

程序设计基础实践 课程设计指导书

合肥工业大学

计算机与信息学院

2023 年 10 月

一、概述

课程设计是对学生的一种全面综合训练，是与课堂听讲、自学和练习相辅相成的、必不可少的一个教学环节。通常，课程设计中的问题比平时的习题复杂的多，也更接近实际。课程设计着眼于原理与应用的结合点，使学生学会如何把书上学到的知识用于解决实际问题，培养软件工作所需要的动手能力；另一方面，能使书上的知识变“活”，起到深化理解和灵活掌握教学内容的目的。平时的习题较偏重于如何编写功能单一的“小”算法，局限于一个或两个知识点，而课程设计题是软件设计的综合训练，包括问题分析，总体结构设计，用户界面设计、程序设计基本技能和技巧，多人合作，以至一整套软件工作规范的训练和科学作风的培养。此外，还有很重要的一点是：计算机是比任何教师更严厉的检查者。

为达到上述目的，使学生更好地掌握程序设计的基本方法和 C++语言的应用，本课程安排了课程设计环节，提供了各类题目供学生选择。每个设计题采取了统一的格式，由问题描述、基本要求、测试数据、实现提示和选做内容等五个部分组成。问题描述旨在为学生建立问题提出的背景，指明问题“是什么”。基本要求则对问题进一步求精，划出问题的边界，指出具体的参量或前提条件，并规定该题的最低限度要求。测试数据部分旨在为检查学生上机作业提供方便。在实现提示部分，对实现中的难点及其解法思路等问题作了简要提示，提示的实现方法未必是最好的，学生不应拘泥与此，而应努力设计和开发更好的方法和结构。选做部分向那些尚有余力的读者提出了更高的要求，同时也能开拓其它读者的思路，在完成基本要求时就力求避免就事论事的不良思想方法，尽可能寻求具有普遍意义的解法，使得程序结构合理，容易修改、扩充和重用。

二、总体要求

1. 系统分析与系统设计

“分析就是在采取行动之前，对问题的研究”（Demarco, 1978）。系统分析在软件开发过程中是非常重要的，其任务就是通过对问题本身的研究，产生一个系统需要做什么的规范的、一致的和可行的需求说明。在此基础上，确定系统中所需考虑类（对象）、类之间的关系、本系统中各个类所涉及的属性及针对这些属性的操作。类及类之间的关系可用类图来表示，对象之间的消息传递可用箭头表示，另外一些重要的操作应给出规格说明。

2. 详细设计与编码

对类中的属性和操作从实现的角度（如可扩充、在派生类中能否直接使用或只需少量修改、访问的效率和方便性等）进一步考察；对类中的操作（即方法）进一步求精：用 if、while、for 和赋值语句加上自然语言写出算法框架；同时考虑能否使用已有类库（包括直接使用或通过派生）以减少编程的工作量和提高程序的可靠性。

编码，即程序设计，是对详细设计的结果的进一步求精，用面向对象语言（如 C++）表达出来。在充分理解和把握语言运行机制的基础上，编写出正确的、清晰的、易读易改和高效率的程

序。另外，在标识符的命名、代码的长度（一个方法长度一般不超过 40 行，否则应划分为两个或多个方法）、程序书写的风格（如缩进格式、空格（空行）的应用、注释等）方面也应注意，遵循统一的规范。

3. 上机调试和测试

上机时要带一本面向对象语言的教材，若有开发环境的用户指南（手册）及类库（库函数）手册则更好。应仔细阅读程序编译和连接时的错误信息（通常是英文的），弄清其确切含义，提高调试效率。要学习并掌握开发环境所提供的调试工具。

经过调试，能够运行的程序并非就是一个正确的程序。实际上，在上机之前，就应根据系统的需求设计相应的测试数据集，特别是一些异常情况的处理（如用户输入数据未按指定格式、数据极大或极小时程序如何处理等一些极端的情况）。

4. 课程设计报告

课程设计报告的内容及要求：

一. 需求和规格说明

描述问题，简述题目要解决的问题是什么？规定软件做什么。原题条件不足时补全

二. 设计

- 1) 设计思想：程序结构（如类图），重要的数据结构。主要算法思想（文字描述，不要画框图）
- 2) 设计表示：类名及其作用，类中数据成员名称及作用，类中成员函数原型及其功能，可以用表格形式表达。
- 3) 实现注释：各项要求的实现程度、在完成基本要求的基础上还实现了什么功能？
- 4) 详细设计表示：主要算法的框架及实现此算法的成员函数接口。

三. 用户手册

即使用说明（包括数据输入时的格式要求）。

四. 调试及测试

调试过程中遇到的主要问题是如何解决的；对设计和编码的回顾讨论和分析；程序运行的时空效率分析；测试数据集；运行实例；改进设想；经验和体会等。

附录

- 1) 源程序清单：打印文本和磁盘文件，磁盘文件是必须的。源程序要加注释，除原有注释外再用钢笔加一些必要的注释和断言。
- 2) 测试数据：即列出测试数据集
- 3) 运行结果：上面测试数据输入后程序运行的结果

注意事项：

1. 以上要求为一般的要求，针对具体问题和具体的开发过程，某些方面可以做适当的增减。
2. 各种文档资料要在程序开发过程中逐渐形成，而不是最后补写（但不排斥最后誉清）。
3. 各种文档要以统一格式的稿纸用钢笔书写，也可录入计算机用 Word 及其它文字编辑软件排版后打印输出。

三、课程设计示例

封面：

课程设计报告	
设计题目：	小型公司人员信息管理
学生姓名：	
专 业：	
班 级：	
学 号：	
指导教师：	
完成日期：	
合肥工业大学计算机与信息学院	

（一）需求和规格说明

某小型公司，主要有四类人员：经理、技术人员、销售经理和推销员。要求存储这些人员的姓名、编号、级别、当月薪水，计算月薪总额并显示全部信息。

人员编号基数为 1000，每输入一个人员的信息，编号顺序加 1。

程序要对所有人员有提升级别的功能。为简单起见，所有人员的初始级别均为 1 级，然后进行升级，经理升为 4 级，技术人员和销售经理升为 3 级，推销员仍为 1 级。

月薪计算办法是：经理拿固定月薪 8000 元；技术人员按每小时 100 元领取月薪；推销员的月薪按该推销员当月销售额的 4%提成；销售经理既拿固定月薪也领取销售提成，固定月薪为 5000 元，销售提成为所管辖部门当月销售总额的 5%。

（二）设计

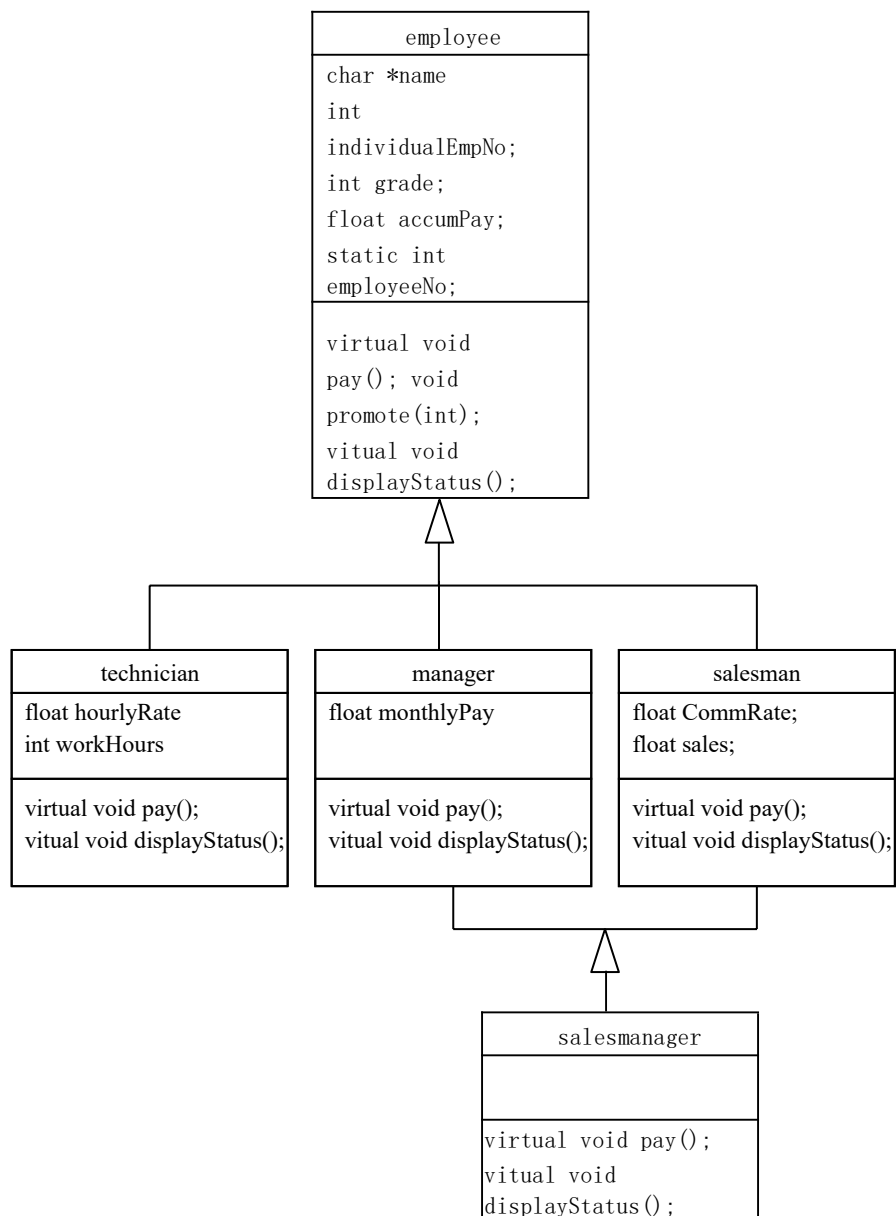
根据上述需求，设计一个基类 employee，然后派生出 technician（技术人员）类、manager（经理）类和 salesman（推销员）类。由于销售经理（salesmanager）既是经理又是销售人员，兼具两类人员的特点，因此同时继承 manager 和 salesman 两个类。

