2007 最新版

高等学校计算机科学与技术教材

- 原理与技术的完美结合
- 教学与科研的最新成果
- 语言简练，实例经典
- 操作性强，实用性突出

GONGGONGJICHU

全国计算机等级考试二级教程

[公共基础]

专家导学 一点即通

主　编　龚乾春
副主编　甘功宇　李　松

UESTCP 电子科技大学出版社

GONGGONGJICHU

全国计算机等级考试二级教程

[公共基础]

高等学校计算机科学与技术教材

全国计算机等级考试二级教程

[公共基础]

主 编 龚乾春

副主编 甘功宇 李 松

电子科技大学出版社

图书在版编目（CIP）数据

全国计算机等级考试二级教程．公共基础/龚乾春主编．
—成都：电子科技大学出版社，2007.3
ISBN 978-7-81114-408-6

Ⅰ．全… Ⅱ.龚… Ⅲ.电子计算机-水平考试-教材
Ⅳ.TP3

中国版本图书馆 CIP 数据核字（2007）第 020290 号

内 容 提 要

本书是为满足人们对计算机基础知识的学习，能迅速、深入地理解和掌握等级考试的相关内容，我们按 2006 年版大纲要求编写了本书。

本书主要介绍程序设计方法与风格，结构化程序设计，面向对象的程序设计方法、对象、属性及继承与多态性；算法的基本概念，数据结构的基本概念及其定义，栈和队列及其基本运算，线性链表及其基本运算，二叉树的基本概念；软件工程的基本概念，结构化分析方法，结构化设计方法，程序的调试方法；数据库管理系统，数据模型，关系代数理论中的基本运算，数据库设计的基本方法和步骤等内容。

每章后面都附有一定数量的习题，以帮助读者巩固所学的知识点。

本书特别适合作为全国计算机等级考试（二级）公共基础的教材和辅导资料，同样也可作为其他计算机基础知识的教学用书。

高等学校计算机科学与技术教材

全国计算机等级考试二级教程

[公共基础]

主编 龚乾春 副主编 甘功宇 李松

出　　版：电子科技大学出版社 (成都市一环路东一段 159 号电子信息产业大厦　邮编：610051)
策划编辑：张　俊
责任编辑：汤云辉
主　　页：www.uestcp.com.cn
电子邮箱：uestcp@uestcp.com.cn
发　　行：新华书店经销
印　　刷：四川省地矿局测绘队印刷厂
成品尺寸：185mm×260mm　　**印张**　8.25　　**字数**　224 千字
版　　次：2007 年 3 月第一版
印　　次：2007 年 3 月第一次印刷
书　　号：ISBN 978-7-81114-408-6
定　　价：18.00 元

前　言

随着社会信息化程度的不断提高，社会各行各业都大量需要掌握计算机硬件和软件技术的人才。由国家教育部考试中心推出的全国计算机等级考试，其测评对象为全社会的非计算机专业人员，为培养各行各业的计算机应用人才开辟了一条新的道路，所以受到社会的欢迎。为适应信息技术的发展，教育部考试中心颁发了2006年版等级考试大纲。为满足人们对计算机基础知识的学习，能迅速、深入地理解和掌握等级考试的相关内容，我们按2006年版大纲要求编写了本书。

本书共分四章。第一章主要介绍程序设计方法与风格，结构化程序设计，面向对象的程序设计方法、对象、属性及继承与多态性；第二章主要介绍算法的基本概念，数据结构的基本概念及其定义，线性表及其基本运算，栈和队列及其基本运算，线性链表及其基本运算，二叉树的基本概念、存储结构及其遍历，最后还介绍了几种常用的查找与排序算法；第三章主要介绍软件工程基本概念，结构化分析方法，结构化设计方法，软件测试的基本方法，程序的调试方法；第四章主要介绍数据库，数据库管理系统，数据库系统的基本概念，数据模型，实体联系模型及E-R图等基本概念，关系代数理论中的基本运算，数据库设计的基本方法和步骤。

