2019 级本科

软件工程

Software Engineering

张昕

zhangxin@cust.edu.cn

计算机科学技术学院软件工程系

Detailed design

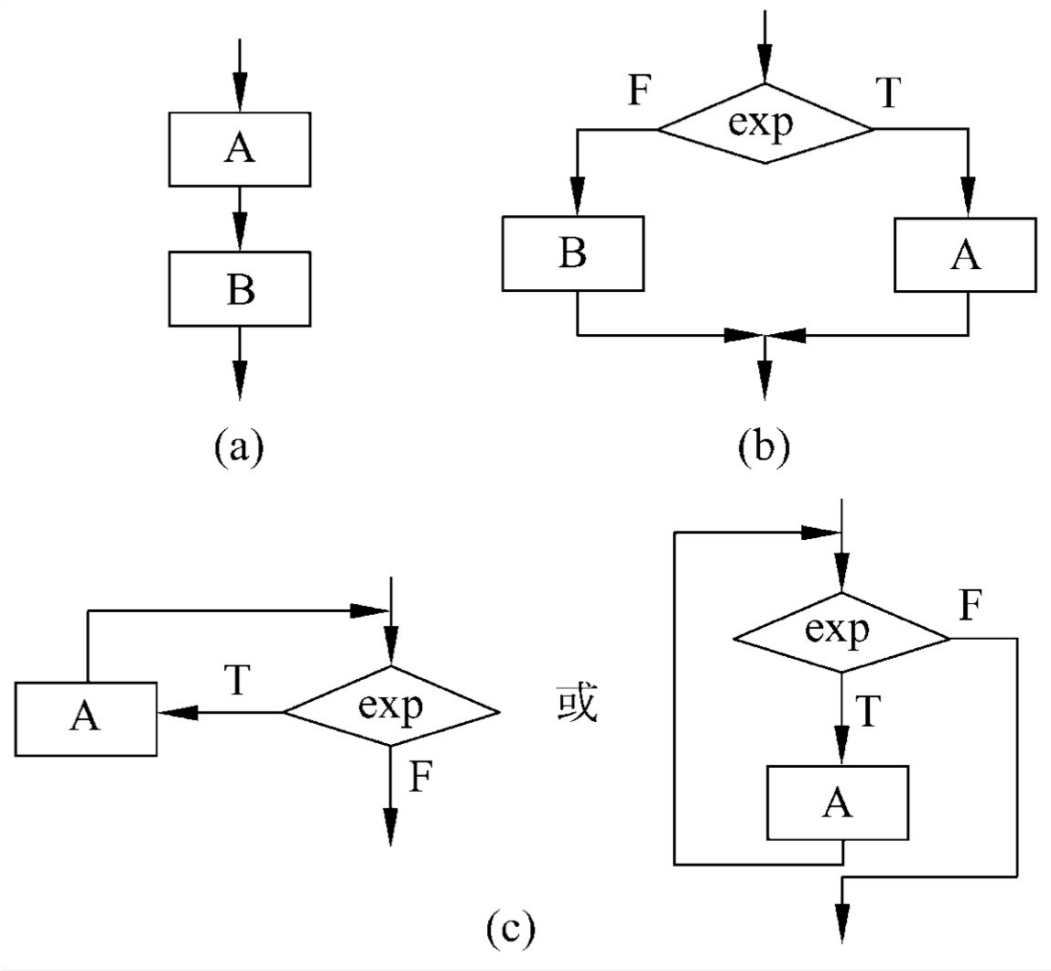
Introduction

- The fundamental goal of the detailed design stage
  - Determine how to implement the required system in detail
    - Get an accurate description of the target system
    - The description can be directly translated into a program at the coding stage
- The result of detailed design basically determines the quality of the final program code
  - The "reader" must be paid attention to when considering the quality of the program code
    - computer
    - human
  - To measure the quality of a program depends not only on whether its logic is correct and whether its performance meets the requirements, but also whether it is easy to read and understand.

Structural programming

- The concept of structural programming was first proposed by E.W.Dijkstra
  - He pointed out at a conference in 1965: "GO TO statements can be eliminated from high-level languages", "The quality of the program is inversely proportional to the number of GO TO statements contained in the program"
  - In 1966, Bohm and Jacopini proved that any single-entry single-export program can be realized with only 3 basic control structures.
- The 3 basic control structures are
  - order
  - choose
  - cycle

Basic control structure



Structural programming

- In theory, there are only two basic control structures
  - The selection structure (also known as IF-THEN-ELSE structure) can be realized with sequence structure and loop structure (also known as DO-WHILE structure)
- In 1968, Dijkstra once again suggested to abolish the *GO TO* statement from all high-level languages, and only use 3 basic control structures to write programs
  - Create a new programming idea, method and style to significantly improve software productivity and reduce software maintenance costs

Case study

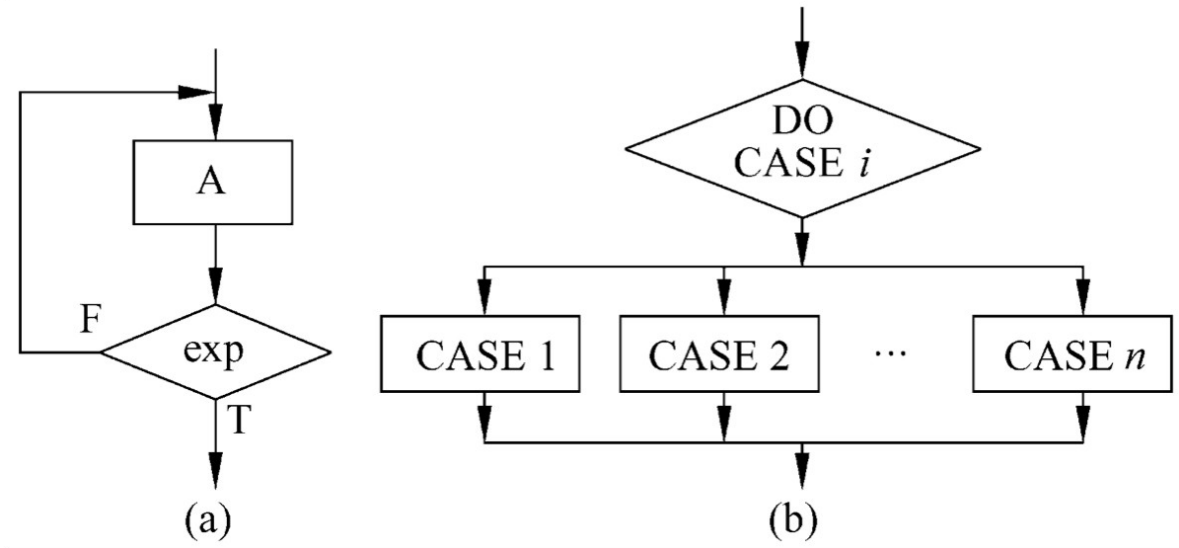
- In 1971, IBM successfully used the structural programming technology in the design of the New York Times Information Database Management System, and then in the design of the NASA Space Laboratory flight simulation system, the structural programming technology was once again a complete success
  - Both systems are quite large
    - The former contains 83,000 lines of high-level language source programs
    - The latter contains 400,000 lines of source code
    - There have been many changes in user requirements during the design process
  - The development work of both systems was completed on time and with high quality.
  - Software productivity has doubled than before, and the structural programming technology has successfully withstood the test of practice
- In 1972, IBM's Mills further proposed that the program should have only one entry and one exit, thus supplementing the rules of structural programming.