在基类中，除了定义构造函数和析构函数以外，还应统一定义对各类人员信息都应有的操作，这样可以规范各派生类的基本行为。但是各类人员的月薪计算方法不同，不能在

基类 employee 中统一定义计算方法。各类人员信息的显示内容也不同，同样不能在基类中统一定义显示方法。因此，在employee 类中用纯虚函数的方式定义了计算月薪函数pay()和显示信息函数 displayStatus()，然后在派生类中再根据各自的同名函数实现具体的功能。

由于 salesmanager 的两个基类又有公共基类 employee，为避免二义性，这里将 employee 类设计为虚基类。

系统类图



属性和方法定义

类名	成员类别	类型	成员名	描述
employee	属性	char *	name	雇员姓名
		int	individualEmpNo	个人编号
		int	grade	级别
		float	accumPay	月薪总额
		int	employeeNo	本公司雇员编号目前最大值
	方法	void	pay()	计算月薪函数（为纯虚函数）
		void	promote(int)	升级函数
		void	DisplayStatus()	显示人员信息（为纯虚函数）

类名	成员类别	类型	成员名	描述
technician	属性	float	hourlyRate	每小时酬金
		int	workHours	当月工作时数
	方法	void	pay()	accumPay=hourlyRate*workHours
		void	DisplayStatus()	显示技术人员信息
manager	属性	float	monthlyPay	固定月薪数
	方法	void	pay()	AccumPay=monthlyPay
		void	DisplayStatus()	显示经理信息
salesman	属性	float	CommRate	按销售额提取酬金百分比
		float	sales	当月销售额
	方法	void	pay()	accumPay=sales*CommRate
		void	DisplayStatus()	显示推销员信息
salesmanager	属性			
	方法	void	pay()	accumPay=monthlyPay+CommRate*sales
		void	DisplayStatus()	显示销售经理信息

（三） 用户手册

程序运行时，首先提示输入雇员姓名。

对于经理直接输出其工资及其它信息；

对于技术人员，程序提示输入其本月工作时数，然后输出其工资及其它信息；

对于推销员，程序提示输入其本月销售额，然后输出其工资及其它信息；

对于销售经理，程序提示输入其管辖部门本月销售总额，然后输出其工资及其它信息。

（四） 调试及测试

由于公司每增加一个雇员，无论他（她）是哪一类人员，其编号均是顺序加 1，也就是 employee 类的所有派生类对象创建时，都要访问同一个 employeeNo，因此将 employeeNo 定义为静态数据成员。

运行实例：

```
please input employee's name: zhang
please input employee's name: wang
please input employee's name: Li
please input employee's name: zhao
input zhang the workHours of this month: 56
Technician: zhang No: 1001 month salary: 5600
Technician: zhang No: 1001 grade: 3 this month salary: 5600
Manager: wang No: 1002 month salary: 8000
Manager: wang No: 1002 grade: 4 this month salary: 8000
input Li the sales of this month: 47900
Salesman: Li No: 1003 month salary: 1916
Salesman: Li No: 1003 grade: 1 this month salary: 1916
input zhao the total sales of the department of this month: 123654
salesman: zhao No: 1004 month salary: 5618.27
salesmanager: zhao No: 1004 grade: 3 this month salary: 5618.27
```

进一步改进

（1）目前程序中，经理月薪，技术人员的小时酬金和销售人员的销售额提成比例均是固定的，这不适应不同公司的需要，可考虑用带参数的构造函数来解决。

（2）销售经理月薪计算中，要输入其管辖部门当月销售总额。实际上，这可以通过将本部门所有推销员销售额相加而得到。可以考虑在推销员类中增加所属部门等属性来完成这方面的功能。

附录 源程序

设计题

本手册提供的设计题大致可分为 3 类：第一类侧重于算法设计与实现；第二类侧重于文件操作；第三类则属于综合类（侧重于程序结构的设计与实现，当然也包括算法设计与文件操作）。前两类问题可用 C++面向过程的方式来实现，而后一类则需用 C++面向对象的方式来实现。当然，这种划分不是绝对的，我们期望同学尽可能用面向对象的方式来完成这些题目。

题目1： 排班系统

学校实验楼有7名保安人员：钱、赵、孙、李、周、吴、陈。由于工作需要轮休制度，一星期中每人休息一天。预先让每一个人选择自己认为合适的休息日。请编程序，打印轮休的所有可能方案。当然使每个人都满意，例如每人选择的休息日如下：

钱：星期一、星期六

赵：星期二、星期四

孙：星期三、星期日

李：星期五

周：星期一、星期四、星期六

吴：星期二、星期五

陈：星期三、星期六、星期日

运行结果：

方案1

赵	钱	孙	李	周	吴	陈
星期四	星期一	星期三	星期五	星期六	星期二	星期日

方案2

赵	钱	孙	李	周	吴	陈
星期四	星期一	星期日	星期五	星期六	星期二	星期三

方案3

赵	钱	孙	李	周	吴	陈
星期四	星期六	星期三	星期五	星期一	星期二	星期日

方案4

赵	钱	孙	李	周	吴	陈
星期四	星期六	星期日	星期五	星期一	星期二	星期三

功能要求：

1、保安人员个人信息、排班情况需要保存到文件

- 2、能从文件读取到已经完成的本周排班情况，并存储到链表进行处理。
- 3、使用链表实现排班情况的修改或删除，修改完毕，结果需要保存至文件。
- 4、文件中要保存最近四周的排班信息，前三周的只能查看，本周的排班情况可以修改。
- 5、输入保安人员的名字，查询最近四周工作哪些天，休息那几天。
- 6、输入保安人员的名字，能显示保安的个人详细信息，保安个人的详细信息应包含人员代码、姓名、年龄（整数型）、性别、家庭住址等。
- 7、能够对保安的个人信息进行修改、删除。

题目2：小学生测验

面向小学1~2年级学生，随机选择两个整数和加减法形成算式要求学生解答。

功能要求：

- （1）进入测试之前先输入用户名、密码登录，以方便保存学生信息。
- （2）出题模块：电脑随机出10道试题，每题10分，测试过程中，试题信息保存在链表中，程序结束时显示学生得分；确保算式没有超出12年级的水平，只允许进行50以内的加减法，不允许两数之和或之差超出050的范围，负数更是不允许的；
- （3）信息保存：要求用户信息、试题信息保存进文件，需要时可以调出试卷进行重新测试、修改试题内容等。
- （4）试卷基本操作功能：试题修改、试题插入、试卷删除等。
- （5）答题模块：每道题学生有三次机会输入答案，当学生输入错误答案时，提醒学生重新输入，如果三次机会结束则输出正确答案；
- （6）计分功能：对于每道题，学生第一次输入正确答案得10分，第二次输入正确答案得7分，第三次输入正确答案得5分，否则不得分；
- （7）成绩保存：将测试记录保存至文件，可以查看学生的解答情况，包括得分等。
- （8）结果显示：总成绩90以上显示“SMART”，80-90显示“GOOD”，70-80显示“OK”，60-70显示“PASS”，60以下“TRY AGAIN”。
- （9）重考功能：不及格的学生，由学生自己选择是否重新来一次，可以重新做以前的测试卷，也可以选择重新做一套试卷。重做机会只有一次。
- （10）排行榜功能：排行榜显示用户名、分数、做题时间等。

