

用户界面设计

- 用户界面应具备的特性
- 用户界面设计的任务分析
- 用户界面任务和工作设计
- 界面设计的基本类型
- 数据输入界面设计
- 数据显示界面设计
- 控制界面的设计

- 1. 界面软件设计概述

用户界面应具备的特性

■ 可使用性

- ◆ 使用的简单性
- ◆ 用户界面中的术语标准化和一致性
- ◆ 拥有HELP帮助功能
- ◆ 快速的系统响应和低的系统成本
- ◆ 用户界面应具有容错能力

■ 灵活性

- ◆ 算法的可隐可显性
- ◆ 用户可以根据需要制定和修改界面方式
- ◆ 能够按照用户的希望和需要，提供不同详细程度的系统响应信息
- ◆ 与其它软件系统应有标准的界面
为使用户界面具有一定的灵活性，需要付出代价，而且有可能降低软件系统的运行效率。

■ 复杂性和可靠性

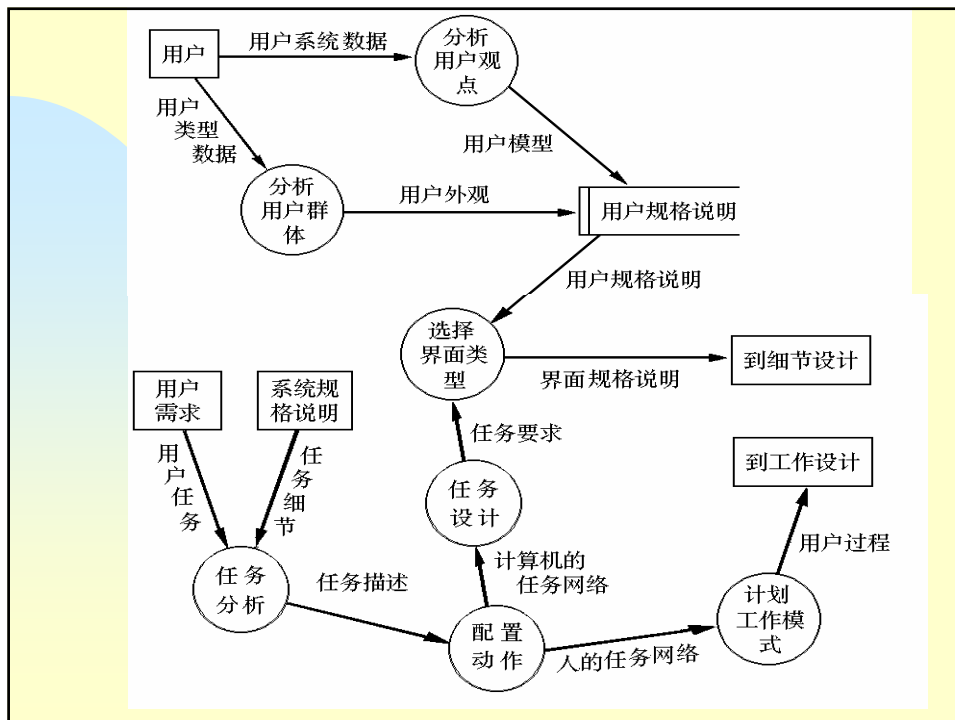
- ◆ 用户界面的规模和组织的复杂程度就是界面的复杂性。
- ◆ 在完成预定功能的前提下，应当使得用户界面越简单越好。但不是把所有功能和界面安排成线性序列就一定简单。
- ◆ 用户界面的可靠性是指无故障使用的间隔时间。
- ◆ 用户界面应能保证用户正确、可靠地使用系统，保证有关程序和数据的安全性。



■ 2. 设计人-机交互子系统

用户界面设计的任务分析

- 这一部分工作应与软件系统的需求分析同步进行。它主要包括
 - 用户特性分析
 - 用户工作分析
 - 记录有关系统的概念和术语
 - 确定界面类型
- 等4个步骤。



用户特性分析

- 用户界面是适应人的需要而建立的，因此，首先要弄清将要使用这个界面的用户类型。
- 用户特性分析的目的是要详细了解所有用户的技能和经验，以便能够预测用户对不同界面设计会做出什么反应，这样在更改界面时，就能做出正确的判断。

1. 用户类型

- ◆ 外行型
- ◆ 初学型
- ◆ 熟练型
- ◆ 专家型

用户的类型并不是一成不变的。因此，要做用户特性测量，以帮助设计者选择适合于大多数用户使用的界面类型和支持级别。

2. 用户特性度量

- 用户特性的度量与用户使用模式及观测到的用户群体能力有关。
 - ◆ **用户使用的频度**：即系统是否经常使用。
 - ◆ **用户是否能够自由选用界面**：所有的界面都应当是良好的。
 - ◆ **用户对计算机的熟悉程度**：对计算机的熟悉程度决定了要使用户达到熟练程度需要多少训练。

- ◆ **用户知识**：有些用户已有相当多的计算机编程和操作的知識。他们需要一种灵活的可编程的或命令语言的界面。
- ◆ **用户思维能力**：这是对用户的综合知识和智力的衡量。
- ◆ **用户的生理能力和技能**：主要指人的视觉、听觉、认知及记忆等方面的特点。因此，应该在这一方面去收集信息。

- 用户群体的度量可以用**打分**的办法来简单地度量。
 - ◆ 平均值反映了用户群体的平均特性。
 - ◆ 标准偏差反映了用户群体的差异大小，主要用于判断用户界面对不同类型的用户的适应范围。

一个图书馆的计算机借阅系统

- 由于图书馆的馆员将每天使用这个系统，因此他们**使用系统的频度很高**。他们**以前几乎没有人使用过计算机**，也没有人有自动化图书馆借阅系统和计算机系统的知识。他们的**绝大多数的智力水平在中等以上**。

用户特性表

系统 ID	自动化图书借阅系统		
群体 ID	图书馆馆员		
	中值	范围	
身体技能	打字 (一些)		
任意性	无		
频率	8	2~10	
计算机熟悉程度	2	1~4	
用户知识	1	1~3	
智力能力	6	4~8	
总分	17	8~25	

用户工作分析

- 用户工作分析，也称为任务分析。它是系统内部活动的分解。
- 用户工作分析与需求分析中结构化分析的方法类似，采用自顶向下，逐步进行功能分解。
- 与常规的功能分解不同的是，所有的系统任务，包括与人相关的活动，都要考虑在内。

- 系统的功能分解，可以用数据流图和数据词典描述。
- 任务可以由一组动作构成，它们规定了为实现这个任务所必须的一系列活动。
- 任务的细节可以使用结构化语言来表达。它描述了动作完成的序列及在完成动作时的所有例外情况。

用户模型和观点

建立用户的思维模型，是为了帮助建立设计者界面模型。

- 理论认知模型
- 用户知识模型
- 用户特性模型
- 用户任务模型
- 用户观点



(1) 理论认知模型

- 由心理学家建立的模型，目的是要了解人的思维过程。
- 例如，典型的问题解决模型**GOMS** (Goals Operator Methods Selection) 就是把一组目标和子目标概括在一个问题空间内，把目标分解成为子目标网络，每一个子目标都有与之相关的测试，通过使用某种方法搜索这个网络，在每一个结点上测试事先做出的假设。



(2) 用户知识模型

- 这种模型描述了一定范畴内的知识以及类型之间的联系。
- 模型借助各个用户的知识来建立，用遍历知识网络的方法来评价用户如何学习。



(3) 用户特性模型

- 这些模型主要依据在**用户特性分析**所描述的技能和能力对用户进行分类，它们也被称为**用户外观**。



(4) 用户任务模型

- 这种模型根据用户要求中提出的任务的**功能和操作次序**，构想该任务的概念。
- 通过它，可以了解用户知道多少有**关系系统操作的知识**，用户期望系统应当如何**工作等等**。
- 一个界面如果与用户预想的**实现和操作方式一致**，它就**容易为用户接受**。



(5) 用户观点

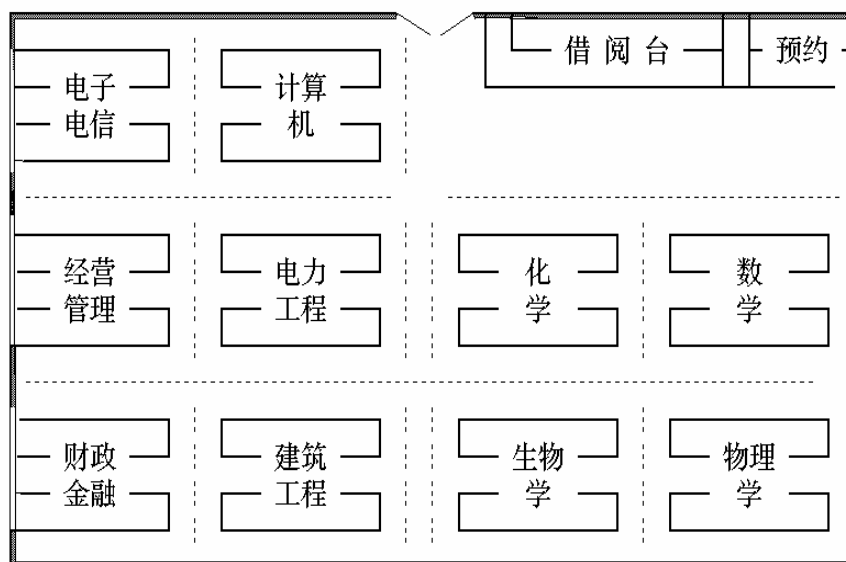
- 指系统结构的用户模型。是用户描述和设想现行系统结构的方法。
- 它可以用图形化的方式表示，也可以用语言方式表示。