Definition of structural programming

- The classic definition of structural programming
  - "If the code blocks of a program are only connected by the three basic control structures of sequence, selection, and loop, and each code block has only one entry and one exit, the program is said to be structured."
- The above classic definition is too narrow
  - Structural programming is essentially not a programming method without GO TO statement, but a programming method that makes the program code easy to read and understand
- Comprehensive definition
  - "Structural programming is a programming method that uses GO TO statements as little as possible. It is best to use GO TO statements only when errors are detected, and forward GO TO statements should always be used."

Definition of structural programming

- Although theoretically speaking, any single-entry single-export program can be realized by only using the above three basic control structures.
  - But for the convenience of actual use, it is often allowed to use DO-UNTIL and DO-CASE two control structures



Definition of structural programming

- Sometimes it is necessary to move out of the loop (or even nested loop) immediately. If the LEAVE (or BREAK) structure is allowed, it will not only be convenient but also increase the efficiency a lot
  - The LEAVE or BREAK structure is essentially a restricted GO TO statement, which is used to transfer to the statement behind the loop structure
- Classic structure programming
  - Only three basic control structures of sequence, IF-THEN-ELSE type branch and DO-WHILE type loop are allowed to be used
- Extended structure programming
  - If in addition to the above three basic control structures, it is also allowed to use DO-CASE type multi-branch structure and DO-UNTIL type cyclic structure
- Modified structure programming
  - If plus the LEAVE (or BREAK) structure is allowed

Human-machine interface design

- Human-machine interface design is an important part of interface design
  - For interactive systems, man-machine interface design is as important as data design, architecture design and process design
  - In individual systems, the design workload of the human-machine interface even accounts for more than half of the total design

- The design quality of the man-machine interface directly affects the user's evaluation of the software product, thereby affecting the competitiveness and lifespan of the software product

Human-machine interface design issues

- 4 common questions
  - System response time
  - User assistance facilities
  - Error message processing
  - Command interaction

- It is best to consider these issues as important design issues at the beginning of the design
  - At this time, the modification is easier and the cost is low

Q1- system response time

- System response time
  - Refers to the time from the user completing a certain control action (for example, pressing the enter key or clicking the mouse) to the software giving the expected response (outputting information or performing an action)
- System response time has two important attributes
  - length
    - If the system response time is too long, users will feel nervous and frustrated
    - When the user's work speed is determined by the man-machine interface, the system response time is too short, which will force the user to speed up the pace of operation, which may make mistakes
  - Variability
    - Refers to the deviation of the system response time from the average response time
    - Even if the system has a long response time and low response time variability, it helps users establish a stable work rhythm

Q2-User Help Facilities

- Common help facilities can be divided into
  - Integrated help facility: designed in the software from the beginning
    - Can shorten the time for users to get help and increase the friendliness of the interface
  - Additional help facilities: added to the software after the system is built, often online user manuals with limited query capabilities
  - Integrated help facilities are better than additional help facilities
- Problems that must be solved when designing specific assistance facilities
  - During the user's interaction with the system, is it possible to get help information about any function of the system at any time?
    - There are two options: to provide help information for some functions and help information for all functions.
  - How do users request help?
    - There are 3 choices: help menu, special function keys and HELP command.
  - How to display help information?
    - There are 3 options: In a separate window, point out a reference to a document (not ideal) and display a short prompt in a fixed position on the screen.
  - How does the user return to normal interaction?
    - There are two options: the back button and function keys on the screen.
  - How to organize the help information?
    - There are 3 options: flat structure, hierarchical structure of information and hypertext structure

Q3-Error message processing

- Error messages and warning messages are "bad news" given by the interactive system when a problem occurs
  - If the error message is not well designed, it will provide users with useless or even misleading information, which will increase the user's frustration.
- The attributes that the error message or warning message should have
  - Information should describe the problem in terms that users can understand
  - The information should provide constructive advice that helps to recover from the error
  - The information should point out what negative consequences the error may cause (for example, damage to data files), so that the user can check whether these problems have occurred, and solve them in time when they do occur.
  - Information should be accompanied by auditory or visual cues
    - For example, when the information is displayed, a warning bell is emitted at the same time, or the information is displayed in a flashing manner, or the information is displayed in a color that clearly indicates an error
  - Information cannot be emotionally charged

Q4-Command interaction

- Users can select software functions from the menu, or call the software functions through a sequence of keyboard commands
- When providing command interaction, the following design issues must be considered
  - Does each menu option have a corresponding command?
  - What kind of command format is used? There are 3 options: control sequence (for example, Ctrl+P), function keys and key-in commands
  - How difficult is it to learn and memorize commands? What should I do if I forget the commands?
  - Can users customize or abbreviate commands?
  - Is there a "command macro mechanism"?
  - Under ideal circumstances, all application software has a consistent command usage method

User interface design process

- User interface design is an iterative process
  - Usually the design model is created first, and then the prototype is used to realize the design model, which is tested and evaluated by users, and then modified according to user opinions
- User Interface Toolbox/User Interface Development System
  - To simplify the creation of windows, menus, device interactions, error messages, commands, and many other elements of the interactive environment, various routines or objects are provided
    - Not only language-based but also graphical-based

Evaluation criteria for user interface design

- The length and complexity of the specifications of the system and its interface
  - It indicates the amount of work required for users to learn to use the system
- The number of commands or actions, the average number of parameters of a command, or the number of individual operations in an action
  - Predicts the interaction time and overall efficiency of the system
- The number of actions, commands, and system states included in the design model
  - It indicates how much content users need to remember when learning to use the system
- Interface style, help facilities and error handling protocol
  - It indicates the complexity of the interface and the user acceptance of the interface

Guidance and suggestions for human-computer interaction design

- General Interaction Guide
  - Keep consistency
    - Should use a consistent format for menu selection, command input, data display, and many other functions in the man-machine interface
  - Provide meaningful feedback
    - The user should be provided with visual and audible feedback to ensure that two-way communication is established between the user and the system
  - Require user confirmation before performing more destructive actions
    - If the user wants to delete a file, or overwrite some important information, or terminate the operation of a program
  - Allow most operations to be cancelled
    - UNDO or REVERSE function
  - Reduce the amount of information that must be memorized between operations
  - Improve the efficiency of conversation, movement and thinking
    - The number of keystrokes of the user should be minimized, and the distance of the mouse movement should be minimized when designing the screen layout
  - Allow mistakes
    - The system should be able to protect itself from serious errors
  - Classify actions by function and design screen layout accordingly
  - Provide help facilities that are sensitive to the user's work content
  - Use simple verbs or verb phrases as command names