题目3：年历及时钟图形化显示

功能要求：

- （1）有要登录界面，用菜单方式工作。
- （2）显示当前的日期和时间，以图形化模式显示时钟钟面，且有时针、分针、秒针的运动。
- （3）输入一个年份，输出是在屏幕上显示该年的日历，显示效果如下图所示。假定输入的年份在1940-2040年之间。
- （4）输入年月，输出该月的日历。

- (5) 输入年月日，输出距今天还有多少天，星期几，是否是公历节日。
- (6) 输入当天的待办事宜，可以输入单条也可以输入多条，并将相关信息保存至文件。
- (7) 输入年月日，可以查询当天的待办事宜，并且可以进行修改、删除。

日历非图形界面显示，格式控制对齐，从星期天、星期一到星期六排列，大致如下图：



题目4： 足球联赛积分

问题描述：

足球联赛采用主客场双循环赛制，胜一场得3分，平局各得1分，负一场得0分，联赛排名以积分多者在前，当两队（或多队）积分相同时，则净胜球（即进球数与失球数之差）多者在前，若净胜球相同，则进球数多者在前，若仍相同，则抽签或踢附加赛决定名次（这在联赛结束后进行，联赛未结束则两队名次并列，本程序不做这方面要求）。试编一程序统计最近一轮比赛后，各队积分及排名。

基本要求：

设积分表结构如下：队名，已比赛的场数，赢的场数，平的场数，负的场数，**进球数**，**失球数**，积分。积分表放在正文文件中。通过用户界面将最近一轮的结果输入，其形式为：主队名（可用代码），客队名（可用代码），主队得分（即进球数），客队得分（即进球数）。程序应根据此轮结果修改各队的积分和名次，所得的最新记分表仍在原积分文件中并同时屏幕上显示。

测试数据：

可选择我国当年的中超联赛的数据输入，并检查与网上公布的数据是否一致。

实现提示：

定义一个球队类，每个球队均是此类的对象。由于联赛中参赛的队伍数是固定的，因此可用对象数组来实现（当然也可以用链表结构）。每输入两个队的比赛成绩，则相应的队的有关数据（比赛场数，赢的场数，平的场数，负的场数，进球数，失球数，积分等）即可进行修改，比赛成绩录入完成，调用联赛排序方法（对象数组作为参数）排出名次并输出。

题目5： 实验设备管理系统设计

实验设备信息包括：设备编号，设备种类(如：微机、打印机、扫描仪等等)，设备名称，设备价格，设备购入日期，经手人，是否报废，报废日期等。

主要功能：

- 1、能够完成对设备的录入和修改，并将各类信息分别保存到文件。
- 2、对设备进行分类统计，统计时要给总价值。
- 3、设备的破损耗费和遗失处理，并将相关信息保存至文件。
- 4、设备的查询功能。
- 5、可以按数字、设备等的购买日期进行排序。
- 5、修改和删除功能使用链表实现。

要求：采用二进制文件方式存储数据，系统以菜单方式工作

题目6： 2048游戏

基本要求

- (1) 实现游戏规则
- (2) 使用图形函数生成界面等。
- (3) 用文件存储用户的进度。
- (4) 用户开始新游戏时，先检测是否有历史记录，有的话可以继续未完成的游戏，也可以重新开始。
- (5) 实现用户排名功能，要求能够将排名信息进行保存，存至文件永久保存。
- (6) 当新用户的成绩需要插入排名列表时，要能够修改原列表信息；如果是同一用户需要更新成绩，则覆盖原成绩。
- (7) 插入、修改、删除排名信息等要求使用链表实现。

游戏规则：

2048游戏共有16个格子，初始时初始数字由2或者4构成。

- 1、手指向一个方向滑动，所有格子会向那个方向运动。
- 2、相同数字的两个格子，相撞时数字会相加。
- 3、每次滑动时，空白处会随机刷新出一个数字的格子。
- 4、当界面不可运动时（当界面全部被数字填满时）

游戏结束：当界面中最大数字是2048时，游戏胜利。

游戏效果如下图



题目7： 单项选择题标准化考试系统设计

一、功能要求：

- 1、采用文本菜单界面（如果能采用图形菜单界面更好）
- 2、用文件保存试题库。（每个试题包括题干、4个备选答案、标准答案）
- 3、试题录入：可随时增加试题到试题库中，但如果出现与题库中已存在题的题干相同的，则给出提示信息，不能增加）
- 4、试题修改、删除、插入等功能要求用链表实现。
- 5、试题抽取：每次从试题库中可以随机抽出N道题（N由键盘输入）
- 6、答题：进入测试之前先输入用户名、密码登录，以方便保存用户信息。用户可实现输入自己的答案
- 7、自动判卷：系统可根据用户答案与标准答案的对比实现判卷并给出成绩。
- 8、排行榜功能，排行榜显示用户名、分数、做题时间等。
- 9、查看历史答卷情况。

题目8：贪吃蛇游戏

1. 目的与要求

- （1）界面友好（图形界面，良好的人机交互）
- （2）实现用户排名功能，要求能够将排名、成绩等信息进行保存，存至文件永久保存。
- （3）当新用户的成绩需要插入排名列表时，要能够修改原列表信息；如果是同一用户需要更新成绩，则覆盖原成绩。
- （4）插入、修改、删除排名信息等要求使用链表实现。

2. 实现提示

贪吃蛇游戏是一个经典小游戏，一条蛇在封闭围墙里，围墙里随机出现一个食物，通过按键盘四个光标键控制蛇向上下左右四个方向移动，蛇头撞倒食物，则食物被吃掉，蛇身体长一节，同时记10分，接着又出现食物，等待蛇来吃，如果蛇在移动中撞到墙或身体交叉蛇头撞倒自己身体游戏结束。

这个程序的关键是表示蛇的图形以及蛇的移动。用一个小矩形表示蛇的一节身体，身体每长一节，增加一个矩形块，蛇头用两节表示。移动时必须从蛇头开始，所以蛇不能向相反方向移动，也就是蛇尾不能改作蛇头。如果不按任何键，蛇自行在当前方向上前移，当游戏者按了有效的方向键后，蛇头朝着指定的方向移动，一步移动一节身体，所以当按了有效的方向键后，先确定蛇头的位置，然后蛇身体随着蛇头移动，图形的实现是从蛇头的新位置开始画出蛇，这时由于没有庆平的原因，原来蛇的位置和新蛇的位置差一个单位，所以看起来蛇多一节身体，所以将蛇的最后一节用背景色覆盖。食物的出现和消失也是画矩形块和覆盖矩形块

题目9： 扫雷游戏

请参考Windows中的扫雷游戏，编程实现一个简易版的扫雷游戏。

基本要求

- (1) 实现扫雷游戏的各个规则
- (2) 使用图形函数生成界面等。
- (3) 用文件存储用户的进度。
- (4) 用户开始新游戏时，先检测是否有历史记录，有的话可以继续未完成的游戏，也可以重新开始。
- (5) 实现用户排名功能，要求能够将排名、成绩信息进行保存，存至文件永久保存。
- (6) 当新用户的成绩需要插入排名列表时，要能够修改原列表信息；如果是同一用户需要更新成绩，则覆盖原成绩。
- (7) 插入、修改、删除排名、成绩等信息等要求使用链表实现。