任务模型

- 任务模型包括**静态的**和**动态的**两种。
- 系统的**静态模型**就是对象—关系模型，用于展现界面的结构设计；
- 系统的**动态模型**是对系统操作或与会话设计有关的对象的动态行为的描述。
- 模型由一些结构组成，既要依据静态的对象与其关系，又要依据活动的过程顺序。

- 例如, 在一个图书馆中, 用户可以将系统视为书籍。这些书籍依次放在按照主题分区排列的书架上, 它们可用一个层次结构表示。图书馆的其它部分, 如流通台、后备书库、公共书库、将要上架的书等, 它们在空间中可看成一个网络。
- 用户可使用对书籍进行分类的 (逻辑) 术语与布局 and 书籍流通的 (物理) 术语来描述一个图书馆, 组织他们关于现行系统的知识。



(b)

用户界面任务和工作设计

- 任务和工作设计的目的在于创造用户的工作环境。
- 通常，任务应组织得多样化一些，要与人的能力相适应。既不能完全由非常复杂的步骤组成，也不能太过简单而使操作者感到厌烦。

任务分配

- 在每个任务中，动作要分配给计算机、用户或者二者。
- 一般地，用户承担需要创造、判断和探索的任务，而计算机承担重复检查、计算和数据处理的任务。
- 数据录入、数据恢复和决策支持是混合任务。这些混合的任务需要通过人和计算机交互来共同完成。

- 任务分配产生两个网络。一个是人的任务网络，一个是计算机的任务网络。
- 人的任务网络说明如何安装、操作和使用系统，它最后将形成操作过程和用户手册的基础
- 计算机的任务网络则描述计算机应担负的工作。
- 这两种网络都可以用数据流图来设计。

任务分配步骤

- 检查数据流图，标出哪些是单独由计算机完成的任务、哪些是单独由人完成的任务、哪些是由两者共同完成的任务。
- 对于共同完成的任务，将任务的每一个动作分配给计算机或人。
- 构成新的计算机的任务网络和人的任务网络。
- 再进一步细化计算机与人的协同动作，以确定人和计算机如何交互。

- 一般情形，一个任务可以划分成一些子任务，按照某种顺序执行这些子任务，实现任务所要达到的目标，因此需要做出结构性的任务序列。
- 但许多事务处理任务是非结构性的，图书馆的馆长会以一个不可预测的顺序来召集一个碰头会、查询馆内各种业务情况。在这种情况下，不存在一个人的任务网络，而仅仅是一些用户需要个别完成的互不联系的任务。

工作方式和系统设计

- 工作设计的目的是使任务需求与操作员的能力相匹配。
- 在界面设计中关于人的因素特性的简单的度量如下：
 - ◆ 复杂性：推理、判断和决策的难度
 - ◆ 专心程度：对细节的注意力
 - ◆ 责任心：任务在总系统中的重要性
 - ◆ 变化性：在以上某一度量中的变化性

工作方式

- (1) **由一个人单独完成一个系统的目标。**例如，在商店业务处理系统中，检查顾客订单的任务中，所做的工作将是：数据输入、顾客赊欠清理、错误分辨和赊欠查询。
- (2) **任务进展跟踪。**确定命令在系统中什么地方执行、确定关键延迟的命令、调查拖延的原因、提出解决的建议等。

- 在做工作设计时，应该检查任务流程以确认不会出现任务过载。
 - ◆ 如果同时有多个任务需要用户给予注意，往往由于命令的时间冲突而使用户陷入困境，结果出现任务过载。
 - ◆ 即使任务流程计划的很好，但由于错误和一些意料之外的事情发生，如果错误处理过程计划的不完全、无条理，也会出现任务过载。

- 导致任务过载的原因，最常见的是出现工作负荷的高峰。因此要计划工作负荷使之均衡，并给出错误处理的时间。

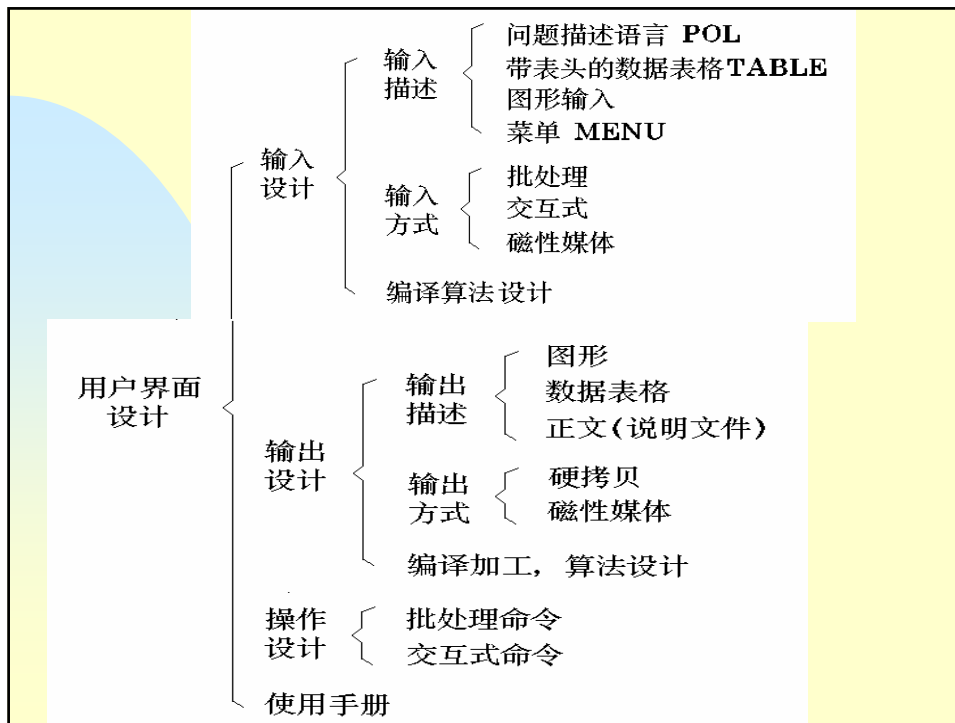


- 3. 界面设计基本类型

界面设计的基本类型

界面设计类型

- 在选用界面形式的时候，应当考虑每种类型的优点和限制。从以下几个方面来考察，进行抉择：
 - ◆ **使用的难易程度**：对于没有经验的用户，该界面使用的难度有多大。
 - ◆ **学习的难易程度**：学习该界面的命令和功能难度有多大。
 - ◆ **操作速度**：在完成一个指定操作时，该界面在操作步骤、击键和反应时间等方面效率有多高。
 - ◆ **复杂程度**：该界面提供了什么功能、能否用新的方式组合这些功能以增强界面的功能。
 - ◆ **控制**：人机交互时，是由计算机还是由人发起和控制对话。
 - ◆ **开发的难易程度**：该界面设计是否有难度、开发工作量有多大。



通常，一个界面的设计使用了一种以上的设计类型，每种类型与一个或一组任务相匹配。

■ 菜单(menu)界面的设计

菜单是由系统预先设置好的，显示于屏幕上的一组或几组可供用户选用的命令。这种菜单命令无需用户通过键盘打入，而是由系统将那些在一定环境下所需用的操作命令(菜单命令)，全部或部分地显示在屏幕上，供用户挑选。

◆ 按照显示的形象或样式来分类

正文菜单

正文菜单实质上是系统命令本身或者是其简写形式。在一个菜单中包含许多菜单项，可以象节目单那样，按某种约定，

在屏幕上成行或成列地排好。

简单的正文菜单的设置与选取方式：

- ① 首字符匹配方式
- ② 序号匹配方式
- ③ 亮条匹配方式

通用设备信息管理系统



- 0 --- 退 出
- 1 --- 数 据 编 辑
- 2 --- 设 备 检 索
- 3 --- 设 备 统 计
- 4 --- 报 表 打 印
- 5 --- 系 统 维 护

