 一个程序由一个主程序段和若干辅程序段组成
- 辅程序段可以是子程序、函数段或数据块
- 每个程序段有一系列的说明语句和执行语句组成，各段可以独立编译
- 模块结构，没有嵌套和递归
- 各程序段中的名字相互独立，同一个标识符在不同的程序段中代表不同的名字

程序结构

- PASCAL
  - PASCAL 程序本身可以看成是一个操作系统所调用的过程，过程可以嵌套和递归
  - 一个 PASCAL 过程

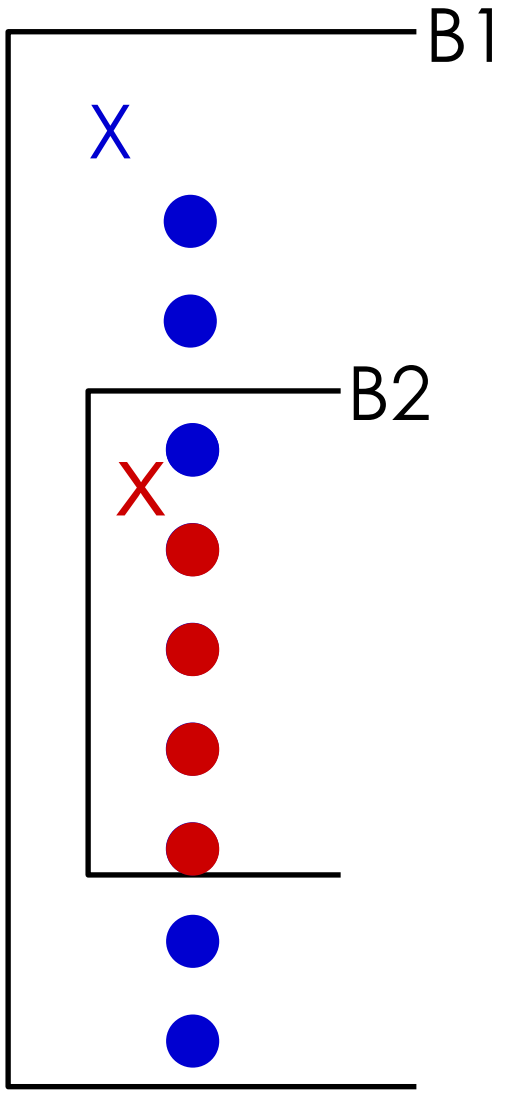
```
过程头;
        说明段（由一系列的说明语句组成）;
begin
        执行体（由一系列的执行语句组成）;
end
```