本书在编写上力求叙述简要，语言通俗易懂。既做到知识的科学、系统，又突出概念。每章后面都附有一定数量的习题，以帮助读者巩固所学的知识点。

本书特别适合作为全国计算机等级考试（二级）公共基础的教材和辅导资料，同样也可作为其他计算机基础知识的教学用书。

全书由龚乾春、甘功宇、李松统稿，审定。同时在编写的过程中得到温洪宇、江龙、勾文波、陈仕波、邓招富等同志的大力帮助，得到君思贤学校许多老师的支持，编者对上述同志表示衷心的感谢。由于时间紧迫，以及作者水平有限，书中难免存有不妥之处，恳请读者批评指正。

编　者

2007年3月

目 录

第一章　程序设计基础理论篇

1.1　程序设计的方法和风格

程序设计是使用计算机系统的指令或语句，组成求解不同问题，实现不同算法所需的完整序列的一个工作过程。

程序设计的最初始阶段是讲究技巧的年代。如何能节省一个字节，如何能提高程序运行的效率，这些都是要严肃考虑的问题。而所谓程序的易读性、可维护性根本不在考虑范围之内。

随着计算机销售价格的不断下降，硬件环境的不断改善，运行速度的不断提升，程序的越写越大，功能的越来越强，讲究技巧的程序设计方法已经不能适应需求。而程序的易读性和可维护性在程序设计时就不得不在考虑范围之内。

一般来讲，程序设计风格是指编写程序时所表现出的特点、习惯和逻辑思路。程序是由人来编写的，为了测试和维护程序，往往还需要阅读和跟踪程序，因此程序设计的风格总体而言应该强调简单和清晰，程序必须是可以理解的。由此可见著名的“清晰第一，效率第二”的论点已成为当今程序设计风格的主导。

1.1.1　程序设计的方法

1．结构化设计方法

（1）要求把程序的结构规定为顺序、选择和循环三种基本结构，并提出了自顶向下、逐步求精、模块化程序设计等原则。

（2）结构化程序设计是把模块分割方法作为对大型系统进行分析的手段，使其最终转化为三种基本结构，其目的是为了解决由许多人共同开发大型软件时，如何高效率地完成可靠系统的问题。

（3）程序的可读性好、可维护性好，这将成为评价程序质量的首要条件。

2．快速原型方法

利用现有的工具和原型方法快速地开发所要的程序。

3．面向对象程序设计方法

（1）对系统的复杂性进行概括、抽象和分类，使软件的设计与现实形成一个由抽象到具体、由简单到复杂的循序渐进过程，从而解决在大型软件研制中存在的效率低、质量难以保证、调试复杂、维护困难等问题。

（2）结构化的分解突出过程，即如何做（How to do）的问题，它强调代码的功能是如何实现的；面向对象的分解突出现实世界和抽象的对象，即做什么（What to do）的问题。

1.1.2 程序设计的风格

要形成良好的程序设计风格，应注意以下内容：

1．源程序形象化

源程序形象化应注意如下几点：

（1）变量的命名：变量的命名应具有一定的含义，以便于对程序功能进行说明；

（2）增加注释：增加注释能够帮助用户理解程序。注释可以分为序言性注释和功能性注释；

（3）程序的视觉组织：为使程序的结构一目了然，可以在程序中利用空格、空行、缩进等技巧使程序层次清晰。

2．数据说明

对程序的数据说明应注意以下几点：

（1）正确具体地说明一切变量；

（2）数据说明的次序应该规范化；

（3）便于查找变量（按顺序排列）；

（4）对复杂的数据结构应注释说明。

3．语句的结构

要提高程序的易读性和可维护性应注意以下几点：

（1）每条语句简单明了；

（2）尽量不用或少用 GO TO 语句；

（3）尽量只采用 3 种基本控制结构编程。

4．输入和输出

输入和输出信息是用户直接关心的，输入和输出在方式和格式上应尽可能方便用户的使用，因为系统能否被用户接受，往往取决于输入和输出的风格。无论是批处理的输入和输出方式，还是交互式的输入和输出方式，在设计和编程时都应该注意以下原则：