Guidance and suggestions for human-computer interaction design

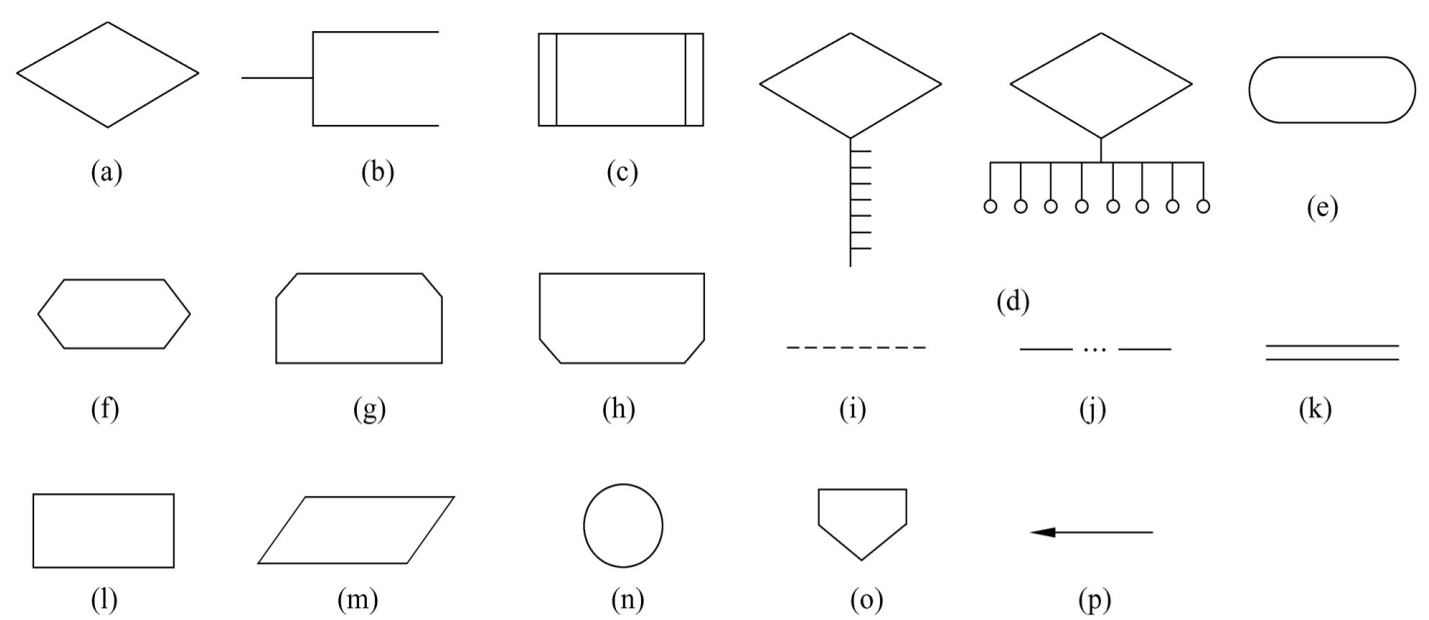
- Information display guide
  - Only display information related to the current job content
    - Users don't have to see irrelevant data, menus and graphics
  - Don't overwhelm users with data
    - For example, you can replace huge tables with graphs or charts
  - Use consistent markings, standard abbreviations, and predictable colors
    - The meaning of the display should be very clear, and the user can understand it without referring to other sources of information
  - Allow users to maintain a visual context
  - Generate meaningful error messages
  - Use capitalization, indentation, and text grouping to aid understanding
  - Use windows to separate different types of information
  - Use "analog" display to represent information to make it easier for users to extract
  - Use the display efficiently

Guidance and suggestions for human-computer interaction design

- Data input guide
  - Minimize user input
  - Maintain consistency between information display and data input
    - The displayed visual features should be consistent with the input field
  - Allow users to customize input
  - The interaction should be flexible and can be adjusted to the user's favorite input method
  - Disable commands that are not applicable in the current action context
  - Let users control the flow of interaction
  - Provide help for all input actions
  - Eliminate redundant input

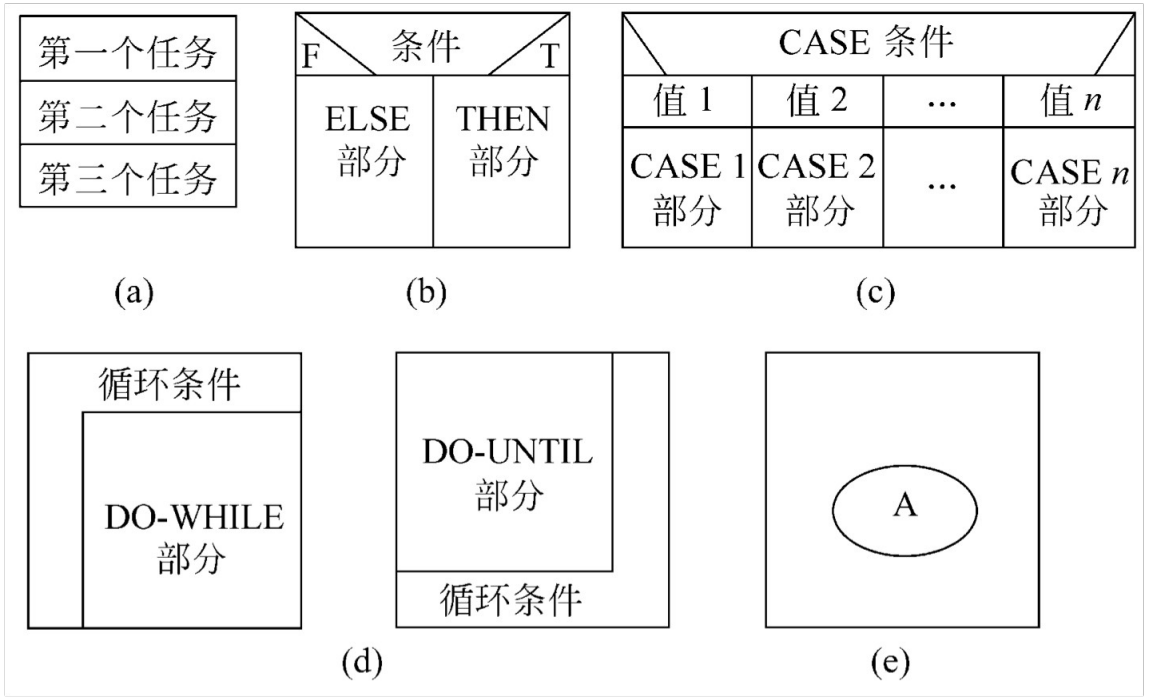
Process design tools

- Program flow chart
  - The program flow chart is not a good tool for step-by-step refinement by nature
    - It is easy to mislead programmers to consider the control flow of the program prematurely, instead of considering the global structure of the program
  - Arrows in the program flow chart represent control flow
    - The programmer is not subject to any constraints, and can completely disregard the spirit of structural programming and transfer control at will.
  - The program flowchart is not easy to express the data structure