作用域

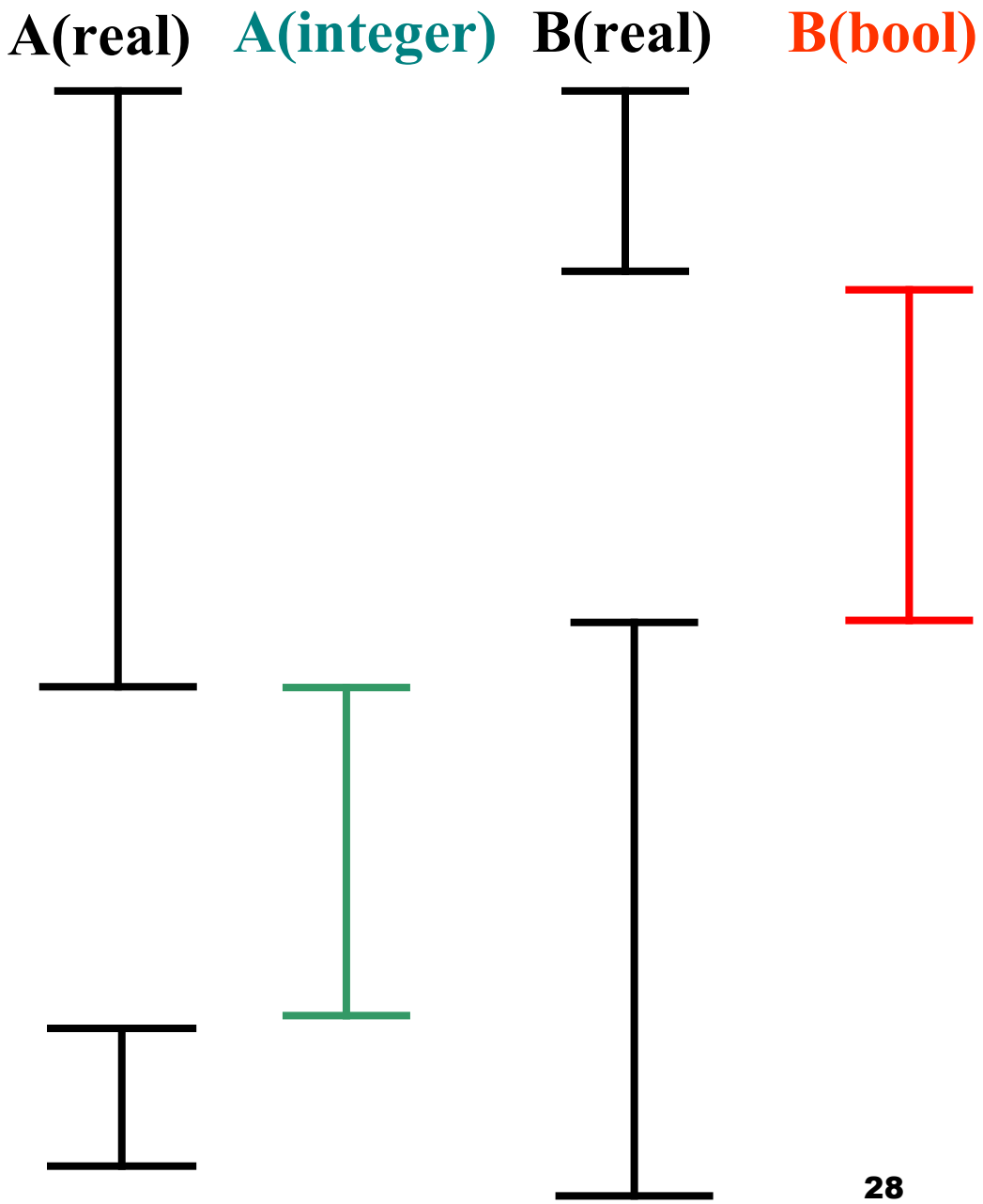
- 作用域
  - 一个名字能被使用的区域范围称作这个名字的作用域
- 允许同一个标识符在不同过程中代表不同的名字
- 名字作用域规则——"最近嵌套原则"

最近嵌套原则

- 一个在子程序 B1 中说明的名字 X 只在 B1 中有效（局部于 B1）
- 如果 B2 是 B1 的一个内层子程序且 B2 中对标识符 X 没有新的说明，则原来的名字 X 在 B2 中仍然有效
- 如果 B2 对 X 重新作了说明，那么， B2 对 X 的任何引用都是指重新说明过的这个 X



```
program main
   var A, B:real;
   …
   procedure P1
      var B:boolean;
      …
    begin
      …
    end
   procedure P2
      var A:integer;
      …
   begin
      …
   end
begin
   …
end
```