☆ 请键入选择的功能的序号 [0 - 5] ☆

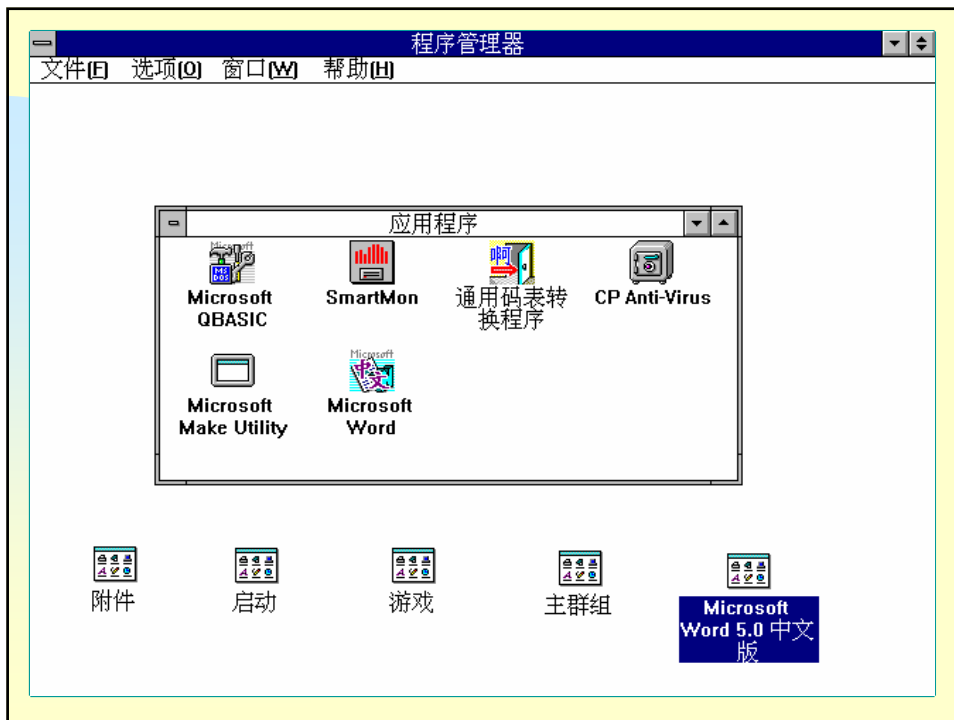
序号匹配方式

图标菜单

图标菜单，简称图标，是安置在一个小方框之中的一幅象形的或表意的图画。图标菜单项在功能上与正文菜单项没有什么差别，只不过图标更形象、更直观。

正文和图标混合的菜单

这种菜单基本上还是正文式的。从屏幕上看，它还是由字符串构成。只不过在字符串的前端或末端缀有图标。



◆ 按屏幕位置和操作风格来分类

④ 固定位置菜单

固定位置菜单每次总是在屏幕的相对固定的位置出现。

通常软件系统的功能划分多为树型结构，要求系统的控制结构也应是树型结构。固定位置菜单可以很方便地实现一种三层结构的菜单机制。

0层—主菜单：定位于屏幕中心或占领整个屏幕，菜单项的内容是所属各个子系统的名称。

1层—子系统层：这一层菜单在水平方向排成一行，安放于屏幕的上沿或下沿。菜单项的内容是所属各操作类的名称。当控制进入下一层次时，这一层菜单并不消失，仍然留在原来的位置上。

2层—当用户挑选了某一个操作类时，屏幕上将按垂直方向显示这组操作类的名称。这一层菜单项的内容与应用系统的具体某项操作有关。

采用固定位置菜单对系统加以控制，其突出的优点是层次清晰。

🕒 浮动位置菜单

浮动位置菜单(弹出式菜单)。其主要特点是：仅当系统需要时，它才被瞬时显示出来供用户选用，完成使命后它立即从屏幕上消失。它的显示位置可以根据用户的操作或根据当时的操作环境来决定。

弹出式菜单与用户当时正在执行的操作密切相关。但是使用弹出式菜单，由于菜单的瞬时性，用户不清楚自己当前究竟处在系统的什么位置，看不到自己处于那一层次。

🕒 下拉式菜单

下拉式菜单将固定位置菜单与浮动位置菜单揉和在一起。其结构分为两层：第一层是各个父菜单项的名字，它们排成一行，放置在屏幕上沿的一个菜单带区中。第二层是各个父菜单项的子菜单项，它们分别隶属于所对应的父菜单项。子菜单项平时是“藏”在屏幕后面的，仅仅当其父菜单项被选上时，才紧挨在其父菜单项的下方立即显示出来，以供用户进一步选用。选完之后它们又立即消失。

下拉式菜单只能描述系统的两个层次的控制结构，但是一般的系统的控制结构不只两层，解决办法有两个：

- ◆ 对于小系统或系统中的一个小范围，可使用弹出式菜单。这种办法比较实用，系统开销也比较小。

- ◆ 如果系统相当大，应将原系统分为若干子系统，子系统还可以再分解下去。对于各个子系统之间的接口控制，可以通过窗口来实现，而在各个子系统内部，仍可以使用下拉式菜单来进行界面管理。

🔍 嵌入式菜单

嵌入式菜单通常并不显式地成行成列地出现在屏幕上，而是混在应用之中。也可以说嵌入式菜单项本身就是它所在应用中的一部分内容。必要时可以用粗体字或字母高亮度显示等方式加以突出。

■ 图像

所谓图像，就是屏幕上一个矩形区域内包含的象素所构成的一个画面。

在用户界面中，加入丰富多彩的画面，将能够更形象地为用户提供有用的信息而达到可视化的目的。

◆ 图像的隐蔽和再现

实用系统中常常频繁地要求把屏幕上的某一块矩形区域内的图像隐蔽起来，然后在以后的适当时间，令其重新显现。

例如，下拉式菜单和弹出式菜单在显示时，就需要预先把将要被遮盖的区域中的原先的屏幕图像隐藏起来，而当选取菜单项的工作完成之后，又需要把原来隐藏的那些图形再现出来。为此，需要设置两个专门用来保存屏幕上用户工作区图像的内存缓冲区。

在执行这一类图像操作时，用于图像缓冲区的内存开销较大。

◆ 屏幕的滚动

通常，用于人机交互活动的物理屏幕仅能容纳用户需要显示的内容中的一部分内容，因此，用户必须通过屏幕滚动或其它措施才能看到全部内容。

屏幕滚动可以将用户的显示内容在物理屏幕上做平行移动，因此需要为此功能设置一个内存缓冲区。

◆ 图案的显示

在人机对话过程中使用图案显示，可以大大提高应用系统的视觉效果。连续地显示预先准备好的某动作过程中的一连串相接的瞬间图案，就构成为动画。

■ 对话

对话(对话框)是系统在必要时显示于屏幕上一个矩形区域内的图形和正文信息。通过对话,实现用户和系统之间的通信。

通常,对话是一种辅助手段,它也可以用来在系统的执行过程中,给出某种警告或提示信息。对话在屏幕上的出现方式与弹出式菜单类似,即瞬时弹出。与弹出式菜单不同的是,对话框在屏幕上显示的位置是由系统所设置的。

有三种对话形式。

🕒 必须回答式

必须回答式的对话在屏幕上出现时,用户必须给予回答,否则系统不再做任何其他工作。

File Name:

█

OK Cancel

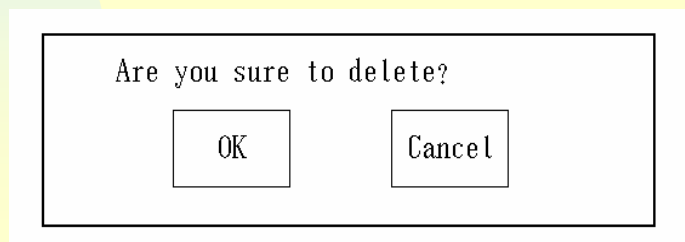
🕒 无需回答式

这类对话在屏幕上的出现,仅仅是为了告诉用户一些参考信息,不需要用户回答。因此,用户可以不理睬它,继续做原来的工作。

① 警告式

这类对话主要用于系统报错或者警告。警告式的对话，根据警告的内容，可以是必须回答式的对话，也可以是无需回答式的对话，同时给出一些必要的警告信息。

例如，在删除一个文件时，键入了删除文件命令后，为了确保不致误删不该删除的文件，屏幕上将出现一个警告式的对话框：



■ 问题描述语言POL

(Problem Oriented Language)

- **任务层：**分析用户需求，对软件全部功能和性能进行分解，确定目标和子目标。
- **语义层：**确定系统面对的对象以及基于这些对象之上的操作，建立问题解决的算法。
- **语法层：**将语义层的操作细化为由各种命令、用户操作、上下文关系和状态变量组成的语言，描述用户和计算机如何按照文法交互。
- **交互层：**根据诸如击键、设备驱动、显示等具体操作来定义用户操作命令和可供用户选择的处理过程。

任务层

- 一个任务可以自顶向下分解成由子任务构成的树形结构，每个任务和它们的动作带有注释和约束的结构化格式来描述。
- 各个任务涉及的对象叫做实体，而动作即是施加于其上的。
- 自顶向下的分解要一直做到实体和任务结构的详细设想完成为止。