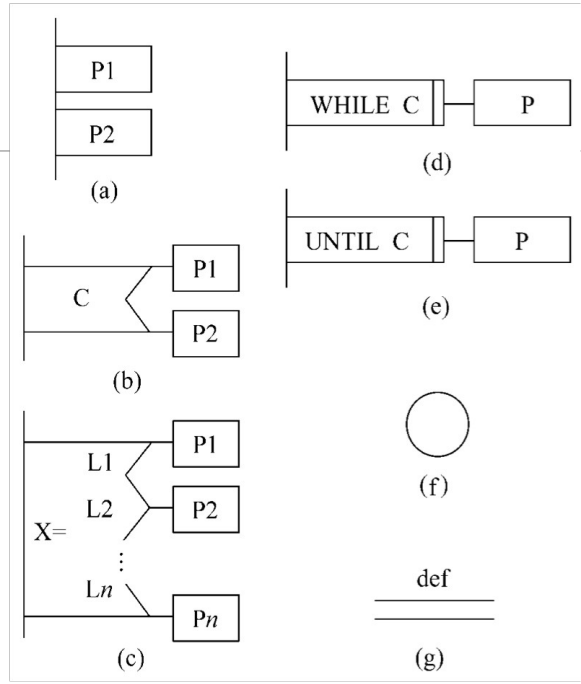
Process design tools

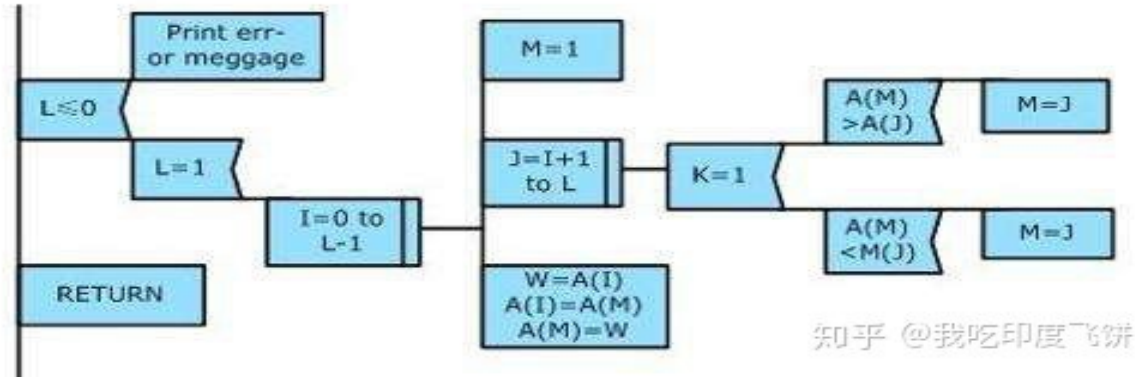
- Box diagram (N-S diagram)
  - Nassi and Shneiderman proposed a box diagram, also known as N-S diagram
    - The functional domain (that is, the scope of a specific control structure) is clear and can be seen from the box diagram at a glance
    - It is impossible to transfer control arbitrarily
    - It is easy to determine the scope of local and global data
    - It is easy to express the nesting relationship, and it can also express the hierarchical structure of the module



Process design tools

- PAD diagram (problem analysis diagram)
  - Programs designed using PAD symbols representing structured control structures must be structured programs
  - The program structure depicted in the PAD diagram is very clear
  - Use PAD diagrams to express program logic, which is easy to read, understand and remember
  - Easily convert PAD diagrams into high-level language source programs, which can be done automatically with software tools
  - It can be used to express program logic, and it can also be used to describe data structure
  - The symbols of the PAD diagram support the use of a top-down, step-by-step refinement method





Process design tools

- Decision table
  - The decision table can clearly indicate the correspondence between the complex combination of conditions and the action to be done
- The decision table consists of 4 parts
  - All conditions are listed on the upper left
  - The bottom left is all possible actions
  - The upper right is a matrix representing various combinations of conditions
  - The bottom right is the action corresponding to each condition combination
- Each column in the right half of the decision table is essentially a rule that specifies actions corresponding to a specific combination of conditions

Example

- The following takes the algorithm of checked baggage as an example to illustrate the organization method of the judgment table
- Assuming that a certain airline stipulates that passengers can check baggage weighing no more than 30kg for free
    - When the baggage weight exceeds 30kg
        - Charge 4 yuan per kilogram for the overweight portion of domestic passengers in first-class cabin
        - 6 yuan per kilo for the overweight portion of domestic passengers in other cabins
        - The charge per kilogram for the overweight portion of foreign passengers is twice that of domestic passengers
        - The fee per kilogram for the overweight part of the disabled passenger is half less than that of the normal passenger

- The decision table can clearly indicate the algorithm for calculating the baggage fee corresponding to each combination of the above conditions

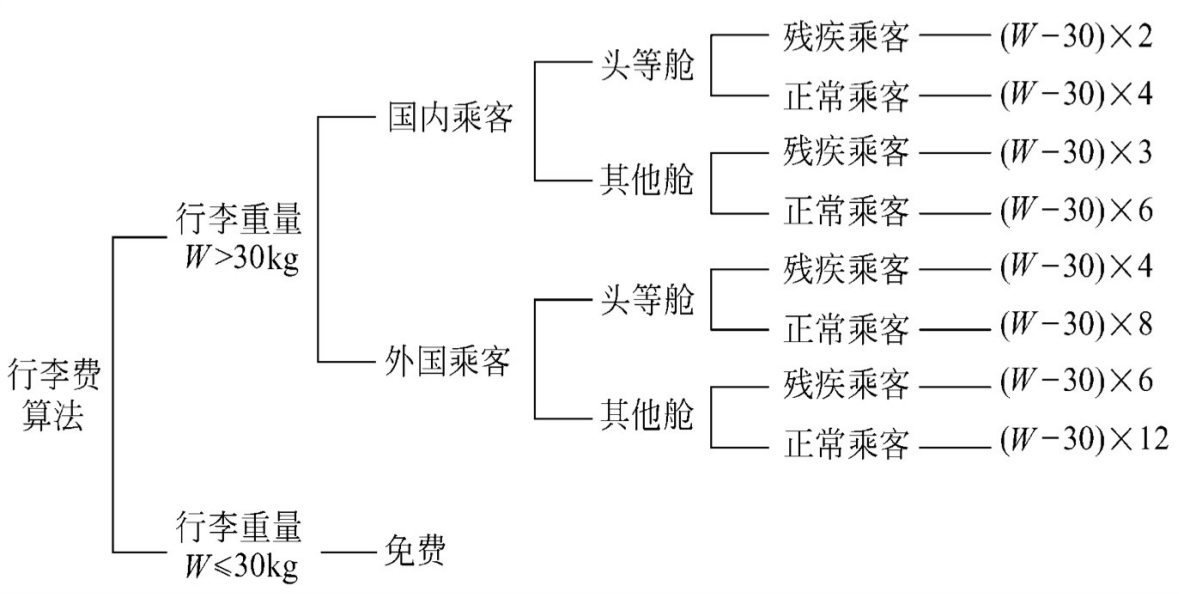
Example

- The following takes the alg… the organization method of the judgment table
- Assuming that a certain air… hing no more than 30kg for free
  - When the baggage we…
    - Charge 4 yuan pe… engers in first-class cabin
    - 6 yuan per kilo fo… er cabins
    - The charge per k… rs is twice that of domestic passengers
    - The fee per kilog… is half less than that of the normal passenger
- The decision table can clea… ee corresponding to each combination of the above c…