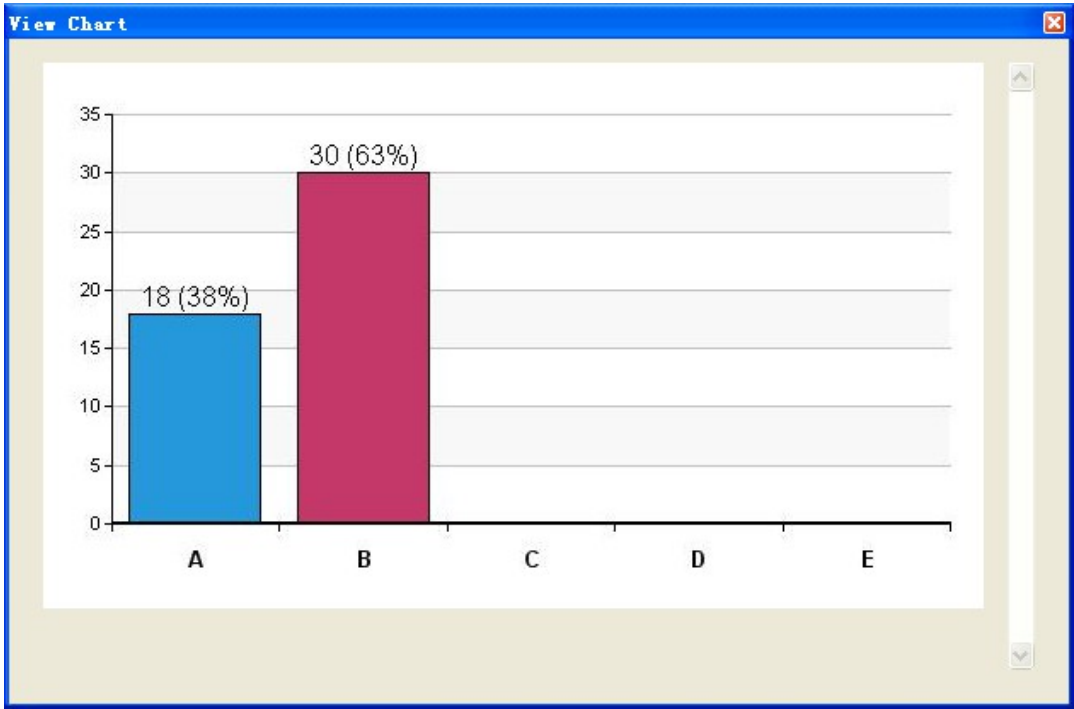
```
program main
   var A, B:real;
   …
   procedure P1
      var B:boolean;
      …
   begin
      …
   end
   procedure P2
      var A:integer;
      …
   begin
      …
   end
begin
   …
end
```

测试

P2 的代码能够调用 P1 吗?

A. 可以

B. 不可以



程序结构

- PASCAL 提供了丰富的数据类型和运算方式，它允许用户动态地申请和退还存贮空间

程序结构

- ADA
  - 程序包 (package)
    - 把数据和操作代码封装在一起，支持数据抽象
  - 一个程序包分为两部分
    - 可见的规范说明部分，它定义了程序包外面可以访问的对象
    - 程序包体，它实际定义程序包的实现细节

```
package STACKS is
   type ELEM is private;
   type STACK is limited private;
   procedure push (S: in out STACK; E: in ELEM);
   procedure pop (S: in out STACK; E: out ELEM);
  …
end STACK;
```

规范说明

```
package body STACKS is
   procedure push(S: in out STACK; E: in ELEM);
   begin
  …… 实现细节
   end push;
   procedure pop (S: in out STACK; E: out ELEM);
   begin
  …… 实现细节
   end pop;
end;
```

程序包体

程序结构

- JAVA
  - 面向对象的高级语言
    - 类（Class）
    - 继承 (Inheritance)
    - 多态性 (Polymorphism) 和动态绑定 (Dynamic binding)

```
class Car{
   int color_number;
   int door_number;
   int speed;
 …
   push_break ( ) {
 …
   }
   add_oil ( ) {
 …
   }
}

class Trash_Car extends car {
   double amount;
   fill_trash ( ) {
 …
   }
}
```

小结

- 程序语言的定义
  - 语法
  - 语义
  - 程序语言的功能
- 高级语言的一般特性
  - 高级语言的分类
  - 程序结构