（1）对所有输入数据进行校验和合理性检查；

（2）输入、输出格式保持一致；

（3）设计良好的输出报表。

1.2 结构化程序设计

荷兰学者 Dijkstra 提出了“结构化程序设计”的思想，它规定了一套方法，使程序具有合理的结构，以保证和验证程序的正确性，这种方法要求程序设计者不能随心所欲地编写程序，而要按照一定的结构形式来设计和编写程序，它的一个重要目的是使程序具有良好的结构，使程序易于设计、易于理解、易于调试和修改，以提高设计和维护程序工作的效率。

1.2.1　结构化程序设计的主要原则

结构化程序设计方法的主要原则可以概括为自顶向下、逐步求精、模块化和限制使用 GO TO 语句。

◆　自顶向下：程序设计时，应先考虑总体，后考虑细节；先考虑全局目标，后考虑局部目标。不要一开始就过多追求众多的细节，先从最上层总目标开始设计，逐步使问题具体化。

◆　逐步求精：对复杂问题，应设计一些子目标做过渡，逐步细化。

◆　模块化：一个复杂问题，应由若干简单的问题构成。模块化是把程序要解决的总目标分解为分目标，再进一步分解为具体的小目标，把每个小目标称为一个模块。

◆　限制使用 GO TO 语句

实际上，结构化程序设计方法的起源来自对 GO TO 语句的认识和争论。肯定的结论是，在块和进程的非正常出口处往往需要用 GO TO 语句，使用 GO TO 语句会使程序执行效率较高；在合成程序目标时，GO TO 语句往往是有用的，如返回语句用 GO TO。否定的结论是，GO TO 语句是有害的，是造成程序混乱的祸根，程序的质量与 GO TO 语句的数量成反比，应该在所有高级程序设计语言中取消 GO TO 语句。取消 GO TO 语句后，程序易理解、易排错、易维护，程序容易进行正确性证明。作为有争议的结论，1974 年 Knuth 发表了令人信服的总结，并证实了：

（1）滥用 GO TO 语句确实有害，应尽量避免；

（2）完全避免使用 GO TO 语句也并非是个明智的方法，在有些地方使用 GO TO 语句，会使程序流程更清楚、效率更高；

（3）争论的焦点不应该放在是否取消 GO TO 语句，而应该放在用在什么样的程序结构上。其中最关键的是，肯定了以提高程序清晰性为目标的结构化方法。

1.2.2　结构化程序的基本结构与特点

对于大型的程序设计，使用一些基本的结构来设计程序，无论多复杂的程序，都可以使用这些基本结构按一定的顺序组合起来。这些基本结构的特点都是只有一个入口、一个出口。由这些基本结构组成的程序就避免了任意转移、阅读起来需要来回寻找的问题。结构化程序设计可以分为以下三种结构：

1．顺序结构

顺序结构是一种简单的程序设计，它是最基本、最常用的结构，如图 1-1 所示。顺序结构是按照程序语句行的自然顺序，一条语句一条语句地执行。

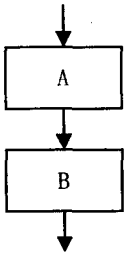

图 1-1　顺序结构

2．选择结构

选择结构又称为分支结构，它包括简单选择结构和多分支选择结构，选择结构可以根据设定的条件，判断应该选择哪一条分支来执行相应的语句序列。如图 1-2 所示列出了包含 2 个分支的简单选择结构。

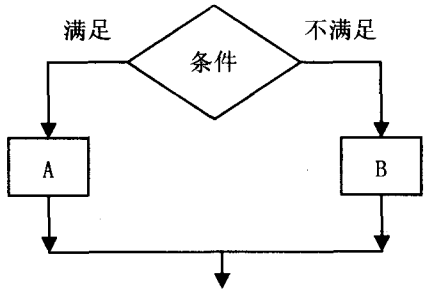

图 1-2　选择结构

3．重复结构

重复结构又称为循环结构，它根据给定的条件，判断是否需要重复执行某一相同的或类似的程序段，利用重复结构可简化大量的程序行。在程序设计语言中，重复结构对应两类循环语句：一是对先判断后执行循环体的称为当型循环结构，如图 1-3 所示；二是对先执行循环体后判断的称为直到型循环结构，如图 1-4 所示。