题目10： 俄罗斯方块游戏

1. 目的与要求

- (1) 游戏规则：7种状态的方块（长条型、Z字型、反Z字型、田字型、7字型、反7字型、T字型）随机产生，自由下落，落下时可由玩家用上、下、左、右控制键控制翻转和移动，以便以万家所需要的形态和位置落下。如果落下时，方块的方格能填满某一行，则这一行可消去。消去一行后，游戏可给玩家加分，若由存在空格的方块填满整个窗口，则游戏失败。
- (2) 游戏界面：游戏的背景色是黑色，方块用蓝色，在一定区域内运动和变形，落下后的障碍物用黄色显示。
- (3) 游戏形状：由数组作为存储方块状态的数据结构，各个方块要能实现下落、移动、旋转，旋转可设为顺时针或逆时针变形，一般为逆时针，实现下落底部方块的处理。
- (4) 键盘处理：方块下落时，可通过键盘方向键（上、下、左、右键）对该方块进行向上（变形），向下（加速）、向左、向右移动。
- (5) 鼠标事件：通过点击菜单栏中相应的菜单项，可以实现游戏的开始、结束，方块形状的变换，分数、等级的显示，以及游戏帮助等功能。
- (6) 显示需求：当不同的方块填满一行时可以消行，剩余方块向下移动并统计分数。当达到一定分数的时候，会增加相应的等级。
- (7) 实现用户排名功能，要求能够将排名信息进行保存，存至文件永久保存。
- (8) 当新用户的成绩需要插入排名列表时，要能够修改原列表信息；如果是同一用户需要更新成绩，则覆盖原成绩。
- (9) 插入、修改、删除排名信息等要求使用链表实现。

2. 实现提示

- (1) 方块如何实现旋转：通用的方法是旋转90度，还可以把每个方块各种变形的样子都用结构体定义好，形成一个封闭的链表，变形一次指针指向下一种形状。
- (2) 方块是否还能下落：用一个带有返回值的函数，若碰撞则说明不能下落，返回1；反则说明没

有碰撞，返回0。即将整个4×4方块数组下落看成一个单位长度，与游戏空间数组有重叠的1，则在当前位置4×4数组是1的地方赋值给游戏空间对应的数组元素，表示停止下落，并画有1的地方。对于左移、右移一个单位长度有重叠的1，则不允许左移、右移，继续自然下落。

(3) 方块如何实现消行：先认为每一行都是满1的，从游戏空间的数组由上到下扫描，一旦测试到某一行中某个列元素为0，则认为这一行没满，跳出这行的扫描循环，进入下一行的扫描。若扫描完某一行的元素都没有发现0，则以这行以上的每一行完完整整地将上一行的元素赋值给下一行，这个过程以由下到上进行，然后将整个游戏空间画黑，再在有1的地方画小正方形。

题目11：五子棋游戏

1. 基本要求

- (1) 实现五子棋游戏的各个规则
- (2) 使用图形函数生成棋盘等。
- (3) 用文件存储用户的进度
- (4) 用户开始新游戏时，先检测是否有历史记录，有的话可以继续未完成的棋局，也可以重新开始。
- (5) 实现用户排名功能，要求能够将排名信息进行保存，存至文件永久保存。
- (6) 当新用户的成绩需要插入排名列表时，要能够修改原列表信息；如果是同一用户需要更新成绩，则覆盖原成绩。
- (7) 插入、修改、删除排名信息等要求使用链表实现。

2. 目的与要求

(1) 游戏规则：传统五子棋的棋具与围棋相同，棋子分为黑白两色，盘为19×19，棋子放置于棋盘线交叉点上。两人对局，各执一色，轮流下一子。如果某一方下棋的棋子最先5子连续连成一线（行、列、对角线），则该方为赢家，游戏一局结束；如果棋盘下满未定胜负则为平局，游戏一局结束。

(2) 功能模块：将程序分为图形显示、玩家控制、胜负判断和玩家计分四个模块。

① 图形显示模块：程序开始运行时，给出欢迎及帮助界面；游戏开始后要求生成19×19的棋盘图像，并在棋盘上方提示当前落子方棋子颜色，游戏进行过程中，要求实时显示棋盘上已落下的棋子；分出胜负后，要求给出游戏结束画面。

② 玩家控制模块：程序开始时，需玩家确定而后开始游戏；游戏过程中，两个玩家通过不同的按键移动光标，选择落子；游戏结束时，有玩家选择是否开始新棋局。

③ 胜负判断模块：实时监测棋盘上棋子，一旦某一色棋子出现五子连线，终止游戏程序，并着色连成一线的五子，弹出该色玩家胜出界面。

④ 玩家计分模块：一方玩家再胜利后通过对文件的操作进行计分，并输出于计分板上。

(3) 其他要求：

① 进入演示程序后，即显示欢迎界面，按任何键进入帮助界面，再按任何键可以进入主界面开始游戏。

② 棋子的移动与落子有键盘上按键控制，本程序选取1P按键为W、S、A、D和空格键，2P按键为↑、↓、←、→和回车键，分别代表上移、下移、左移、右移光标和落子。在光标移动的过程中，光标按照玩家按键移动；在玩家按下落子按键后，程序自动调用棋子显示子程序和判断胜负子程序。1P、2P的落子后，程序会为落子处的数组元素赋一个特定值，用于判定胜负。

③ 游戏中esc键可以直接退出游戏，backspace键可以操作进行悔棋。

④ 游戏过程中，如果玩家1或者玩家2有一方获得胜利后，程序自动将提示哪一方获得了胜利，并将着色构成连珠的5个子。

⑤ 游戏结束且玩家选择不再继续后，跳出结束界面。退出程序。

3. 实现提示

(1) 程序不涉及人机交互，算法较为简单。首先，以落子为出发点，分别沿着水平、竖直和两条对角线方向（分别为45°和135°）进行搜索，看在这四个方向是否最后落子的一方构成连续五个棋子。为了提高搜索速度，尽量减少搜索范围。以落子为中心两侧各判断四子共九子，判断这九子中是否有最后落子的一方的连续五个棋子。只要最后落子的一方在任一方向上有连续五个棋子，就表示该盘棋局已经分出胜负。

(2) 监控键盘输入的函数及各键的ASCII码值；图形方式下的输入、输出及其相关的函数；判定五子成一线的方法，即矩阵中行向、列向、两条对角线方向上是否有连续五子；调用系统提供的声音函数等。

题目12：组数游戏

问题描述：

输入正整数的个数为 n ，输出 n 个数连接成的最大的多位数。找出 n 个数中最大数字的位数，然后将所有的数字通过后面补零的方式扩展成最大位数。将变换后的 n 位数从大到小排序，然后将添加上的零去掉，按顺序输出的序列即为所求的最大数字。

说明：数据结构采用结构体，由于正整数的位数不确定，可能非常大，可以用字符类型。

题目13：医院预约系统

医生信息：医生编号、姓名、科室、诊室。

预约信息：医生编号、年月日、预约时间段、预约人姓名、就诊卡号

医院预约系统的功能要求：

- (1) 医生信息和预约信息，以文件形式进行保存；
- (2) 医生信息录入；
- (3) 修改医生信息和预约信息；
- (4) 删除医生信息和预约信息；
- (5) 医生排班，生成预约信息（医生编号、年月日、预约时间段）
- (6) 就诊：选择医生的上班時間，输入预约人的信息，同一时间只能预约一个病人
- (7) 就诊信息记录：按时就诊，缺诊，超时就诊

题目14：选择题自动判卷系统

功能要求：

- (1) 用文件形式保存试题库。（每个试题包括：题干、4个备选选项、标准答案）；
- (2) 试题录入：可随时增加试题到试题库中；
- (3) 试题抽取：每次从试题库中随机抽取 N 道题（ N 由键盘输入）；
- (4) 答题：用户输入自己的答案。答完题之后，保存，退出；
- (5) 自动判卷：系统可根据用户答案与标准答案的对比实现判卷，并给出成绩。

题目15：保龄球游戏

打保龄球是用一个滚球去撞击10个站立的瓶，将瓶击倒。一局分10轮，每轮可以滚球1次或多次，以击倒的瓶数计分，一局得分为10轮得分之和，而每轮的得分不仅与本轮的滚球情况有关，还可能与后一轮或两轮的滚球情况有关，即：某轮某次滚球击倒半瓶数不仅要计入本轮得分，还可能计入前一轮或两轮得分。计分规则如下：

(1) 若某一轮的第一次滚球就击倒全部10个瓶，则本轮不再滚球（若是第10轮还需加2次滚球），该轮得分为本次击倒瓶数10与后面2次滚球所击倒瓶数之和；

(2) 若某一轮的第一次滚球未击倒全部10个球，则对剩下未击倒的瓶再滚球一次，如果这2次滚球击倒全部10个瓶，则本轮不再滚球（若是第10轮还需要加1次滚球），该轮得分为这2次击倒瓶数10与后面1次滚球所击倒瓶数之和；