语义层

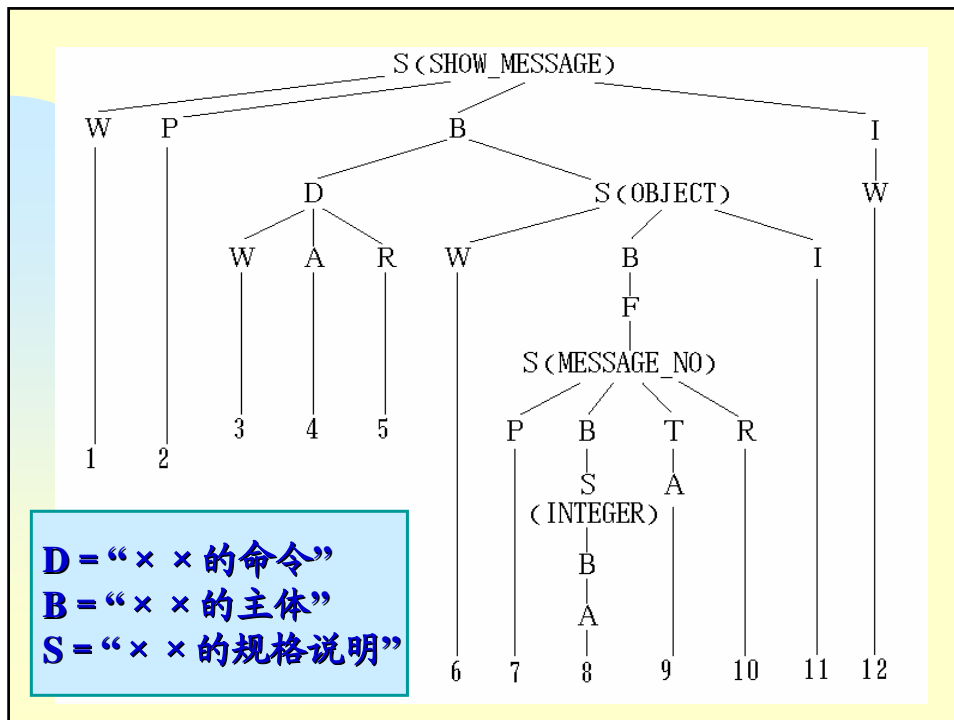
- POL描述概念实体和实现任务所需的运算。
- 概念实体可以是抽象的概念，例如某种联系，也可以是具体的对象，例如某个消息。
- 系统本身由实体的集合组成，而操作只是与某个对象相联系。
- 在语义层中还要给出为了完成任务所需的方法或过程，它们是一个程序段，说明任务的处理过程，涉及到某些实体和操作。

语法层

- 将操作和方法定义成命令。这些命令是由用户来使用的。
- 命令由语义操作建立，并具有上下文关系。通过上下文关系，利用显示、命令或状态变量，就可以描述整个系统。
- 在语法层，系统实体细化为更具体的对象，对应于屏幕显示，对象的描述将更加细致。包括行文的布局、屏幕的显示区域、目录结构等等。

交互层

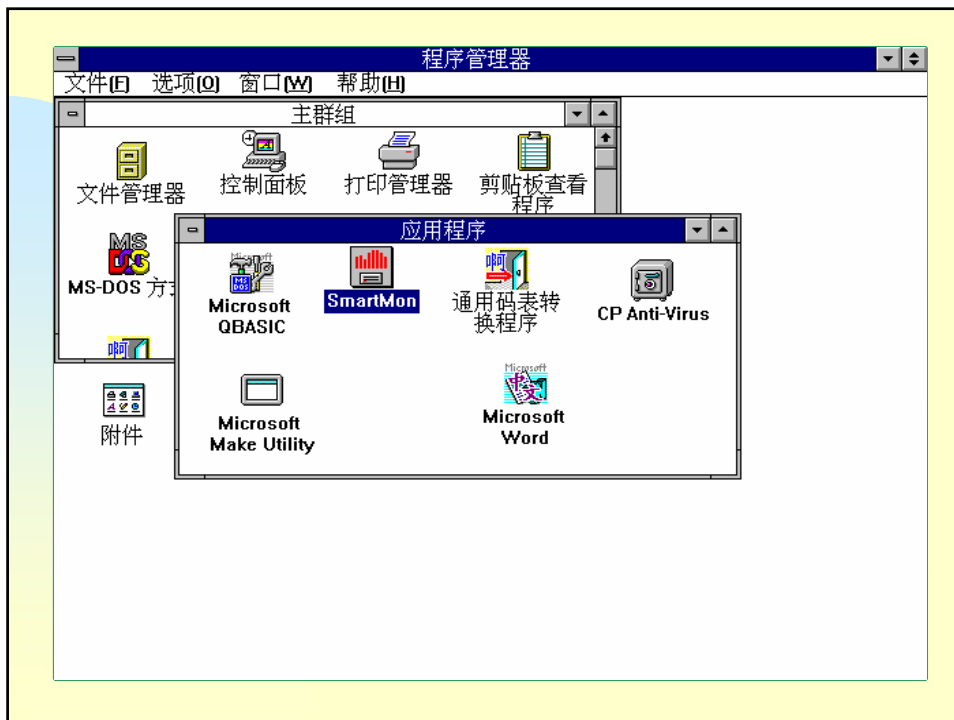
- POL的文法有**终极符** (自定义的不能再分的) 和由终极符组成的**非终极符**。终极符为
 - W — When** (时态规格说明原语)
 - P — Prompt** (原始系统动作：提示)
 - R — Response** (原始系统动作：响应)
 - A — Action** (原始用户动作：击键)这些终极符可组成非终极符结构。
- 对于每一个命令操作，可将提示、响应和状态的执行序列用树形结构来描述。

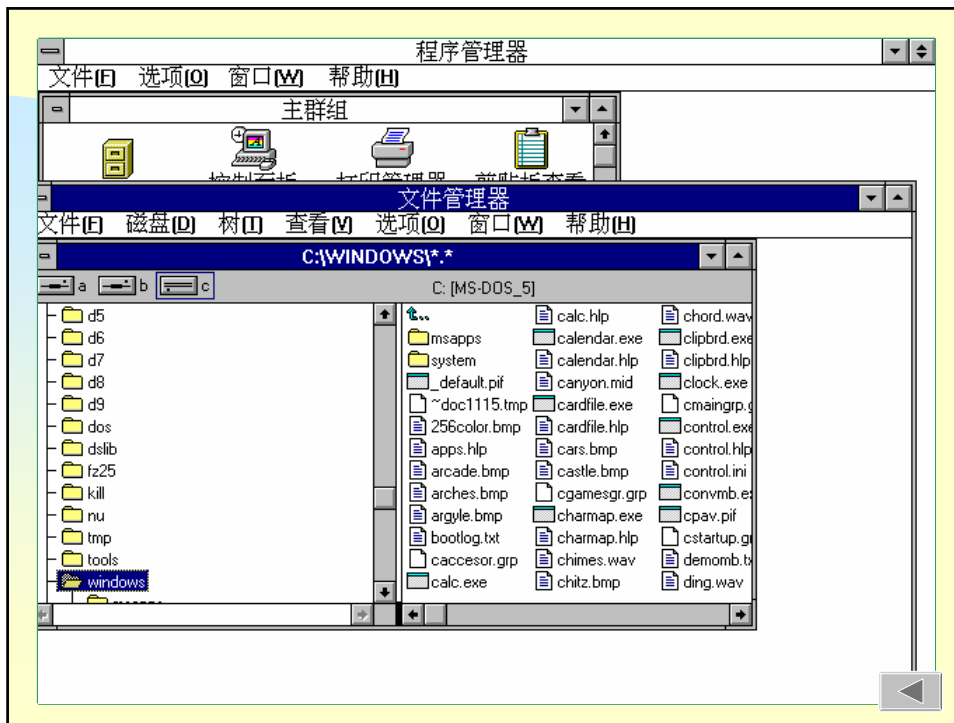


■ 窗口 (window)

- ◆ 窗口是指屏幕上的一个矩形区域，在图形学中叫做视图区 (viewport)。
- ◆ 用户可以通过窗口显示、观察其工作领域内的全部或一部分内容，并可以对所显示的内容进行各种系统预先规定好的正文和图形操作。
- ◆ 由于物理条件的限制，窗口面积的大小一般都不能满足用户要求，在窗口显示的内容只占用户空间的一部分。

- 在用户界面，为了能够通过窗口看到整个用户空间的全貌，一个简单的办法就是让窗口在用户空间滚动，即所谓屏幕滚动。
- 事实上，窗口本身并不属于用户空间，它仅仅是用于观察、组织用户空间的内容，并对其进行操作的用户接口工具。
- 习惯上把窗口视为虚拟屏幕，相对地，显示器屏幕就称为物理屏幕。采用滚动技术，通过窗口能够看到的用户空间，比物理屏幕显示的内容要多得多；而另一方面，在同一物理屏幕上又可以设置多个窗口，各个窗口可以由不同的系统或系统成分分别使用。