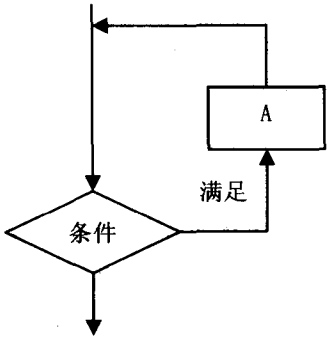

图 1-3　当型循环结构

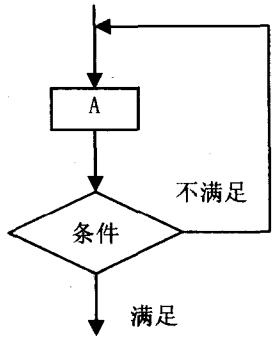

图 1-4　直到型循环结构

三种基本结构都具有以下特点：

（1）有一个入口；

（2）有一个出口；

（3）结构中每一部分都应当有被执行到的机会，也就是说，每一部分都应当有一条从入口到出口的路径通过它（至少通过一次）；

（4）没有死循环（无终止的循环）。

结构化程序要求每一基本结构具有单入口和单出口的性质是十分重要的，这是为了便于保证和验证程序的正确性。设计程序时按结构顺序依次写下来，整个程序结构顺序清楚，层次分明。在需要修改程序时，可以将某一基本结构单独进行修改，由于单入口、单出口的性质，不会影响到其他的基本结构。

1.2.3　结构化程序设计原则与方法

基于对结构化程序设计原则、方法以及结构化程序基本构成的掌握和了解，在结构化程序设计的具体实施中，要注意把握以下几点：

1. 使用程序设计语言中的顺序、选择、循环等有限的控制结构表示程序的控制逻辑；
2. 选用的控制结构只准许有一个入口和一个出口；
3. 程序语句组成容易识别的块，每块只有一个入口和一个出口；
4. 复杂结构应该用嵌套的基本控制结构进行组合嵌套来实现；
5. 语言中所没有的控制结构，应该采用前后一致的方法来模拟；
6. 严格控制 GO TO 语句的使用。其意义是指：

（1）用一个非结构化的程序设计语言去实现一个结构化的构造；

（2）若不使用 GO TO 语句会使功能模糊；

（3）在某种可以改善而不损害程序可读性的情况下使用。

1.3　面向对象的程序设计

关于面向对象方法，对其概念有许多不同的看法和定义，但都涵盖对象及对象属性与方法、类、继承、多态性几个基本要素。下面分别介绍面向对象方法中这几个重要的基本概念，这些概念是理解和使用面向对象方法的基础和关键。

1.3.1　对象（Object）

对象是面向对象方法中最基本的概念。对象是基本运行时的实体，它既包括数据（属性）也包括作用于数据的操作（行为）。例如，一头牛、一台电脑等，都可以作为一个对象。

一个对象把属性和行为封装为一个整体，通常可由对象名、属性和操作三部分组成。比如，一台电脑是一个对象，它包含了电脑的属性如配置、价格等。

对象有以下一些基本特点：

（1）标识唯一性。指对象是可区分的，并且由对象的内在本质来区分，而不是通过描述来区分。

（2）分类性。指可以将具有相同属性和操作的对象抽象成类。

（3）多态性。指同一个操作可以是不同对象的行为。

（4）封装性。从外面看只能看到对象的外部特性，即只需知道数据的取值范围和可以对该数据施加的操作，根本无需知道数据的具体结构以及实现操作的算法。对象的内部，即处理能力的实行和内部状态，对外是不可见的。从外面不能直接使用对象的处理能力，也不能直接修改其内部状态，对象的内部状态只能由其自身改变。

（5）模块独立性。对象是面向对象软件的基本模块，它是由数据及可以对这些数据施加的操作所组成的统一体，而且对象是以数据为中心，操作围绕对其数据所需做的处理来设置，没有