(3) 若某一轮2次滚球未击倒全部10个瓶，则本轮不再滚球，该轮得分为这2次滚球所击倒瓶数之和

说明：1) 模拟10个人各打一局保龄球比赛过程，统计每局各轮得分和累计总分

2) 对10人的得分由低到高排序并显示

3) 将排序结果存入到文件中

题目16：酒店住宿信息管理系统

住宿信息包括：住宿编号、姓名、身份证号码、联系电话、入住时间、房号、单价、入住接待人姓名、退房时间、退房接待人姓名、优惠费用、使用房内小物品费用、应付费用、备注。酒店住宿信息管理系统的功能要求：

- (1) 系统以菜单方式工作，执行一次操作后重新显示一次菜单；
- (2) 住宿信息利用文件保存；
- (3) 住宿信息的录入；
- (4) 修改住宿信息；
- (5) 删除住宿信息；
- (6) 按条件（住宿编号、姓名、身份证号码、联系电话、入住时间、房号）查询住宿信息；
- (7) 退房时根据退房时间与单价自动计算应付费用， $\text{应付费用} = \text{单价} \times \text{入住天数} - \text{优惠费用} + \text{使用房内小物品费用}$ ，入住天数根据入住时间和退房时间自动计算；
- (8) 按天、周和月统计酒店的营业额。

注意：系统自动对各种数据合理性的检查。

题目17：二手物品交易系统

物品信息包括：编号、物品名称、物品价格、数量、描述、发布者姓名、发布者联系电话、发布时间。物品交易信息包括：编号、物品价格、购买数量、总价、购买者姓名、购买者联系电话、购买时间、寄货地址。二手物品交易系统的功能要求：

- (1) 系统以菜单方式工作，执行一次操作后重新显示一次菜单；
- (2) 物品信息和物品交易信息利用文件保存；
- (3) 物品信息的录入；
- (4) 修改物品信息；
- (5) 删除物品信息；
- (6) 按条件（编号、物品名称、发布者姓名、发布者联系电话、发布时间）查询物品信息；
- (7) 购买二手物品，输入编号，自动显示物品名称和物品价格，输入购买数量，自动计算总价，生成物品交易信息后，自动修改物品信息中的数量；
- (8) 修改物品交易信息，若修改购买数量，自动修改物品信息中的数量；

(9) 删除物品交易信息，自动修改物品信息中的数量。

注意：物品信息和物品交易信息的关联，购买的物品数量不能超过物品的数量。

注意：系统自动对各种数据合理性的检查。

题目18：双端队列(deque)容器的实现

双端队列(deque)容器内部数据表示为分段数组，并提供相应的索引数组来维护多个分段数组。

双端队列(deque)容器的基本功能要求：

(1) 容器的通用功能

- 用默认构造函数构造空容器
- 支持关系运算符：`==`、`!=`、`<`、`<=`、`>`、`>=`
- `begin()`、`end()`：获得容器首、尾迭代器
- `clear()`：将容器清空
- `empty()`：判断容器是否为空
- `size()`：得到容器元素个数
- `s1.swap(s2)`：将s1和s2两容器内容交换

(2) 相关数据类型 (S表示deque容器类型)

- `S::iterator`：指向容器元素的迭代器类型
- `S::const_iterator`：常迭代器类型

(3) 可逆容器功能

- `rbegin()`：指向容器尾的逆向迭代器
- `rend()`：指向容器首的逆向迭代器

(4) 随机访问容器功能

- `s[n]`：获得容器s的第n个元素

(5) 顺序容器的接口功能

- ✎ 赋值 `assign`
- ✎ 插入函数 `insert`, `push_front`, `push_back`
- ✎ 删除函数 `erase`, `clear`, `pop_front`, `pop_back`
- ✎ 其他顺序容器访问函数 `front`, `back`
- ✎ 改变大小 `resize`

19：简单计算机模拟程序

功能要求：设计模拟计算器软件，能实现实数的加、减、乘、除四则运算，要求有界面，可参考如下：



题目20：篮球联赛积分

问题描述：

篮球联赛采用主客场双循环赛制，胜一场得2分，负一场得1分，联赛排名以积分多者在前，当两队（或多队）积分相同时，则胜场数多者在前，若胜场数相同，则看净胜分。净胜分就是球队在比赛中得分的总数减去失分的总数。如果净胜分也相同，则抽签或打附加赛决定名次（这在联赛结束后进行，联赛未结束则两队名次并列，本程序不做这方面要求）。试编一程序统计最近一轮比赛后，各队积分及排名。

基本要求：

设积分表结构如下：队名，已比赛的场数，赢的场数，负的场数，得分数，失分数，积分。积分表放在正文文件中。通过用户界面将最近一轮的结果输入，其形式为：主队名（可用代码），客队名（可用代码），主队得分，客队得分。程序应根据此轮结果修改各队的积分和名次，所得的最新记分表仍在原积分文件中并同时也在屏幕上显示。

测试数据：

可选择我国当年的CBA联赛的数据输入，并检查与网上公布的数据是否一致。

实现提示：

定义一个球队类，每个球队均是此类的对象。由于联赛中参赛的队伍数是固定的，因此可用对象数组来实现（当然也可以用链表结构）。每输入两个队的比赛成绩，则相应的队的有关数据（比赛场数，赢的场数，负的场数，总得分，积分等）即可进行修改，比赛成绩录入完成，调用联赛排序方法（对象数组作为参数）排出名次并输出。

题目21：员工管理系统设计

需求分析：

设计一个员工管理系统，用于记录和管理公司员工的基本信息，包括姓名、工号、职位、薪资等属性。系统应提供员工信息的录入、查询、编辑和删除功能，并将员工信息保存到文件中。

总体设计：

该系统设计为员工信息管理模块，包括员工信息录入、查询、编辑和删除功能，以及文件操作模块，用于读取和保存员工信息文件。

详细设计：

文件操作模块：

提供函数用于读取员工信息文件，将文件中的员工信息加载到系统中。

提供函数用于保存员工信息到文件，将系统中的员工信息写入员工信息文件。

员工信息管理模块：

提供员工信息的录入功能，允许管理员输入员工姓名、工号、职位、薪资等信息，并将录入的信息保存到员工信息文件中。

提供员工信息查询功能，允许管理员根据姓名、工号或职位等条件进行查询，并将查询结果显示给管理员。

提供员工信息编辑功能，允许管理员选择要编辑的员工，并修改其基本信息。修改后的信息将更新到员工信息文件中。

提供员工信息删除功能，允许管理员选择要删除的员工，并从系统和员工信息文件中删除该员工的信息。

请根据上述设计思路，实现员工管理系统，并根据需要添加更多功能和细节。在实际开发过程中，还需考虑数据持久化、输入验证、界面设计等方面的需求，并合理处理文件读写操作的异常情况。

题目22：旅行者问题

假设你是一个旅行者，你要从一个起点出发，经过多个城市最终到达目的地。每个城市之间有不同的距离和花费。你希望选择一条路径，使得总花费最小。设计一个程序，使用动态规划算法，计算从起点到目的地的最小花费路径。

问题输入：

城市数量 n （包括起点和目的地）

距离矩阵 D ， $D[i][j]$ 表示从城市 i 到城市 j 的距离（ $0 \leq i, j < n$ ）

花费矩阵 C ， $C[i][j]$ 表示从城市 i 到城市 j 的花费（ $0 \leq i, j < n$ ）

问题输出：

最小花费路径的总花费

实现要求：

使用动态规划算法，设计一个函数，接受城市数量 n 、距离矩阵 D 和花费矩阵 C 作为输入，并返回最小花费路径的总花费。

测试数据：

假设有4个城市，距离矩阵 D 如下：

```
D = [
    [0, 10, 15, 20],
    [10, 0, 35, 25],
    [15, 35, 0, 30],
    [20, 25, 30, 0]
]
```

花费矩阵 C 如下：

```
C = [
    [0, 2, 3, 4],
    [2, 0, 5, 3],
    [3, 5, 0, 6],
    [4, 3, 6, 0]
]
```

根据上述数据，计算从起点到目的地的最小花费路径的总花费。

题目23：电影院放映

假设你是一个电影院经理，你需要安排一天的电影放映计划。电影院有多个放映厅，每个放映厅在同一时间只能放映一部电影。每部电影有一个开始时间和结束时间，以及对应的利润。你希望选择一组