数据输入界面设计

数据输入是指所有供计算机处理的数据的输入。数据输入界面是系统的一个重要组成部分，它常占用户的极大部分使用时间。

数据输入的规则

- 数据输入界面的目标是尽量简化用户的工作，并尽可能地减少输入的出错率。为此，在设计时要考虑尽可能减少用户的记忆负担，使界面具有预见性和一致性，防止用户输入出错，以及尽可能增加数据自动输入。

- 在软件设计的范围，可以通过以下方法来减少用户输入的工作量。
 - ☆ 对共同的输入内容设置默认值（缺省值）。
 - 🕒 使用代码和缩写。
 - 🕒 自动填入已输入过的内容或需要重复输入的内容。
 - 🕒 如果输入内容是来自一个有限的备选集，可以采用列表选择或指点方式。
- 数据输入屏幕应当设计成尽量与输入格式相匹配。如果没有输入格式，或旧的输入格式设计得不好，就应当设计新的屏幕格式。

- 数据内容应当根据它们的使用频率，或它们的重要性，或它们的输入次序进行组织。数据输入对话设计的一般规则。
 - ☆ **明确的输入：**只有当用户按下输入的确认键时，才确认输入。这有助于在输入过程中一旦出现错误能及时纠错。
 - 🕒 **明确的动作：**在表格项之间自动地跳跃/转换并不总是可取的，尤其是对于不熟练的用户，往往会被搞得无所适从，要使用TAB键或回车键控制在表格项间的移动。

- 🕒 **明确的取消:** 如果用户中断了一个输入序列, 已经输入的数据不要马上丢弃。这样才能对一个也许是错误的取消动作进行重新思考。
- 🕒 **确认删除:** 为避免错误的删除动作可能造成的损失, 在键入删除命令后, 必须进行确认, 然后才执行删除操作。例如, 可以用 **Deleteyousure? [Y / N]** 来确认。
- 🕒 **提供反馈:** 若一个屏幕上可容纳若干输入内容, 可将用户先前输入的内容仍保留在屏幕上, 以使用户能够随时察看, 明确下一步应做的操作。

- 🕒 **允许编辑:** 在一个文件输入过程中或输入完成后, 允许用户对其编辑, 以修改他们正在输入的数据或修改他们以前输入的数据。应采纳一种前后一致的编辑方式。
- 🕒 **提供复原 (Undo):** 应允许用户恢复输入以前的状态。这在编辑和修改错误的操作经常用到。
- 🕒 **自动格式化:** 用户可以采用自由格式进行输入。例如, 用COBOL语言时, 用79而不是0079去适应格式PIC 9 (4) 的要求。输入对空格应不敏感。

🕒 **提示输入的范围：**应当显示有效回答的集合及其范围。例如，显示“在1~10之间输入折扣量”。

输入表格设计

- 数据表格设计是对较复杂的数据录入时使用得最广泛的一种对话类型。
- 这种方法是在屏幕上显示一张表格，类似于用户熟悉的填表格式，以供用户向计算机内输入数据。

- 在这种输入数据表格中，对于每一种输入信息，都有一个表格项，并带有一个表格项头，以提示输入信息的内容及位置。由用户使用移位键或者特殊定义的功能键控制屏幕上的光标，在各个表格项上定位以及数据的输入。用户键入数据之后，还可以以相同的屏幕格式显示、修改这些数据。
- 数据表格的优点是它的视觉布局用户比较熟悉，而且全部信息都可以显示在屏幕上，只要表格设计得好，操作步骤非常简便。

订单输入		日期 □□/□□/□□	
姓名 <input type="text"/>		邮政编码 □□□□□□	
地址 【国】 <input type="text"/>		【省】 <input type="text"/>	
【市】 <input type="text"/>		【区】 <input type="text"/>	
【街/门牌号】 <input type="text"/>			
目录号	数量	单价	合计
项 1	<input type="text"/>	<input type="text"/> · <input type="text"/>	<input type="text"/> · <input type="text"/>
项 2	<input type="text"/>	<input type="text"/> · <input type="text"/>	<input type="text"/> · <input type="text"/>
项 3	<input type="text"/>	<input type="text"/> · <input type="text"/>	<input type="text"/> · <input type="text"/>
项 4	<input type="text"/>	<input type="text"/> · <input type="text"/>	<input type="text"/> · <input type="text"/>
项 5	<input type="text"/>	<input type="text"/> · <input type="text"/>	<input type="text"/> · <input type="text"/>
项 6	<input type="text"/>	<input type="text"/> · <input type="text"/>	<input type="text"/> · <input type="text"/>
项 7	<input type="text"/>	<input type="text"/> · <input type="text"/>	<input type="text"/> · <input type="text"/>
项 8	<input type="text"/>	<input type="text"/> · <input type="text"/>	<input type="text"/> · <input type="text"/>
操作提示 ☆			
→ 光标右移		← 光标左移	
↑ 光标到上一字段		↓ 光标到下一字段	
ENTER 字段输入确认		DEL 字段输入作废	
ESC 不存盘退出		Ctrl+D 存盘退出	

数据表格设计的规则

🕒 数据验证

数据输入很容易出错。出错的原因可能是忽略了某一项，或在某一项的输入中键入了不正确的数据，或是数字或字符敲错。数据验证是要检查是否所有必需的项目都已填充，数据输入是否正确，是否合理。

出错验证可能得到以下三种结果：

☆ **致命错误**：引起处理混乱的错误。此时，用户要么重新输入一个正确的数据，要么退出输入，不允许其它做法。

🕒 **警告：**由很不可信的数据引起的错误。
此时应停止处理并提请用户重新输入数据。

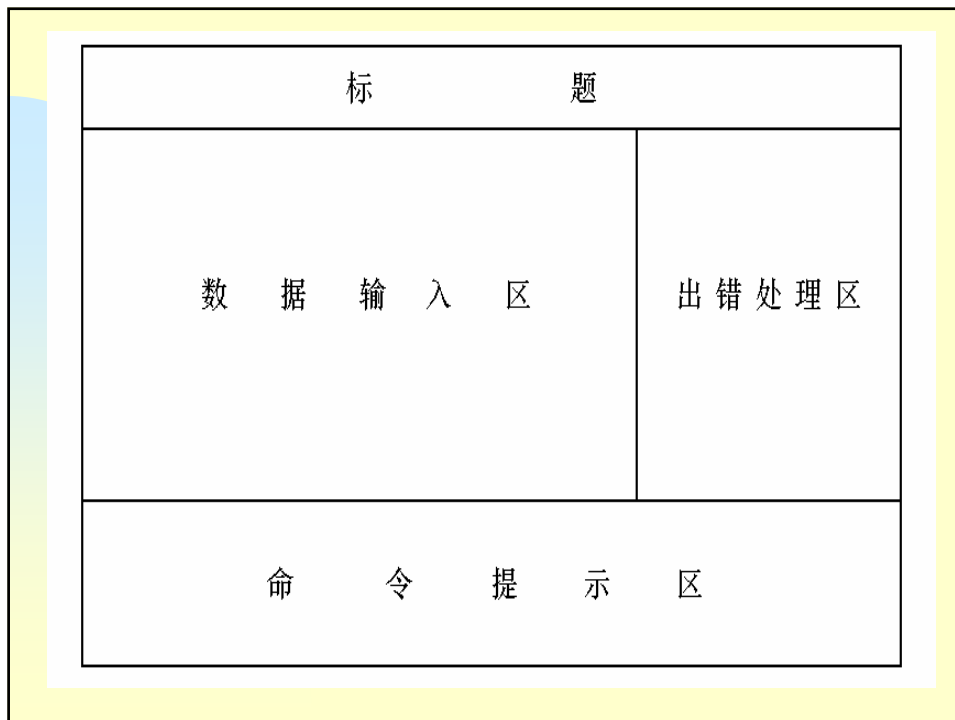
🕒 **建议：**由不大可信的数据引起的错误。
此时，处理不必停止，但要发出一个警告信息，使得用户或是立即停止检查，或是在处理结束时进行检查。

🕒 **屏幕设计**

通常把屏幕划分为数据输入、命令与出错处理三个区域。

在屏幕设计时，应注意以下几点：

☆ 应用不同的底色来区别各个区域



- 🕒 数据输入区内各个输入项应左侧对齐。在空间允许时，最好一行仅对应一个输入。
- 🕒 当回答中包含的字符数已知时，数据输入区应设置有相应格式的回答区域。
- 🕒 如果输入中有量的单位时，单位应在输入项中的左边指定。
- 🕒 标题、命令、重要的提示和填充指令应是简练、准确的。应使用为用户易于理解的词汇。

🕒 报信

- ◆ 报信对于通知用户出错的类型，为用户提供控制输入顺序和修改错误是很重要的。
- ◆ 在报信时所用的行文应当用词准确、简明、完备。
- ◆ 出错信息的提示应当报告错误出在何处，是什么错误，为什么错了，以及要修改错误应当采取什么措施。

- ◆ 提示信息不应使用专业术语，应当使用肯定方式和主动语态。例如，用“做...”来表达，不要用“不做...”来表达；用主动语态“按任意键继续...”，而不要用被动语态“通过按Break键此段可被终止”。