	规则								
	1	2	3	4	5	6	7	8	9
国内乘客		T	T	T	T	F	F	F	F
头等舱		T	F	T	F	T	F	T	F
残疾乘客		F	F	T	T	F	F	T	T
行李重量 $W \leqslant 30$kg	T	F	F	F	F	F	F	F	F
免费	×								
$(W-30)\times 2$				×					
$(W-30)\times 3$					×				
$(W-30)\times 4$		×						×	
$(W-30)\times 6$			×						×
$(W-30)\times 8$						×			
$(W-30)\times 12$							×		

Example

- Decision tree
  - Although the judgment table can clearly indicate the corresponding relationship between the complex combination of conditions and the action to be done, its meaning is not easy to read and requires intermittent learning
  - When the value of the data element is more than two (for example, assuming that the ticket needs to be subdivided into multiple classes such as first class, second class, and economy class), the simplicity of the judgment table will also decrease
  - The decision tree is a variant of the decision table, and it can also clearly indicate the correspondence between the complex combination of conditions and the actions to be done.



Process design tools

- Process design language
  - PDL, also known as pseudo code
  - Features
    - The fixed grammar of keywords provides the features of structured control structure, data description and modularity
    - The free grammar of natural language, which describes processing characteristics.
    - Means of data description
    - Module definition and call technology should provide various interface description modes

Process design tools

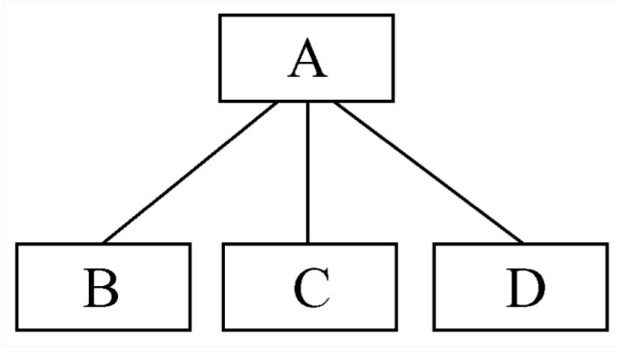
- Advantages of process design language
  - Can be inserted directly in the middle of the source program as a comment
  - Can use ordinary text editing program or word processing system to write and edit
  - There is an automatic processing program, and the program code can be automatically generated by PDL
- shortcoming
  - It is not as intuitive as a graphical tool. When describing the corresponding relationship between a complex combination of conditions and actions, it is not as clear and simple as the judgment table

Data structure-oriented design method

- Ultimate goal
  - Get a description of the process of the program
- Most suitable for use in the detailed design stage
  - After completing the software structure design, it is used to design the processing process of each module

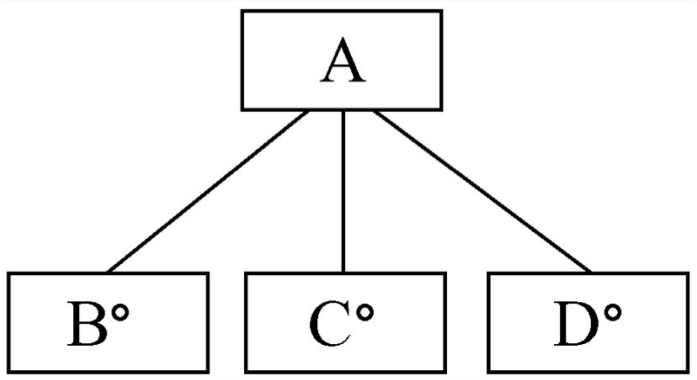
Data structure-oriented design method

- Jackson graph
  - Sequence structure
    - The data of the sequential structure is composed of one or more data elements, and each element appears once in a certain order



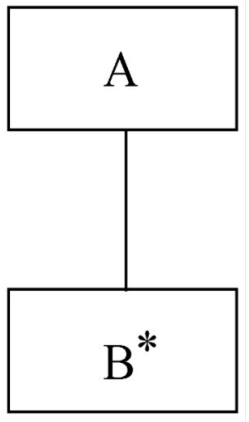
Data structure-oriented design method

- Jackson graph
  - Conditional structure
    - The data of the selected structure contains two or more data elements, and each time this data is used, one of these data elements is selected according to certain conditions



Data structure-oriented design method

- Jackson graph
  - Repeatative structure
    - Data with repetitive structure is composed of zero or more occurrences of a data element according to the conditions of use

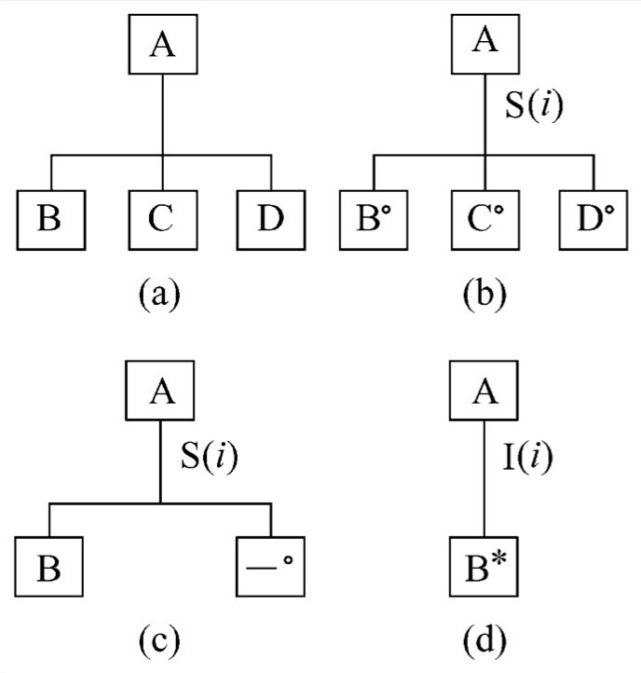


Data structure-oriented design method

- Advantages of Jackson diagram
  - Convenient to express hierarchical structure, and it is a powerful tool for top-down decomposition of structure
  - The image is intuitive and readable
  - Can express both data structure and program structure
    - Because structural programming also only uses the above three basic control structures
- Disadvantages of Jackson diagrams
  - When using this graphic tool to represent selection or repetition structure, the selection condition or loop end condition cannot be directly represented on the graph
    - Affects the expressive ability of the diagram, and it is not easy to directly translate the diagram into a program