无关的操作。从模块的独立性考虑，对象内部各种元素彼此结合得很紧密，内聚性强。

1.3.2 类（Class）和实例（Instance）

类定义了一组大体上相似的对象，也就是说，类是具有共同属性、共同方法的对象的集合。所以，一个类所包含的方法和数据描述了一组对象的共同行为和属性。类是在对象之上的抽象，对象是类的具体化，是类的实例。

需要注意的是，当使用“对象”时，既可以指一个具体的对象，也可以泛指一般的对象，但是，当使用“实例”时，必然是指一个具体的对象。

例如，Integer 是一个整数类，它描述了所有整数的性质。因此任何整数都是整数类的对象，而一个具体的整数“111”是类 Integer 的一个实例。

1.3.3 消息（Message）

消息是对象之间进行通信的一种构造。消息是一个实例与另一个实例之间传递的信息，它请求对象执行某一处或回答某一要求的信息，它统一了数据流和控制流。消息的使用类似于函数调用，消息中指定了某一个实例、一个操作名和一个参数表（可空）。接收消息的实例执行消息中指定的操作，并将形式参数与参数表中相应的值结合起来。消息传递过程中，由发送消息的对象（发送对象）的触发操作产生输出结果，作为消息传送至接受消息的对象（接受对象），引发接受消息的对象一系列的操作。所传送的消息实质上是接受对象所具有的操作方法名称，有时还包括相应的参数，如图 1-5 所示表示了消息传递的概念。

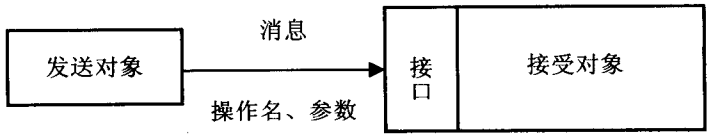

图 1-5 消息传递示意图

消息中只包含传递者的要求，它告诉接受者需要做哪些处理，但并不指示接受者应该怎样完成这些处理。消息完全由接受者解释，接受者独立决定采用什么方式完成所需的处理，发送者对接受者不起任何控制作用。一个对象能够接受不同形式、不同内容的多个消息；相同形式的消息可以送往不同的对象，不同的对象对于形式相同的消息可以有不同的解释，能够做出不同的反映。一个对象可以同时与多个对象传递信息，两个对象也可以同时向某个对象传递消息。

例如，一个汽车对象具有“行驶”这项操作，那么要让汽车以时速 50 公里行驶，需传递给汽车对象“行驶”及“时速 50 公里”的消息。

通常，一个消息由下述三部分组成：

① 接收消息的对象名称；

② 消息标识符（也称为消息名）；

③ 零个或多个参数。

例如，MyCircle 是一个半径 5cm、圆心位于（100，200）的 Circle 类的对象，也就是 Circle 类的一个实例，当要求它以绿颜色在屏幕上显示时，在 C++语言中应该向它发送以下消息：

MyCircle.Show（GREEN）；

其中，MyCircle 是接收消息的对象名字，Show 是消息名，Green 是消息的参数。

1.3.4　继承（Inheritance）

继承是父类和子类之间共享数据方法的机制。继承是使用已有的类定义作为基础建立新类的定义技术。已有的类可当作基类来引用，则新类可当作派生类来引用。

广义地说，继承是指能够直接获得已有的性质和特征，而不必重复定义它们。

面向对象软件技术的许多强有力的功能和突出的优点，都来源于把类组成一个层次结构的系统：一个类的上层可以有父类，下层可以有子类。这种层次结构系统中的一个重要性质是继承性，一个类直接继承其父类的描述（数据和操作）或特性，子类自动共享基类中定义的数据和方法。

为了更深入、更具体地理解继承性的含义，如图 1-6 所示给出了实现继承机制的原理图。

图中以 A、B 两个类为例，其中类 B 是从类 A 派生出来的子类，它除了具有自己定义的特性（数据和操作）之外，还从父类 A 继承特性。创建类 A 的实例 a1 时，a1 以类 A 为样板建立实例变量。