👉 数据输入对话控制

- ◆ 数据输入的对话控制是为了防止错误发生，
- ◆ 如果一旦发生了错误，它应为用户提供简单有效的改错方法。
- ◆ 数据输入对话则应当精心设计以便编辑和改错。

- ◆ 在对话序列中应设置若干断点，以休息和复位点来提高注意力。断点的设置要考虑到信息块及屏幕布局。

- 👉 对于比较复杂的控制命令序列，如果采用语法制导编辑技术，提供各类的语法模板，用户可以不必记忆许多繁琐的语法公式，关键字和标识符，只要按语法结构选择控制结构就可以了。在这种情况下，各种语法模板也可以看作是一些数据表格，用户在模板的提示下，键入需要的参数，使得这个复杂的命令序列得以顺利执行。

其它数据输入的方法

- 它们分为两类，一类是用菜单或关键词进行软件设计，另一类是用硬件方法来自动完成全部或部分数据输入任务。

☆ 菜单选择输入

如果数据从一个确定的可供选择的清单中选取输入，则可用菜单方式。方法很简单，把所有的选择项都显示在屏幕上，用户只需输入代表各项的数字代码，就可选择一个或几个数据，较复杂的选择方式是使用光笔或鼠标器对文字菜单或图标进行选择。

时令水果蔬菜销售					
订单号码 1024	输入订单				
	日期 26/5/96				
输入水果号码，选择所需数量	或 输入蔬菜号码，选择所需数量				
↓	↓				
1 ---- 苹果	6 ---- 土豆				
2 ---- 桔子	7 ---- 菜花				
3 ---- 香蕉	8 ---- 韭菜				
4 ---- 梨	9 ---- 油菜				
5 ---- 菠萝	10 ---- 西红柿				
输入10以内的数字: <input style="width: 100px;" type="text"/>					
选择数量 (CR 用于选择, 空格用于移动)	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">500 克</td></tr> <tr><td style="text-align: center;">1000 克</td></tr> <tr><td style="text-align: center;">1500 克</td></tr> <tr><td style="text-align: center;">2000 克</td></tr> </table>	500 克	1000 克	1500 克	2000 克
500 克					
1000 克					
1500 克					
2000 克					
本次订购	输入另一项				
500 克 西红柿	(输入0结束)				
1500 克 土豆					

进行菜单项显示设计时，应按照执行逻辑将内容组织在一起，以便引导用户找到所需要的内容。

🕒 关键词数据输入

关键词数据输入比菜单选择数据输入更快速、更有效。并可以以不同的顺序输入，允许更复杂的文件输入。例如，在绘图系统中，利用**关键词**line、**brok**、**rect**和**circ**作为画直线、折线、矩形和圆的助记符，进行识别和操作。

🕒 光学标记 / 识别 (OMR)

光学标记 / 识别在表格中使用。用户在表格的一个区域中打标记□或■，然后让表格通过一个光敏读入设备，其中用暗标记■表示“是”，用亮标记□（即未标记过）表示“否”。

🕒 光学字符识别 (OCR)

OCR系统可让计算机通过模式比较来识别一些具有不同字体和大小的印刷体。首先它让字符识别系统熟悉铅字字体的特征。经过若干次尝试，使计算机系统了解这种字体的规则，并将这些规则记忆到模式匹配算法中。

🕒 磁性墨水字符识别 (MICR)

MICR字体就是在银行支票上的帐号和分类号所使用的字符。

🕒 条形码 (Bar Code)

条形码由许多粗细不等的竖线组成的标签，这些竖线条在特定位置上出现或不出现就表示某个特定的数据。条形码的代码由一个特殊的光敏装置或条形码读入器读入，读入器在横穿过条形码时挑选出暗带，并根据暗带在位置 x ， $x+1$ 等处是否出现而将条形码序列翻译成数据，计算机将条形码与检查相比较以计算出商品的号码或数值。

🕒 声音数据输入

声音数据输入有许多很明显的优点。它输入速度很快，可用于不宜使用纸张及不能使用键盘的场合。这种方式不需要书写，只需用户使用自己的声音器官发声即可。**声音数据的输入包括了语音和自然语言对话的所有问题。**在目前的技术条件下，在限定200~8000个词汇量的范围内，进行有限的单个关键词的输入是可能的。现在已经有一种声控打字机，它拥有的词汇量达到8000，已能满足人们日常会话所使用用语的词汇量。



数据显示界面设计

数据显示界面包括屏幕查询、文件浏览、图形显示和报告。

数据显示的规则

- 进行数据输出显示设计，应当了解数据显示的要求，解决应该显示哪些数据，屏幕上一次显示多少信息的问题。显示的信息对于用户任务来说应当是适当的，不要过于拥挤。
- 选择显示内容，应当考虑以下准则。
 - ☆ 只显示必需的数据。与用户需求无直接关系的一律省略。

🕒 在一起使用的数据应显示在一起。

🕒 显示出的数据应与用户执行的任务有关。

🕒 每一屏数据的数量，包括标题，栏题等等，不应超过整个屏幕面积的30%。

- 利用这些规则，并根据用户要求，下一步应当将数据分组，然后将每组数据按一定的结构形式来安排，总的目的是使得用户感到使用方便。
- 显示设计要使得相关的数据成组地出现，并由用户与系统的会话来控制。
- 可根据屏幕的大小，使每帧屏幕包含若干个子区域，让每个子区域显示不同的信息。

- 进行屏幕布局时，还需要考虑其它一些规则。
 - ☆ 应尽量少使用代码和缩写，不应让读者去翻译或猜测这些代码或缩写。
 - 🕒 如果安排了若干个显示画面，最好建立一个统一的格式。
 - 🕒 提供明了的标题、栏题以及其它提示信息。帮助用户浏览各种显示画面。
 - 🕒 遵循用户的习惯。采用在分析过程中得到的用户模型，并保留用户使用的术语。
 - 🕒 采用颜色、字符大小、下划线或不同的字体等方式来强化重要数据。

- 设计了数据的显示结构之后，根据是图形显示还是字符显示，进一步考虑细节设计。

字符数据的显示

- 字符数据的画面显示主要是屏幕布置和数据内容安排格式，以便于用户查找和阅读的问题。

☆ 纯正文的显示

英文正文中应避免连续使用大写字母，大写字母应使用印刷体，且一般为强调而使用。英文正文应当是左边顶格，右边可以参差不齐。如果要求左、右两边都顶格，参差不齐的间隔容易分散视力。

🕒 列表和表格

数据列表应当竖排而不应横排，因为这样有助于计算总和。栏题应当安排在数据列之上。

姓 名	数学成绩	英语成绩	物理成绩	化学成绩
卢晓波	91	83	88	94
浦柯玖	90	91	95	87
张超回	94	86	92	89

显示不应是一个固定的格式。一般来说，显示的内容应由用户控制。

🕒 控制显示

用户应当拥有一种获得不同显示画面的灵活的手段。数据显示对话应允许用户在无需进入数据检索状态下就能实现换屏和上下滚动显示。在换屏控制显示时，前一次显示的部分内容应显示在新显示画面的顶部或底部，以使用户感到连贯。在画面滚动显示时，其速度应由用户控制，使得不感兴趣的内容快速跳过而感兴趣的内容缓慢通过，以便查看。

图形显示

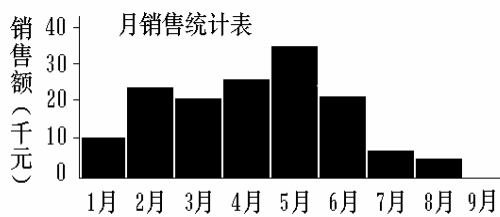
- 由于图形从数据集中概括出某些特性并且具有“直观”的优点，因此对于识别和分析处理结果更有效。为了做好图形显示，必须仔细地选择图形类型和进行布局设计。

☆ 图形类型与数据集分类

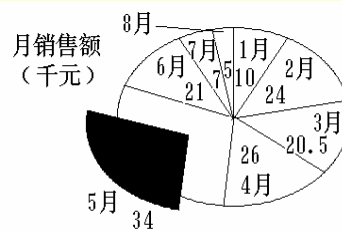
在一定程度上，图形类型的选择是有限的，因为它根据数据类型来决定的。图形的数据集来自三个方面，其数值可以是：顺序的（布尔值，即有/无）；标称的（整数）；十进制的（实数）。数据集可以按标绘图类型来分类。