电影，使得在一天内能够获得最大的总利润，并且电影之间不能有时间上的冲突。设计一个程序，使用贪心算法，计算在一天内能够获得的最大总利润。

问题输入：

电影列表，每部电影包括开始时间、结束时间和对应的利润

问题输出：

在一天内能够获得的最大总利润

实现要求：

使用贪心算法，设计一个函数，接受电影列表作为输入，并返回在一天内能够获得的最大总利润。

测试数据：

假设有以下电影可供选择：

电影1：开始时间 9:00，结束时间 11:00，利润 100

电影2：开始时间 10:30，结束时间 12:30，利润 120

电影3：开始时间 11:30，结束时间 13:30，利润 150

电影4：开始时间 12:00，结束时间 14:00，利润 180

根据上述数据，计算在一天内能够获得的最大总利润。

题目24：快递员配送

假设你是一个快递员，需要在一天内完成一系列快递的派送任务。每个快递包裹都有一个截止时间和对应的利润。你希望选择一组快递包裹，使得在一天内能够获得最大的总利润，并且能按时完成所有派送任务。设计一个程序，使用贪心算法，计算在一天内能够获得的最大总利润。

问题输入：

快递包裹列表，每个包裹包括截止时间和对应的利润

问题输出：

在一天内能够获得的最大总利润

实现要求：

使用贪心算法，设计一个函数，接受快递包裹列表作为输入，并返回在一天内能够获得的最大总利润。

测试数据：

假设有以下快递包裹可供选择：

包裹1：截止时间 10:00，利润 100

包裹2：截止时间 11:30，利润 120

包裹3：截止时间 10:45，利润 150

包裹4：截止时间 12:00，利润 180

根据上述数据，计算在一天内能够获得的最大总利润。

题目25：排球联赛积分

问题描述：

排球联赛采用主客场双循环赛制，五局三胜制，胜三局的队胜一场，胜一场得3分，负一场得0分。

前4局比赛，以先得25分并超出对方2分的队为胜1局。决胜局以先得15分并超出对方2分的队为获胜队。全场比赛均不设最高分限。联赛排名以积分多者在前，当两队（或多队）积分相同时，则胜场数多者在前，若胜场数相同，则赢得局数多者在前，若仍相同，则抽签或打附加赛决定名次（这在联赛结束后进行，联赛未结束则两队名次并列，本程序不做这方面要求）。试编一程序统计最近一轮比赛后，各队积分及排名。

基本要求：

设积分表结构如下：队名，已比赛的场数，赢的场数，负的场数，赢得局数，积分。积分表放在正文文件中。通过用户界面将最近一轮的结果输入，其形式为：主队名（可用代码），客队名（可用代码），主队得分，客队得分。程序应根据此轮结果修改各队的积分和名次，所得的最新记分表仍在原积分文件中并同时也在屏幕上显示。

测试数据：

可选择我国当年的排球联赛的数据输入，并检查与网上公布的数据是否一致。

实现提示：

定义一个球队类，每个球队均是此类的对象。由于联赛中参赛的队伍数是固定的，因此可用对象数组来实现（当然也可以用链表结构）。每输入两个队的比赛成绩，则相应的队的有关数据（比赛场数，赢的场数，负的场数，赢得局数，积分等）即可进行修改，比赛成绩录入完成，调用联赛排序方法（对象数组作为参数）排出名次并输出。

题目26：单人跳台跳水比赛

问题描述：

比赛进行预赛、半决赛和决赛。预赛中成绩最好的18名选手进入半决赛。半决赛中成绩最好的12名选手进入决赛。预赛、半决赛、决赛的成绩不相互累加。

跳水比赛动作选自跳水规则制定的“动作难度表”，一般包括自选动作（有难度系数）和无难度系数限制的动作。每轮比赛由7个裁判根据运动员的起跳、空中动作、入水动作打分，基础分为：去掉两个最高分，去掉两个最低分，剩下3位裁判打分求和，然后乘以难度系数。共跳8轮，最后把运动员的每轮得分相加，然后进行排名。若有选手得分相同，则抽签或打附加赛决定名次（这在联赛结束后进行，联赛未结束则两队名次并列，本程序不做这方面要求）。试编一程序统计最近一轮比赛后，各队积分及排名。

基本要求：

设积分表结构如下：选手名，预赛总分，半决赛总分，决赛总分。积分表放在正文文件中。通过用户界面将最近一轮的结果输入，其形式为：选手名（可用代码），通过动作难度系数和各裁判所打分数计算出的总分。程序应根据此轮结果修改各选手的积分和名次，所得的最新记分表仍在原积分文件中并同时也在屏幕上显示。

测试数据：

可选择我国当年的游泳世锦赛的数据输入，并检查与网上公布的数据是否一致。

题目27：链表问题处理

（1）反转

设计一个程序，实现链表的反转。给定一个单链表的头节点，将其反转，并返回反转后的链表头节点。

问题输入：链表的头节点

问题输出：反转后的链表头节点

实现要求：设计一个函数，接受链表的头节点作为输入，并返回反转后的链表头节点。

（2）删除链表目标值

设计一个程序，删除链表中值为给定数值的所有节点。给定一个单链表的头节点和一个目标值，删除链表中所有值为目标值的节点，并返回删除节点后的链表头节点。

问题输入：链表的头节点目标值

问题输出：删除节点后的链表头节点

实现要求：设计一个函数，接受链表的头节点和目标值作为输入，并返回删除节点后的链表头节点。

（3）合并有序链表

设计一个程序，合并两个有序链表。给定两个有序链表的头节点，将其合并为一个有序链表，并返回合并后的链表头节点。

问题输入：两个有序链表的头节点

问题输出：合并后的有序链表头节点

实现要求：设计一个函数，接受两个有序链表的头节点作为输入，并返回合并后的有序链表头节点。

问题涉及链表的基本操作，可以通过定义链表节点和使用指针操作来实现。

题目28：图书借阅管理系统

图书借阅管理信息包括：图书名称、图书编号、单价、作者、存在状态、借书人姓名、性别、学号等功能描述

1) 新进图书基本信息的输入。

2) 图书基本信息的查询。

3) 为借书人办理注册。

4) 办理借书手续。

5) 办理还书手续

要求：使用二进制文件方式存储数据，系统以菜单方式工作

（1）需求分析

图书管理信息采用文件保存，因而要提供文件的输入输出操作；要实现对图书基本信息的查询，则需提供查找操作（提供按书名、作者名查询两种查询方式）和显示操作；要实现对撤销图书信息的删除则需提供文件记录的删除操作；办理借书或还书手续需要提供修改操作；另外还要提供键盘式选择菜单以实现功能选择。

（2）总体设计

整个系统被设计为数据输入模块、数据查询模块、数据删除模块和数据修改模块。

（3）详细设计

数据结构采用结构体，设计图书信息结构体

1) 数据输入模块

采用fwrite或fprintf把图书基本信息写入图书信息文件。

2) 数据查询模块

通过菜单选择查询方式，提供按书名查询和按作者名查询两种查询方式。采用基本查找算法即可。

3) 数据删除模块

通过菜单选择删除操作，由于C语言没有提供直接删除文件记录的函数，因而需要自己实现：读记录，判断是否要删除（与输入的要删除的记录比较），如果要删除，则舍弃；否则重新写入文件。

4) 数据修改模块

办理借书手续时，先要在文件中查找要借的书的存在状态，若可以出借，则允许借出，把借书人姓名、性别和学号添加到该书的记录中，并把存在状态改为已经借出；否则，显示“抱歉，该书已借出”。

办理还书手续时，在文件中查找该书的记录，然后把存在状态改为允许出借，把借书人姓名、性别和

学号删除。具体修改操作可以参考删除操作

题目29：航班信息管理系统

问题描述：

飞机航班系统的数据包括两部分：

- ① 航班信息：航班号、最大载客数、起飞地点、起飞时间、降落地点、降落时间，单价；
- ② 乘客信息：航班号、身份证号码、姓名、性别、出生年月、座位号。乘客订票的主要方式是：乘客提出航班号、起飞地点、起飞时间、降落地点、订票数等订票要求，根据事先保存的航班数据决定乘客能否订票？只有全部满足了乘客的订票要求并且所订航班有足够的未订座位之后才能完成订票处理，并且修改该航班的未订座位数（每个航班的未订座位数的初始值就是该航班的最大载客数）；否则，订票失败，并且给出不能订票的原因。

要求将航班数据保存在数据文件中，在处理时按航班的起飞地点建立不同的链表。

功能要求：

- (1) 增加航班记录。将新的航班记录增加到原有的航班数据文件中。在进行处理时必须检查所要增加的航班记录是否存在，如果已经存在，应给出提示信息后停止增加；
- (2) 航班取消。如果某次航班的乘客数太少（已订票的少于本次航班最大载客数的10%），将取消该航班，但该航班的记录仍然保存在原有的航班数据文件中；
- (3) 航班查询。应该有以下几种基本的查询方式：按航班号、按起飞地点和起飞时间、按降落地点，按起飞地点和降落地点；
- (4) 航班订票。按上述问题描述中的乘客订票方式完成航班订票处理。
- (5) 设计一个菜单，至少具有上述操作要求的基本功能