Data structure-oriented design method

- Improved Jackson graph
  - When using this graphic tool to represent selection or repetition structure, the selection condition or loop end condition cannot be directly represented on the graph
    - Affects the expressive ability of the diagram, and it is not easy to directly translate the diagram into a program



Data structure-oriented design method

- The execution steps of the Jackson method
  - (1) Analyze and determine the logical structure of input data and output data, and describe these data structures with Jackson diagrams
  - (2) Find the corresponding data unit in the input data structure and output data structure
    - The so-called corresponding relationship refers to
      - There is a direct causal relationship, a data unit that can be processed at the same time in the program
        - For repetitive data units, the order and number of repetitions must be the same to have a corresponding relationship

Data structure-oriented design method

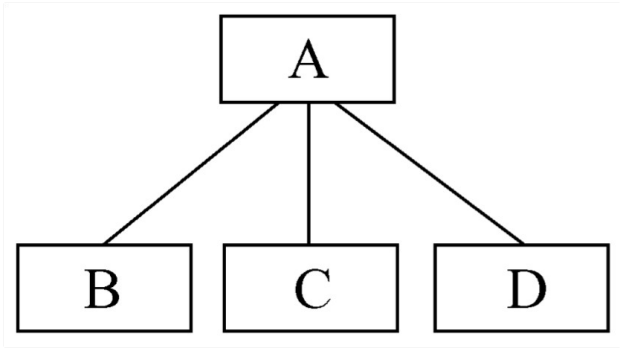
- The execution steps of the Jackson method [cont.]
  - (3) Use the following 3 rules to derive the Jackson diagram describing the program structure from the Jackson diagram describing the data structure
    - First, for each pair of data units that have a corresponding relationship, draw a processing box on the corresponding level of the program structure diagram according to their level in the data structure diagram.
      - Note that if the pair of data units are at different levels in the input data structure and the output data structure, the level of their corresponding processing blocks in the program structure diagram and the lower level of them in the data structure diagram That corresponds
    - Second, according to the level of each remaining data unit in the input data structure, draw corresponding processing boxes for them on the corresponding level of the program structure diagram.
    - Third, according to the level of each remaining data unit in the output data structure, draw corresponding processing boxes for them on the corresponding level of the program structure diagram.
    - The Jackson diagram depicting the structure of the program should be derived by integrating the hierarchical relationship between the input data structure and the output data structure
      - In the process of exporting the program structure diagram, since the improved Jackson diagram stipulates that there cannot be repeated or selected elements in the elements constituting the sequence structure, it may be necessary to increase the intermediate level processing frame.

Data structure-oriented design method

- The execution steps of the Jackson method [cont.]
  - (4) List all operations and conditions (including branch conditions and loop end conditions), and assign them to appropriate positions in the program structure diagram
  - (5) Use pseudo code to represent the program
    - The pseudo code used in the Jackson method corresponds exactly to the Jackson graph

Data structure-oriented design method

- Jackson graph
  - Sequence structure
    - The data of the sequential structure is composed of one or more data elements, and each element appears once in a certain order



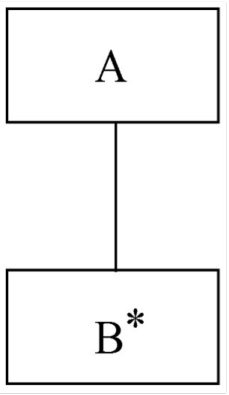
Corresponding peusdo code:

"seq" and "end" are keywords.

```
A seq
        B
        C
        D
A end
```

Data structure-oriented design method

- Jackson graph
  - Repeatative structure
    - Data with repetitive structure is composed of zero or more occurrences of a data element according to the conditions of use



Corresponding peusdo code:

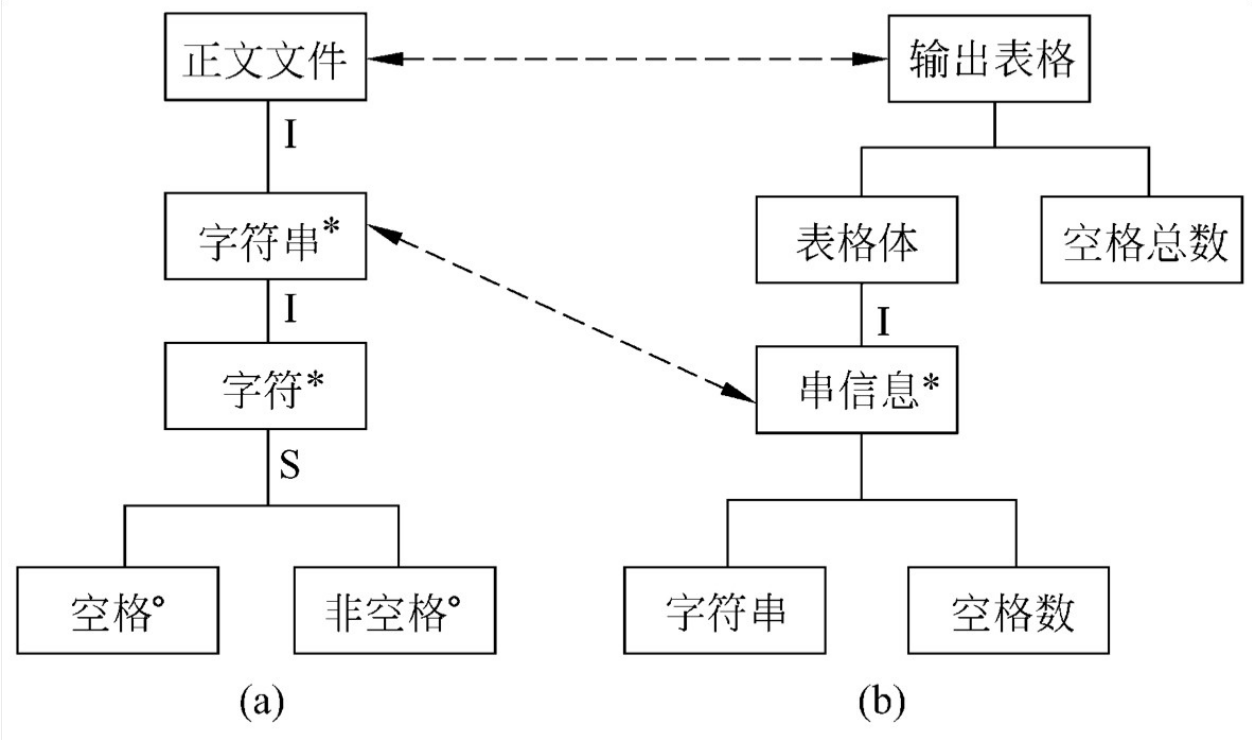
"iter", "until", "while" and "end" are keywords.