常用的二维图形

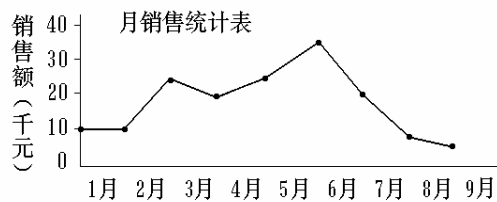
在商用的图形库中，常取下列4种图形。



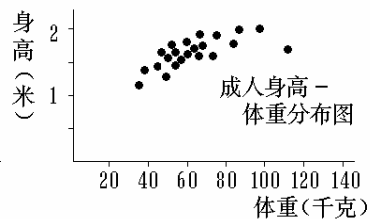
(a) 直方图



(b) 饼图



(c) 折线图

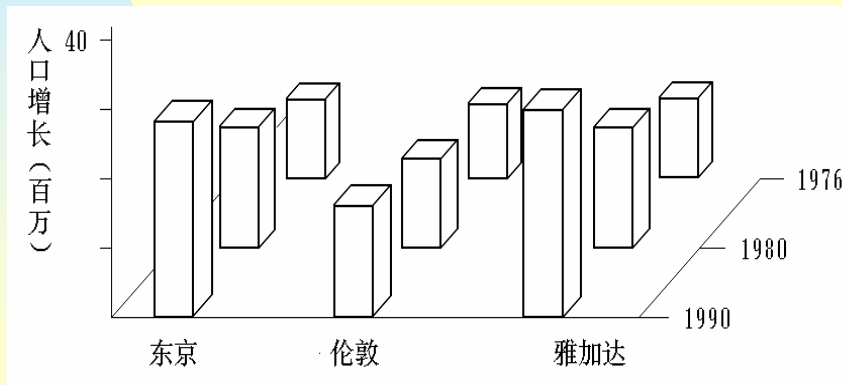


(d) 散布图

🕒 三维图形

当每个对象有三个测量值 (x, y, z) 时, 可用三维图形方法。

三维直方图, 三维饼图, 三维线图 etc



报告

■ 报告的功能和分类

报告是输出字符的一个子集, 它的功能可以从一个系统向另一个系统传递信息; 也可以是某一系统的总结; 还可以是一个历史文献、简单的列表等等。报告可以归为四类。

☆ **文宗报告**: 报告中输出的是数据处理的结果。它主要用于在各个系统之间传递信息, 包括正在处理的对象的信息。例如, 订货单、发货单、发票、购物单、帐单等。

- 🕒 **信息报告：**报告中包含的系统信息有描述系统进程和活动的的数据。这些信息主要是系统管理人员用来监测、控制和修改系统的行为。例如，异常报告、监控和分析报告、管理总结报告等。
- 🕒 **历史和档案报告：**此类报告所载的信息记录了系统在某一时刻的状况，或者是记录系统的历史。今后或许还会用到的已处理过的数据也存在档案报告中。
- 🕒 **浏览报告：**这相当于显示查询结果的屏幕或文件列表。信息一般用比较简单的格式表示。用户可以用各种方法使用它。

■ 报告的用途分析

报告中的内容应当在系统的输出数据流和用户要求中指明。但是，在报告中数据如何分组，还受到下面一些因素的影响。

- ◆ **提出报告的频度：**有随机提交的报告、定期提交的报告。各种报告所需信息不尽相同。因此，时效性的需求要求考虑信息应当存放在那一种报告中。
- ◆ **报告的打印量：**需求量不同的报告应分别进行处理。

- ◆ **信息的时效性和精确性：**要求信息的及时程度以及陈旧信息对用户的使用价值，都会影响提出报告的时间。对于数值数据，还要考虑它的精度。
- ◆ **保密：**信息的机密程度以及需要采取什么预防措施以保证数据不被未经授权的人存取，这一点与打印报告的设备及分发报告的管理有关。

布局设计

- 报告有三种布局设计。
 - ◆ 列表适用于记录信息的简单罗列，浏览报告和档案报告即属于这一类；
 - ◆ 块结构是用行、列和总计等形式来组织数据，信息报告就属于这一类；
 - ◆ 分组结构则在块内进一步做更复杂的成组的信息布局，文宗报告一般属于此类。
- 报告设计的步骤可以应用前述的屏幕显示设计的许多准则。

🕒 列表报告

- ◆ 数据通常以记录格式给出，数据按记录规定格式成行打印。
- ◆ 各页应加页号。
- ◆ 如果列表中的数据以某种顺序排列或分类，在各分类的组间应加空行，使结构更清楚。
- ◆ 数据域应分列安排并加上栏题。

🕒 块结构报告

- ◆ 在报告中信息分块，各块按设计要求顺序排列。

- ◆ 可变项应安排在报告右边的列中，以免显得凌乱。
- ◆ 信息块应当用空格分开，而不要用分隔符。
- ◆ 在信息块顶部的左侧应加上题头，为信息块做标记。
- ◆ 若有总计数字，必须紧跟在与其有关的数据后面安排。如果有多层次的合计数字，需要增加总结页，说明合计数字间的层次和所属关系。

🕒 组结构报告

可以利用通常的报告设计的规则来设计组的内容和数据的布局顺序。

- ◆ 信息组应以空格分隔开，或者是用方框或背景色来分隔，应避免使用过多的分隔符。
- ◆ 报告的标题应居中安排。特别对于文宗报告，在报告的右上角应使用清晰而唯一的代码对报告进行标识或索引。
- 具体布局
布局的规则类似于屏幕显示中的规则。再加一些参考规定：
 - ◆ 应检查数据内容的类型和格式，以确定它所需的打印字符的数目。

- ◆ 字符左侧对齐，数字右侧对齐，有小数点则对齐小数点。负号放在前面会使数列变得不规整，可把负号放在后面。
- ◆ 各列之间的间隔至少用三个空格。
- ◆ 标题应安排在列的正中。
- ◆ 可用粗体字、不同的字体、下划线或不同的色彩等来强调重要的部分。
- ◆ 给每页加页码和名称。
- ◆ 每次要给报告加上日期和时间。
- ◆ 当一切细节设计好之后，不能忘记征求用户的意见。需要同用户磋商，交换意见，然后做出修改以满足其要求。



控制界面的设计

设计控制界面的主要目的是让用户能够主动地控制计算机上软件系统的工作，使得用户能够很容易地访问计算机的各种设备。其主要方式有控制对话、菜单、功能键、图标、直接指点、窗口、命令语言和自然语言等。

用控制对话选择操作命令

- 控制对话可以是简单的问答形式，系统提出是否需要某个操作，然后用户以Y/N的方式回答。

- 复杂的对话形式是基于菜单的系统。这种对话方式容易使用，但每次的操作都是单调重复的，老的用户往往会感到冗长乏味。
 - ◆ 在设计对话的时候，需要注意的要点是：
 - ◆ 每次只能有一个提问，避免多个询问；
 - ◆ 当需要几个关联的回答时，应重新显示上一个回答。如果以前的回答在后面还需要用到时，在用到的时候要重新显示它，否则会因短期记忆出现错误。
 - ◆ 保持提问的顺序与原文档或用户模型一致。

用菜单界面进行控制

- 通常用户在菜单条件下做出应答有两种方式：
 - ◆ 使用应答码在屏幕显示的菜单做出选择。应答码可以是数字，也可以是字符。字符码通常应当能够记忆并且能够代表这个选择的实际含义，例如，利用“F”表示对磁盘格式化。但有时找不到合适的字符来表示选择，例如，用“E”表示“edit”，还是用“E”表示“exit”。因此需要使用较长的字符串。
 - ◆ 使用鼠标器按钮，或者用周转选择法，即用户用移位键，逐行地把高亮度的菜单选择条往下拉，从菜单的底部还可回到顶部，或者逐行地把高亮度的菜单选择条往上拉，到了菜单的顶部又可回到底部，然后，按回车键以确定当前选定的菜单选择条所代表的操作。
 - ◆ 在多数系统中，采用了多级菜单结构。这种结构必须使可选项的组织与用户的模型一致，以及把系统中的功能和可选项正确地分组。

◆ 对于多级菜单的深度和宽度，需要权衡。

- ☞ 在一个菜单中放置很多的选择项，将使多级菜单加宽，使用户在菜单中做选择时需要较多的查询时间，但菜单的层次就会少一些。
- ☞ 如果多级菜单设置很多级别，会使层次加深，而每个菜单中选择项变少，这样每级菜单的查询时间会变短，但在菜单上巡航的时间会增加。
- ☞ 经验表明，在一级菜单中包含7~9个选择项最为适宜。

- 对于一个小的系统，一般采用宽菜单比较好。例如，Word Star的菜单就是一个例子。因为它不需要搜索层次的时间。但是对于一个大的系统，需要一个明显的层次结构，以帮助用户了解系统。这时，采用多级菜单的形式，对于一个大的系统是合适的。
- 对于多级菜单，一些有经验的用户不希望每次都按层次上下，而要求能够从一个选择转换到另一个选择。为此，需要设计一个菜单旁路工具，以便直接访问。

C:\TEST.DOC		P01 L13 C01 Insert Align		
EDIT MENU				
CURSOR	SCROLL	ERASE	OTHER	MENUS
^E up	^W up	^G char	^J help	^O onscreen format
^X down	^Z down	^T word	^I tab	^K block & save
^S left	^R up screen	^Y line	^V turn insert off	^P print controls
^D right	^C down	DEL char	^B align paragraph	^Q quick function
^A word left	screen	^U unerase	^N split the line	ESC shorthand
^F word right			^L find/replace again	