题目30：女子体操团体比赛

预赛采用543赛制（即五个正式选手，每个项目上四个人，其中的三个成绩计入有效分），决赛采用533赛制（即五个正式选手，每个项目上三个人，三个成绩全部计入有效分）。有A、B两组裁判进行评分。

A分为A组裁判打出的难度价值分，根据运动员成套动作中最好的9个动作加上下方法共10个动作，计算难度分值，动作组别从A—G，分值0.1—0.7。A组裁判判断连接价值，并根据动作结构组的完成情况判断**加分**。

• B分艺术、技术、编排、完成情况出现的错误的**扣分**。小错0.1分，中错0.3分，大错0.5分，掉下0.8分。10分减掉艺术扣分、完成错误扣分与技术及编排错误扣分得到最后的B分。

两组裁判给分都是独立的，最高分和最低分将被去掉，然后算出的平均成绩就是每组的打分数。A分和B分加起来为一个项目的最后得分。最后取一个团队在所有项目上的成绩的总和作为该参赛队的团体决赛成绩。

如果两队得分一样，则先看B分，B分高的胜出；如果B分都一样，那么在有效的B分中去掉一个最低的，把剩下的B分平均后比较，直到减到只剩下一个分数。若仍有队伍得分相同，则抽签决定名次（这在比赛结束后进行，比赛未结束则两队名次并列，本程序不做这方面要求）。试编一程序统计最近一轮比赛后，各队积分及排名。

基本要求：

设积分表结构如下：队名，积分。积分表放在正文文件中。通过用户界面将最近一轮的结果输入，其形式为：队名，积分。程序应根据此轮结果修改各队的积分和名次，所得的最新记分表仍在原积分文件中并同时在屏幕上显示。

测试数据：

可选择我国当年的体操锦标赛的数据输入，并检查与网上公布的数据是否一致。

题目31：汽车客运公司管理系统软件

基本要求

使用面向对象的系统分析和设计，开发基于MFC对话框或者单文档界面的C++语言应用软件。

任务陈述

客车的班次任务由调度部门确定并输入数据，一般在一段时间内不作调整。每个班次的基本信息包括：班次号、车型、发车时间、终点、座位数量、票价等。

- (1) 旅客购票时，应登记身份证号码、购票日期、发车日期、车次、座位号等信息。
- (2) 旅客购票时，可以查询指定发车日期、目的地的客车班次信息，在查询到的班次中，如果还有未售座位，就可以买票。旅客可以在未售座位中选择座位，也可由系统自动选择座位。购票时也可直接输入发车日期、目的地和班次，由系统自动出票，如果无票可售，则系统给与提示。座位不能重复销售，不允许售无座票。
- (3) 系统中应该保有从当天算起的3天的票源数据，开始时创建今，明，后三天的，以后每天创建后天的，每天的票源数据，应根据调度计划安排。
- (4) 每天的每趟班车在发售第1张车票时，创建这个班次的旅客登记表。
- (5) 旅客可以办理售票，售票时在旅客登记表中添加旅客信息（座位号不能重）。
- (6) 旅客可以办理退票，退票时即在旅客登记表中删除旅客信息。在开车前退票收取20%退票费，开车后退票收取50%退票费。
- (7) 旅客可以办理改签，在开车前可以改签同一目的地的其他车次（3天以内），不收改签费，开车后收20%改签费。
- (8) 可以输出指定班次的旅客登记表。表中包括该班次的票款合计。

加分项：1) 使用Access数据库存储数据，使用SQL语句操作数据；

2) 使用曲线或者图表展示数据

题目32：基于C++的网络聊天室开发

基本要求：

设计一个简单的网络聊天室程序，可以让用户在客户端输入消息并发送到服务器端，其他用户可以在客户端看到服务器端广播的消息。要求使用TCP/IP协议进行通信，并实现简单的用户身份验证、消息广播等功能。

任务陈述：

1. 创建Socket。使用socket()函数创建一个Socket，指定Socket的类型（TCP/IP协议对应SOCK_STREAM）。
2. 绑定Socket到本地地址。使用bind()函数将创建的Socket绑定到本地地址和端口上。
3. 监听连接。使用listen()函数监听客户端连接请求。
4. 接受连接。使用accept()函数接受客户端的连接请求，并获取一个新的Socket来进行通信。
5. 接收和发送数据。使用send()函数向客户端发送数据，使用recv()函数从客户端接收数据。
6. 关闭Socket。使用close()函数关闭Socket。

在使用Socket进行通信时，需要遵守TCP/IP协议中的一些规则，比如应该使用正确的地址族、正确的IP地址、正确的端口等等。此外，还需要处理一些异常情况，比如当客户端断开连接时，服务端需要及时关闭相应的Socket。

题目33：基于C++的音乐播放器实现

设计思路：

- (1) 播放器有3个状态：停止状态、播放状态、暂停状态
- (2) 播放器有3个按键：播放按键、暂停按键、停止按键
- (3) 播放器启动后进入到停止状态，停止状态下只响应播放按键，按下播放按键进入播放态，并开始播放音乐；播放状态下，不响应播放按键，按下暂停键暂停音乐播放，按下停止键停止音乐播放；暂停状态下，不响应暂停按键，按下播放键继续播放音乐，按下停止键停止播放音乐。

补充说明：

本案例重点体现设计思想和编程思想

界面不限，但是必须基于图形界面，界面越好看分越高

不需要真正能播放音乐，通过输出不同的字符串来表示执行了相关动作即可，比如：播放开始打印“Playing”字符串，停止时打印“Stop”字符串

题目34：基于C++的中国象棋实现

设计思路：

- (1) 确定规则：中国象棋的基本规则是很简单的，但是在实现过程中，我们还需要考虑一些特定情况，比如：将帅不对面、过河兵等。可以先画出游戏棋盘，并根据规则画出棋子的走法，边画边思考，形成一份详细的游戏规则。
- (2) 设计数据结构：在C++ 中，我们可以使用结构体或者类来表示棋子、棋盘等信息。对于单个棋子，我们可以使用结构体来表示其属性，比如名称、颜色、坐标等；对于整个棋盘和游戏状态，可以使用类来表示。
- (3) 实现游戏逻辑：接下来我们开始实现游戏的逻辑。首先，需要实现玩家交互的部分。这包括输入、输出和移动棋子等操作。如果你是初学者，可以先实现基本的游戏逻辑，逐步添加高级功能。
- (4) 实现游戏图形界面：最后，你可以使用任何一个C++ GUI库来实现游戏界面。一些常用的GUI库包括 Qt、MFC 和 wxWidgets

补充说明：

如果可以实现电脑AI加分：如果你想让游戏更具有挑战性，你可以实现电脑AI来增加游戏难度。在实现电脑AI时，需要考虑棋局的整体形势和下一步可行的走法等。

题目35：基于朋友圈的商品推荐系统

设计思路：

模拟基于朋友圈的商品推荐，可以用于任何电商，如淘宝、京东等。

假设目前用户有100人，每人均有自己的朋友圈，且购买过若干商品(50种)。实现基于朋友圈的商品推荐，要求根据朋友们购买过的商品，为用户推荐其可能感兴趣的物品。

具有用户和商品添加、删除、查找和排序，以及商品推荐等功能。通过此课题，熟练掌握文件、数组、指针的各种操作，以及一些算法思想的应用。

(1) 管理信息应包括：

用户信息(用户名、密码、朋友圈、所购商品等)、商品信息(商品名和购买次数等)和朋友圈

(2) 需要实现的功能

用户输入用户名和密码完成注册和登录

实现相关信息的建立，包括录入、修改和删除

对用户和商品进行查询

对商品进行排序

根据朋友购买的信息，为用户推荐商品

将所有相关信息内容存为文件

(3) 推荐算法：统计朋友们购买的所有商品，累加每种商品的数量并排序。当然，如果有其他你认为更优的方法更好！

可以建立二维数组保存用户与用户朋友圈信息，以及用户与商品之间购买信息。

题目36：单位的考勤管理系统

系统保存每个员工的考勤情况（姓名，编号，所在部门，职位，日期，当天是否缺勤，当天缺勤的类型（迟到，早退，请假，旷工，出差，其它））。要求在系统中用到面向对象的设计方法。