当创建类 B 的实例 b1 时，b1 既要以类 B 为样板建立实例变量，又要以类 A 为样板建立实例变量，b1 所能执行的操作即类 B 中定义的方法，又有类 A 中定义的方法，这就是继承。

继承具有传递性，如果类 C 继承类 B，类 B 继承类 A，则类 C 继承类 A。因此，一个类实际上继承了它上层的全部基类的特性，也就是说，属于某类的对象除了具有该类所定义的特性外，还具有该类上全部基类定义的特性。

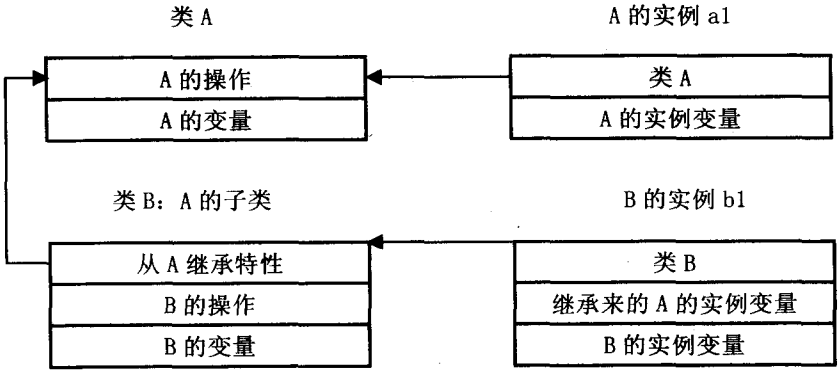

图 1-6　实现继承机制的原理图

继承分为单继承与多重继承。单继承是指一个类只允许有一个父类，即类等级为树形结构。多重继承是指一个类允许有多个父类。多重继承的类可以组合多个父类的性质构成所需要的性质。因此，功能更强，使用更方便；但是，使用多重继承时要注意避免二义性。继承性的优点是，相似的对象可以共享程序代码和数据结构，从而大大减少了程序中的冗余信息，提高软件的可重用性，便于软件修改维护。另外，继承性使用户在开发新的应用系统时不必完全从零开始，可以继承原有的相似系统的功能或者从类库中选取需要的类，再派生出新的类以实现所需要的功能。

1.3.5 多态性（Polymorphism）

对象根据所接受的消息而做出动作，同样的消息被不同的对象接受时可导致完全不同的动作，该现象称为多态性。在面向对象的软件技术中，多态性是指子类对象可以像父类对象那样使用，同样的消息既可以发送给父类对象也可以发送给子类对象。

例如，在两个类 Male（男性）和 Female（女性）都有一项属性为 Friend。一个人的朋友必须属于类 Male 或 Female，这是一个多态性的情况。因为，Friend 指向两个类之一的实例。如果 Tom 的朋友或者是 Mary 或者是 john，类 Male 就不知道 Friend 应该与哪个类关联。这里参照量 Friend 必须是多态的，多态意味着可以关联不同的实例，而实例可以属于不同的类。

多态性机制不仅增加了面向对象软件系统的灵活性，进一步减少了信息冗余，而且显著地提高了软件的可重用性和可扩充性。当扩充系统功能增加新的实体类型时，只需派生出与新实体类相应的新的子类，完全无需修改原有的程序代码，甚至不需要重新编译原有的程序。利用多态性，用户能够发送一般形式的消息，而将所有的实现细节都留给接受消息的对象。

【典型例题】

（1）下列选项中不属于结构化程序设计方法的是_____。

A. 逐步求精　　B. 自顶向下　　C. 模块化　　D. 可复用

参考答案：D。

解析：20 世纪 70 年代以来，提出了许多软件设计方法，主要有（A）逐步求精：对复杂的问题，应设计一些子目标做过渡，逐步细化。（B）自顶向下：程序设计时应先考虑总体，后考虑细节；先考虑全局目标，后考虑局部目标。不要一开始就过多追求众多的细节，先从最上层总目标开始设计，逐步使问题具体化。（C）模块化：一个复杂问题，肯定是由若干简单的问题构成。模块化是把程序要解决的总目标分解为分目标，再进一步分解为具体的小目标，把每个小目标称为一个模块。（D）可复用：是面向对象程序设计的一个优点。