L-----!-----!-----!-----!-----!-----!-----!-----!-----!-----!-----R

常规的用户界面设计的步骤如下：

- (1) 标识输入文件；
- (2) 设计输入文件上抽象的数据类型，数据结构和表格形式；
- (3) 设计由输入文件转换成软件内部数据结构的算法；
- (4) 设计装入机器初始化的操作；
- (5) 对批处理操作，设计启动点，检查点及重新启动点；对交互式操作，实施进程分割，设计每一进程正确运行的辨认格式及选择下一进程的准则，中间输入的描述格式机启动命令。
- (6) 设计输出中间结果和最终结果的图表和正文的格式，并确认输出设备。
- (7) 以用户欢迎的方式和他们易于理解的语言编写"用户使用手册(初稿)"。

Display	Center	ChkRest	ChkWord	Del Blk	HideBlk	MoveBlk	CopyBlk	BegBlk	EndB
1help	2Undo	3Undrlin	4Bold	5Delline	6DelWord	7Align	8Ruler	9Save &	0Done

■ 菜单设计的准则如下：

- ◆ 按逻辑相关性把选择项分组，组成菜单块或单独的菜单屏；
- ◆ 根据通常标准，例如操作顺序、使用频度、重要程度等确定菜单的次序；
- ◆ 指明所期望的应答，并使之与选择项相联系；
- ◆ 根据菜单及菜单选择项的功能来命名；
- ◆ 提供给用户关于菜单级别、错误等的反馈信息；
- ◆ 提供退出路径及旁路机制；





- ◆ 防止错误的应答。例如，若1~7是选择，0是退出，就应当使一旦按下其它的键时，能够得到出错的信息而不导致系统的失败。

用功能键定义操作命令

- 功能键是与选择菜单等效的硬件。通过定义和使用键盘上特定的键来选择可以节省屏幕空间。功能键可以用硬编码，也可以用软编码。硬编码是将功能键的操作固定到某个特殊键。对于像文字处理机这种功能不变的专用硬件来说，这种硬编码非常有效。

- 对大多数系统，功能键都是软编码的。用软编码时，命令调用可通过应用程序分配到各个功能键上，每个键可以对应一个或多个命令。如果多个命令对应单个键，用户必须随时跟踪系统所处的状态。
- 例如，键F2在某种情况下是删除一个字，而在另一情况下是存入一个文件。为了帮助用户，需要在屏幕上显示一个副菜单，在这个菜单上显示了选择码的分配，以及在屏幕上的键盘布局图像。
- 大多数计算机硬件提供了10~12个功能键。

用图标表示对象或命令

- 为了使得用户能够识别一个图符所表示的物体或命令，图标应当是逼真的，以至于用户不用专门学习就可对它所代表的意义立即做出有据的判断。
- 例如，用“”表示擦图操作，用“”表示图形拖曳操作，用“”表示喷涂操作，等等。
- 有时图标可能会有多义性，对于同一个图标，不同的人有不同的解释。例如，图标“”有人理解为计算器，有人理解为传真机。为了防止多义性，在图标下面加文字解释。

设计图标时的几点建议

- 让用户来确认图标的含义；
- 使得图标尽可能逼真；
- 图标应有一个清晰的轮廓，以利于辨认；
- 当显示命令时，要给出在此命令下操作对象的具体表象；
- 要避免使用符号，除非其意义十分明确。

直接操纵

- 这种界面的主要思想是想让用户能够观看并直接操纵系统中的对象。它包括图标、指点，以及与**WIMP** (**窗口、图标、鼠标、上托式菜单**)有关的特性。
- 对象用图标表示，并且可通过鼠标或其它类似的光标控制器(如光笔、数字化仪等)的指点来进行寻址及选择对象。然后，依据指点和选择调用一个系统操作。
- 例如，用拖曳操作，可使选中的对象在屏幕上移动，如把一个文件放在文件夹里，或把一条信文放在邮箱里。

直接操纵界面的基本特性

- **明确的动作**：用户在屏幕上指点并操纵对象。
- **即时的反馈**：用户操作的结果立即可见。例如，当用户选择一个图标时，它即呈高亮度。
- **增量效应**：用户的动作应当有一个模拟/顺序的尺度。例如，当一个图标被拖曳着在屏幕上移动时，应当随着用户鼠标的移动而连续地移动，而不应突然跳到一个新的位置。
- **直观的交互作用**：交互作用应当与有关如何操作系统的用户概念模型相匹配，并且能够显示对象的直观图像。

- **剥皮式的学习**：当用户学习系统功能时，其复杂性应按层逐渐增加。
- **可逆的动作**：使用退回操作，可以顺序复原到原先的动作状态。
- **事先验证**：只允许合法的交互产生效果，如果用户指向一个对象而交互动作对现行任务没有意义时，显示屏幕上应无反应。

用窗口划分屏幕

直接操纵有时要求能够同时具有许多不同的界面，或者对于同一对象有多个视图，这就需要窗口的支持。

- 窗口把物理屏幕划分成几部分，在屏幕上同时可以进行不同的操作。
- 存在两种类型的窗口：
 - ◆ **砖状的窗口**：即把整个物理屏幕规则地划分成几个子屏幕，并且不重叠；
 - ◆ **重叠的窗口**：即把窗口依次嵌套叠在别的窗口上方，好似层层深入一样。
- 如果在一段对话中间需要一段子对话，就可以打开控制窗口，使得用户可以在不同的窗口中运行两个或多个进程。窗口以这种方式允许多任务处理进入“挂起/继续”状态。

- 在办公室的环境中，需要并行地处理多个任务，窗口就很适合于这种工作。
- 窗口还在监控信息方面有用，在窗口上可以保持背景或挂起任务的状态，故而能够周期地监控任务的进展情况。

以下的建议，对于窗口的使用是有益的。

- ◆ 对于初学者来说，简单的砖状的窗口就足够用的了，重叠的窗口将增加不必要的复杂性。
- ◆ 利用窗口可进行任务切换(如从编辑到任务管理又再次返回)，但要使任务数量尽量少。

- ◆ 避免在不活动的窗口上经常改变图像。因为这会使得人对正在处理的作业的注意力分散。
- ◆ 要删去与现行任务不直接有关的旧窗口，否则，旧窗口会使得屏幕杂乱无章。
- 窗口和直接操纵界面需要先进的接口软件去控制屏幕的显示和高分辨率的显示器。该软件的作用是充当应用软件和用户之间的翻译，并且管理所有的交互作用和通信。具有这种特性的接口软件叫做“用户接口管理系统”。

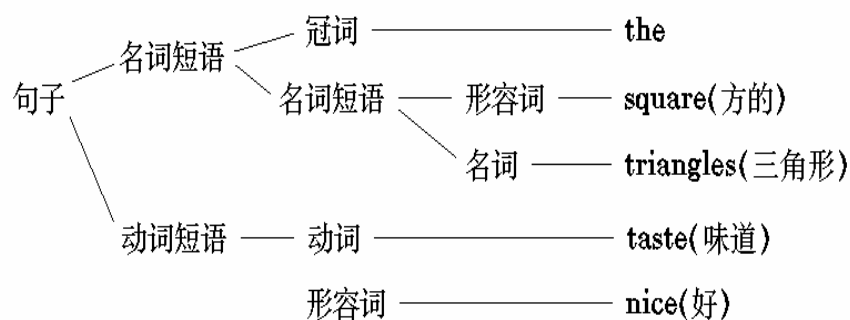
命令语言

- 命令语言是潜在的最强有力的控制界面。其主要优点是：可节省屏幕空间；可通过名字对目标和功能直接使用(从而不必提供存取层次)。
- 命令的组合可以使系统功能更灵活。
- 所有的命令语言都有一个词典和一个语法。词典给定单词的集合，语法给出说明单词组合方式的规则。

自然语言

- 像命令语言一样，自然语言由词典和语法组成。但与命令语言和程序设计语言不同的是，自然语言的语法成分有许多规则，使其具有更灵活的表达式和多义性的解释。
- **语法**
自然语言是由词(词汇)构成的，词可划分成名词、动词、形容词等等。语法规则规定了如何把各种词类联接起来构成一个完整的句子。如在英语中的句子：“He must go to the station to catch the train” (他必须去车站赶火车)。

- ◆ 句子可利用语法分析方法来分解，以检查句子中词的排列顺序是否符合允许的组合规则。常用语法分析树来表示句子的语法结构。
- ◆ 语法只能告诉人们一个句子是否符合语言的语法规则，要理解句子的意思，必须考虑句子的语义。如“the square triangle staste nice” (方的三角形味道好) 这句话，语法上看是正确的，但语义上看，明显地是荒谬的。



■ 语义

语义是指从词与词的关联知识中产生的含义。它形成了语言、记忆和经验的联系。语义规则可放入语法中，以便把无意义的句子删去。

■ 4. 多媒体用户界面设计

- 多媒体界面设计的特点;
- 虚拟现实: 多感知性\存在感\交互性\自主性;
- 多通道人机交互技术:
 - ◆ 非精确交互
 - ◆ 多通道交互

利用**CASE**工具辅助设计