cond1 is the condition.

```
A iter until (or while) cond
    B
A end
```

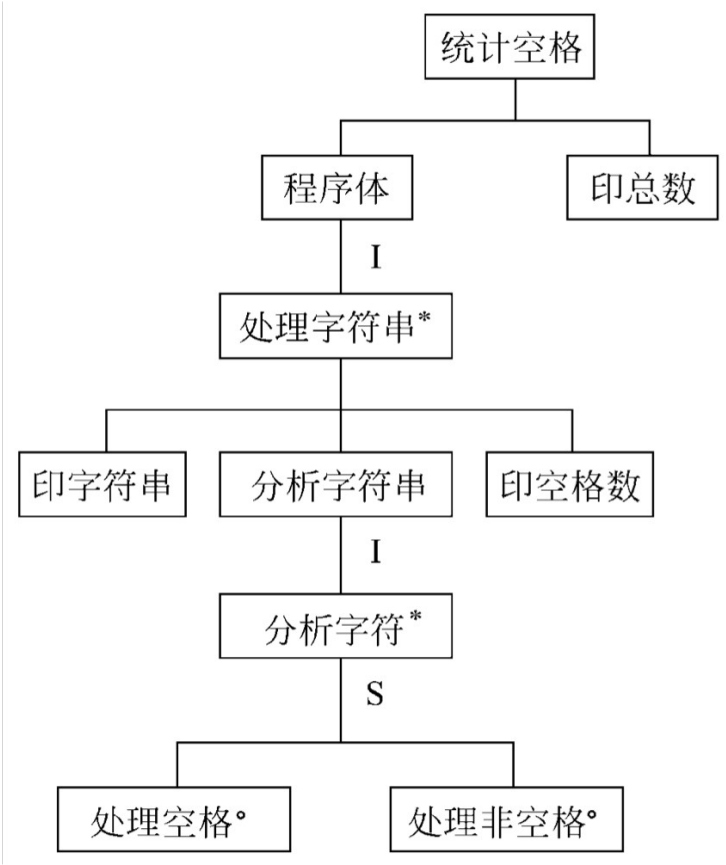
Example

- A text file is composed of several records, and each record is a character string. It is required to count the number of space characters in each record and the total number of space characters in the file. The required output data format is that after each line of input string is copied, the number of spaces in this string is printed on a new line, and the total number of spaces in the file is printed out at the end.



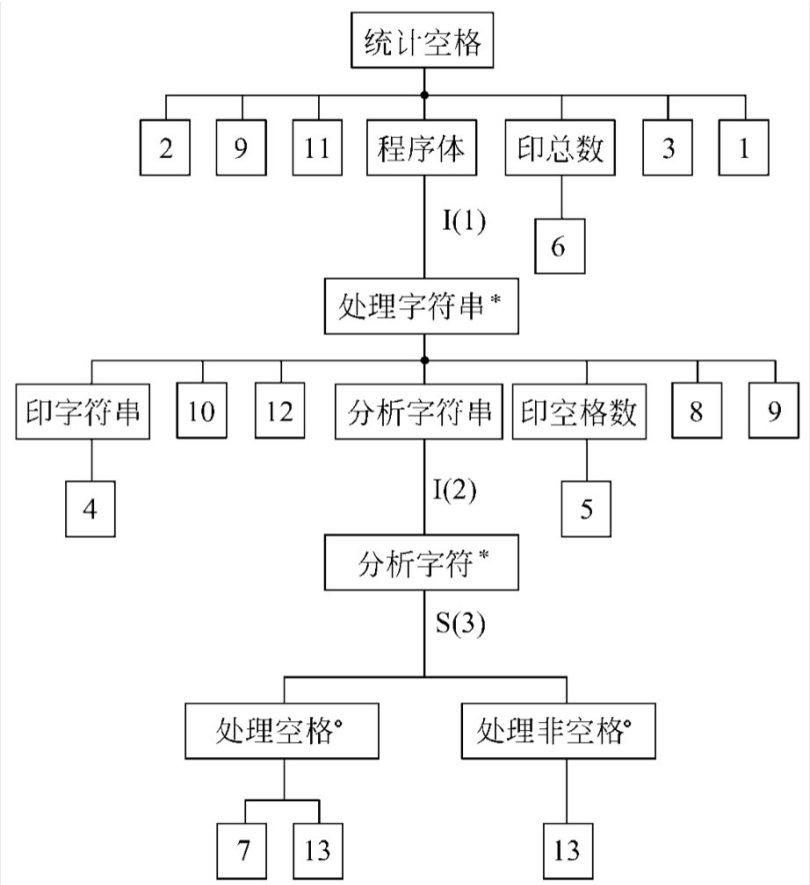
Example

- A text file is composed of several records, and each record is a character string. It is required to count the number of space characters in each record and the total number of space characters in the file. The required output data format is that after each line of input string is copied, the number of spaces in this string is printed on a new line, and the total number of spaces in the file is printed out at the end.



Example

- A text file is composed of several records, and each record is a character string. It is required to count the number of space characters in each record and the total number of space characters in the file. The required output data format is that after each line of input string is copied, the number of spaces in this string is printed on a new line, and the total number of spaces in the file is printed out at the end.



Quantitative measurement of program complexity

- Multiplying the complexity of the program by an appropriate constant can estimate the number of errors in the software and the amount of work required for software development
- The results of quantitative measurements can be used to compare the pros and cons of two different designs or two different algorithms
- The quantitative complexity of the program can be used as the precision limit of the module size

Quantitative measurement of program complexity

- Flow graph (流图)
  - The McCabe method quantitatively measures the complexity of the program according to the complexity of the control flow of the program, and the measured result is called the loop complexity of the program
  - The so-called flow graph is essentially a "degraded" program flow chart
    - It only depicts the control flow of the program
    - Does not show specific operations on data and specific conditions for branching or looping at all
    - Process design results expressed in any method can be translated into flow diagrams
- How to draw
  - Use circles to represent nodes in the flow graph, and one circle represents one or more sentences
    - A sequential processing block sequence and a diamond decision block in the program flowchart can be mapped to a node in the flow graph.
  - The arrow line in the flow diagram is called the edge, which is similar to the arrow line in the program flow chart and represents the control flow
    - An edge in the flow graph must end at a node, even if this node does not represent any sentence
      - Is actually equivalent to an empty statement
  - The area enclosed by edges and nodes is called area
    - When calculating the number of areas, the area outside the map that is not enclosed should be included