（2）结构化程序设计主要强调的是_____。

A. 程序的规模　　B. 程序的易读性

C. 程序的执行效率　　D. 程序的可移植性

参考答案：B。

解析：结构化程序设计主要强调的是结构化程序清晰易读，可理解性好，程序员能够进行逐步求精、程序证明和测试，以保证程序的正确性。

（3）对建立良好的程序设计风格，下面描述正确的是_____。

A. 程序应简单、清晰、可读性好

B. 符号名的命名要符合语法

C. 充分考虑程序的执行效率

D. 程序的注释可有可无

参考答案：A。

解析：要形成良好的程序设计风格，主要应注重和考虑下述一些因素：符号名的命名应具有一定的实际含义，以便于对程序功能的理解；正确的注释能够帮助读者理解程序；程序编写

应优先考虑清晰性，除非对效率有特殊要求，程序编写要做到清晰第一，效率第二。

（4）下面对对象概念描述错误的是_____。

A. 任何对象都必须有继承性　　B. 对象是属性和方法的封装体

C. 对象间的通信靠消息传递　　D. 操作是对象的动态性属性

参考答案：A。

解析：对象是由数据和允许的操作组成的封装体，与客观实体有直接的对应关系。对象之间通过传递消息互相联系，以模拟现实世界中不同事物彼此之间的联系。

（5）在结构化方法中，软件功能分解属于下列软件开发中的_____阶段。

A. 详细设计　　B. 需求分析　　C. 总体设计　　D. 编程调试

参考答案：C。

解析：总体设计过程通常由两个主要阶段组成：系统设计，确定系统的具体实现方案；结构设计，确定软件结构。为确定软件结构，首先需要从实现角度把复杂的功能进一步分解。分析员结合算法描述，仔细分析数据流图中的每个处理，如果一个处理的功能过分复杂，必须把它的功能适当地分解成一系列比较简单的功能。

（6）在设计程序时，应采纳的原则之一是_____。

A. 程序结构应有助于读者理解　　B. 不限制GO TO语句的使用

C. 减少或取消注解行　　D. 程序越短越好

参考答案：A。

解析：滥用 GO TO 语句将使程序流程无规律，可读性差；添加的注解行有利于对程序的理解，不应减少或取消；程序的长短要依照实际需要而定，并不是越短越好。

（7）在面向对象方法中，_____描述的是具有相似属性与操作的一组对象。

参考答案：类。

解析：将属性、操作相似的对象归为类，也就是说，类是具有共同属性、共同方法的对象的集合。所以，类是对象的抽象，它描述了属于该对象类型的所有对象的性质，而一个对象则是其对应类的一个实例。

（8）结构化程序设计方法的主要原则可以概括为自顶向下、逐步求精、_____和限制使用GO TO语句。

参考答案：模块化。

解析：结构化程序设计方法的主要原则可以概括为自顶向下、逐步求精、模块化和限制使用GO TO语句。

自顶向下：程序设计时，应先考虑总体，后考虑细节；先考虑全局目标，后考虑局部目标。不要一开始就过多追求众多的细节，先从最上层总目标开始设计，逐步使问题具体化。

逐步求精：对复杂问题，应设计一些子目标做过渡，逐步细化。

模块化：一个复杂问题，肯定是由若干简单的问题构成。模块化是把程序要解决的总目标分解为分目标，再进一步分解为具体的小目标，把每个小目标称为一个模块。

限制使用 GO TO 语句：GO TO 语句能破坏程序的结构化设计，使代码难于测试，且包含大量 GO TO 的代码模块不易理解和阅读。它一直遭结构化程序设计思想所抛弃，强烈建议程序员不易使用它。

（9）与结构化需求分析方法相对应的是_____方法。