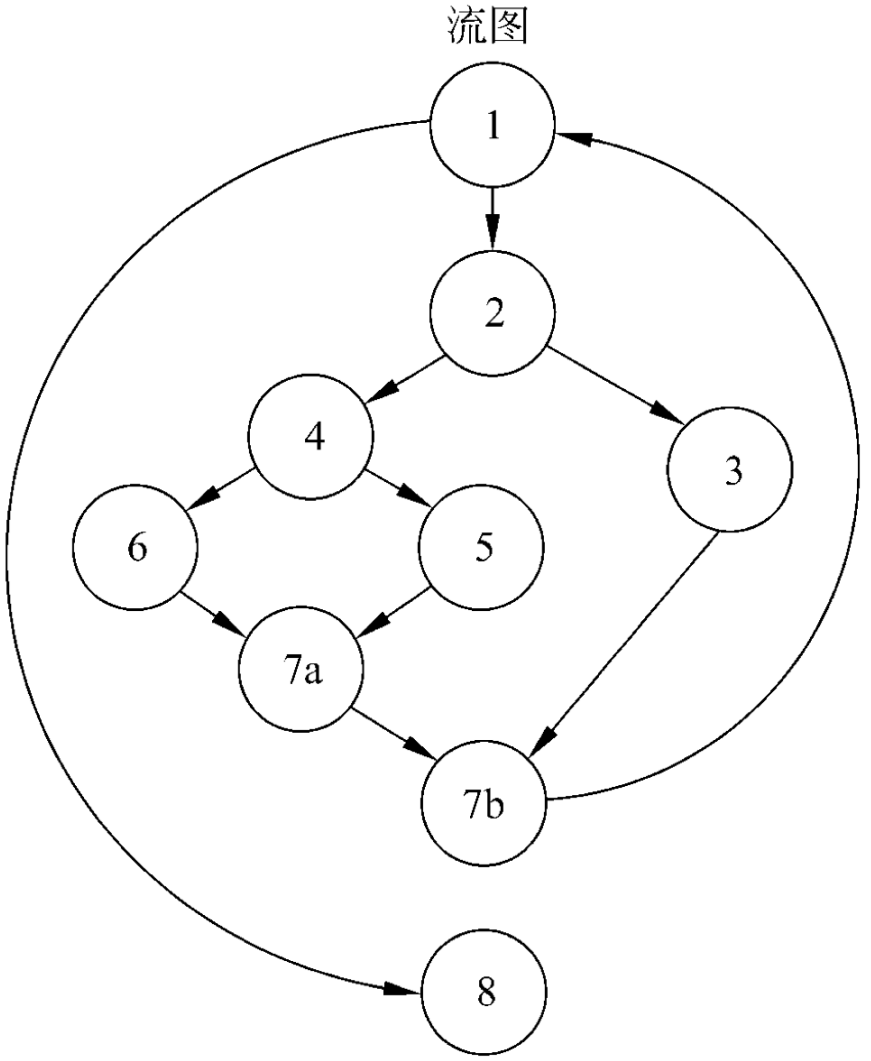
Quantitative measurement of program complexity

- Flow graph
  - The McCabe method qu … exity of the control flow of the … am
  - The so-called flow grap …
    - It only depicts the …
    - Does not show spe … l
    - Process design res …
- How to draw
  - Use circles to represen …
    - A sequential proce … can be mapped to a node i …
  - The arrow line in the fl … m flow chart and represents tl …
    - An edge in the flo … ce
      - Is actually eq …
  - The area enclosed by e …
    - When calculating t … included

PDL

```
procedure:sort
1:    do while records remain
2:       read record;
         if record field 1=0
3:          then process record;
               store in buffer;
               incremert counter;
4:       elseif record field 2=0
5:          then reset counter;
6:          else process record;
               store in file;
7a:      endif
         endif
7b:   enddo
8:    end
```

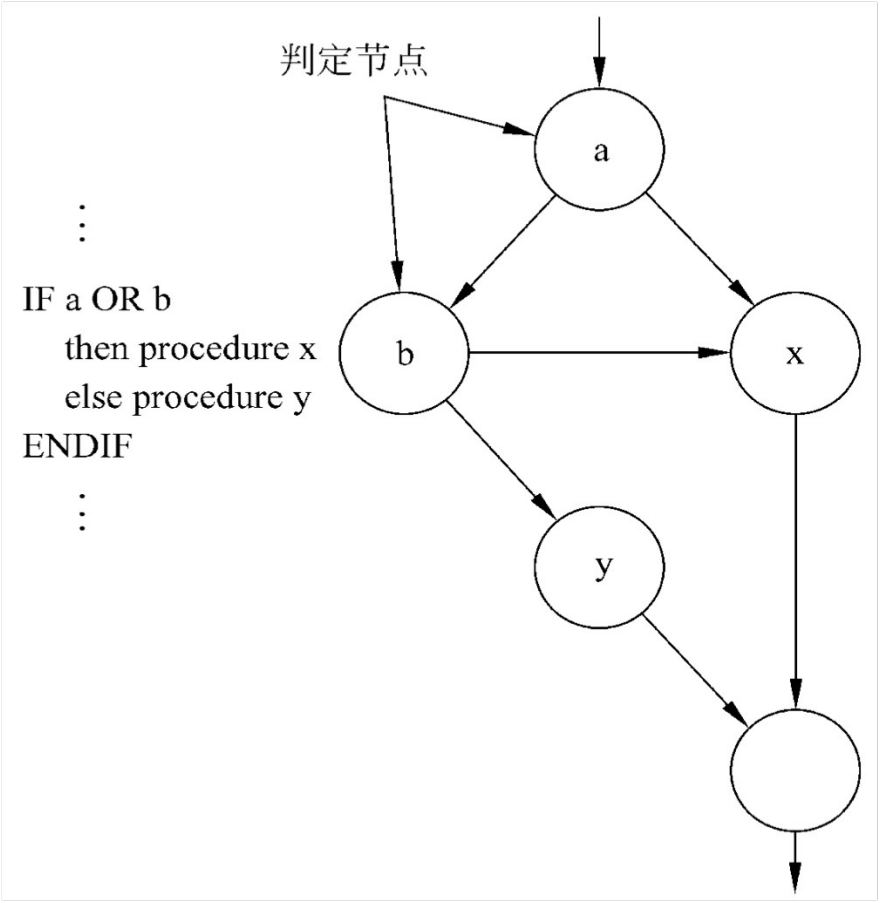
流图



Quantitative measurement of program complexity

- Flow graph (流图)

  Flow graph mapped from PDL containing compound conditions



Method of calculating ring complexity

- Circular complexity quantitatively measures the logic complexity of a program
- Based on the flow graph, you can use any of the following three methods to calculate the ring complexity.
  - (1) The number of regions in the flow graph is equal to the loop complexity
  - (2) The ring complexity of flow graph *G* is V(*G*)=E-N+2
    - E is the number of edges in the flow graph, N is the number of nodes
  - (3) The ring complexity of flow graph *G* V(*G*)=P+1
    - P is the number of decision nodes in the flow graph

Method of calculating ring complexity

- Programs with high loop complexity are often the most difficult and problem-prone programs
  - Practice shows that the appropriate size of the module is $V(G) \leq 10$
    - $V(G)=10$ is a more scientific and precise upper limit of the module size