系统具有如下功能：

- （1）系统以菜单方式工作
- （2）信息录入功能（保存在文件中）
- （3）浏览功能
- （4）查询功能
- （6）统计某段时间内，缺勤员工的姓名及缺勤次数
- （7）统计某段时间内，某种类型缺勤员工的姓名及缺勤次数（比如统计所有旷工的员工）
- （8）信息的删除与修改

题目37：英语单词学习系统

系统建立英汉词典（有限的一部份单词，比如50个单词左右）。系统能实现单词的查找、增加，删除，修改功能。要求在系统中用到面向对象的设计方法。系统具有如下功能：

- （1）系统以菜单方式工作
- （2）信息录入功能（保存在文件中）
- （3）浏览功能
- （4）查询功能
- （5）删除功能
- （6）修改功能
- （7）游戏功能，先提示用户登录，选择答题的题目数目。然后屏幕上随机出现一个单词（英文或中文），提示用户给出答案，答对给分。游戏结束后，给出总分。显示排行榜，排行榜为得分前10名的用户名和得分。

题目38：医院挂号管理系统—就诊者管理系统。

系统创建病人信息数据和医生信息数据。就诊者信息包括：姓名，性别，出生年月，电子病历，挂号信息。医生信息包括，姓名，性别，职称，科室。（医生的信息已经保存在文件中）。要求在系统中用到面向对象的设计方法。

系统具有如下功能：

- （1）系统以菜单方式工作
- （2）信息录入功能，就诊者可以注册自己的个人信息。
- （3）修改功能，就诊者可以修改自己的个人信息。
- （4）查询功能，就诊者可以查看自己的个人信息，包括挂号信息，病历
- （5）挂号功能，就诊者可以查询某科室的所有医生，然后决定挂某个医生的号。

题目39：医院挂号管理系统—医生管理系统。

系统创建病人信息数据和医生信息数据。就诊者信息包括：姓名，性别，出生年月，电子病历，挂号信息。（就诊者的信息已经保存在文件中）。医生信息包括，姓名，性别，职称，科室。要求在系统中用到面向对象的设计方法。

系统具有如下功能：

- （1）系统以菜单方式工作
- （2）信息录入功能，医生可以注册自己的个人信息。
- （3）修改功能，医生可以修改自己的个人信息。
- （4）查询功能，医生可以查看自己的个人信息
- （5）病历填写功能，医生查看到所有挂自己号的就诊者信息，并可以填写就诊者的病历

题目40：银行排队系统

某银行有n个窗口开展用户业务，用户需要排队办理业务。用户到达银行时先拿到排队号码，并

能知道需要等待的人数。如果是VIP用户则直接到VIP窗口，不用加入普通用户的等待。要求在系统中用到面向对象的设计方法。

系统具有如下功能：

- (1) 系统以菜单方式工作
- (2) 拿号功能，选择用户类型（VIP, 普通），再输入卡号，再拿自己的排队号码
- (3) 评价功能，对所在窗口的服务进行评价（根据满意度选择相应的项目）
- (4) 业务查询，查询每个窗口正在给几号用户服务
- (5) 排队查询，查看前面有多少人在排队

题目41：课堂测试系统

设计一个课堂测试系统。要求在系统中用到面向对象的设计方法。

系统具有如下功能：

- (1) 系统以菜单方式工作
- (2) 试题库录入功能，能录入单项选择题，单选题。
- (3) 试题库维护功能，按题目进行查询，然后修改题目及答案。
- (3) 测试功能，用户先登录，然后系统随机分配题目，进行测试
- (4) 查询功能，用户先登录，可以查询自己的得分，以及错题项

题目42：任务管理系统

问题描述：

设计并实现一个简单的任务管理应用程序，用于帮助用户管理任务和日程安排。每个任务记录包括任务名称、截止日期、优先级和状态。

功能要求：

添加任务：用户可以输入任务的名称、截止日期、优先级和状态（未完成、已完成等）。

显示所有任务：在屏幕上显示所有任务的信息，包括任务名称、截止日期、优先级和状态。

修改任务信息：可以修改任务的截止日期、优先级和状态。

删除任务：用户可以删除指定的任务。

查询任务信息：根据任务名称或状态查找任务的相关信息，如果找到，显示任务名称、截止日期、优先级和状态。

提供存储和装入功能，即将任务信息保存在一个文件中，以便以后装入并重新使用。

题目43：骑士巡游问题

问题描述：

编写程序求解骑士巡游问题：在 n 行 n 列的棋盘上（如 $n=5$ ），假设一位骑士（按象棋中“马走日”的行走法）从初始坐标位置 (x_1, y_1) 出发，要遍访（巡游）棋盘中的每一个位置一次。请编一个程序，为骑士求解巡游“路线图”（或告诉骑士，从某位置出发时，无法遍访整个棋盘——问题无解）。

当 $n=5$ 时，意味着要在5行5列的棋盘的25个“点”处，按骑士行走规则，依次将1至25这25个“棋子”（数码）分别摆放到棋盘上（摆满25个位置则成功，否则失败问题无解）。例如，当 $n=5$ 且初始坐标位置定为 $(1, 1)$ ——即最左上角的那个点时，如下是一种巡游“路线图”。程序执行后的输出结果为： $(x_1, y_1)? \Rightarrow (1 \Rightarrow 5, 1 \Rightarrow 5) : 1\ 1$

1	6	15	10	21
14	9	20	5	16
19	2	7	22	11
8	13	24	17	4
25	18	3	12	23

提示：

(1) “棋盘”可用二维数组B表示。

(2) 编制一个具有如下原型的递归函数solve，用于完成任务：从(i, j)点出发，做第k至第n*n（即n的平方）次的移动——将k直到n的平方这些数码按规则分别摆放到棋盘即数组B中，若成功则通过引用参数ok返回true，否则返回false。

```
void solve(int i, int j, int k, bool& ok);
```

(3) 编制主函数，让用户输入作为巡游起点的初始坐标位置(x1, y1)，在该处摆放“棋子”（数码）1，而后进行调用“solve(x1, y1, 2, ok);”来完成所求任务。

欲处理的初始问题为：从某点(x1, y1)出发，按所给行走规则，作24次移动，遍访棋盘中未被访问过的各点（或发现无路可走）。

可分解简化为如下两个子问题（正是形成递归函数的基础）：

① 由点(x1, y1)出发，按所给行走规则作1次移动到达(g, h)（或发现无路可走）；

② 从(g, h)点出发，按所给行走规则，作23次移动，遍访棋盘中未被访问过的各点（或发现无路可走）。

solve函数具体实现时，若由(i, j)点出发已“无路可走”，则将引用参数ok置为false而递归出口；否则，先“迈一步”到达(g, h)点，而后再进行递归调用：solve(g, h, k+1, ok)；以实现从新点(g, h)出发，将k+1直到25这些“棋子”（数码）分别摆放到棋盘上，若成功则通过引用参数ok返回true（否则返回false）。

点评：

(1) 也可编制第二种解法的主函数：将棋盘上的n平方个点依次作为巡游起点的初始坐标位置(x1, y1)，判断从每一位置出发是否有解或无解（输出“OK!”或“NO!”，但并不输出“路线图”）。

(2) 若更改程序中的n值（如改为4或6等），便可求解其他阶数的巡游“路线图”。

(3) 可改用非递归方法设计并编写solve函数，那样的话，通常要增加一个记录摆放“棋子”信息的数组，可记录下是沿着什么方向到达了当前的什么位置（在那儿摆放了“棋子”）等，而

且对上述数组可按照栈(stack)的方式来使用（栈总是采用FILO即所谓的先进后出使用方式），以便在“无路可走”的情况下，回退（回溯）到上一个位置，接着按照另外的方向去寻找其他的“行走”方法。

题目44：模拟电影院订票系统

要求实现以下功能：

- 定义一个电影类，包含电影名称、类型、时长、评分、票价等属性，以及构造函数。
- 定义一个影厅类，包含影厅编号、座位数、座位分布等属性，以及构造函数。
- 定义一个场次类，包含场次编号、电影对象、影厅对象、放映时间等属性，以及构造函数。
- 定义一个订单类，包含订单编号、场次对象、订票人姓名、订票人电话、订票数量、订票座位等属性，以及构造函数。
- 使用文件输入输出操作，实现数据的持久化存储。
- 使用链表或者向量来存储电影对象、影厅对象、场次对象和订单对象，并实现相应的管理功能。
- 使用堆栈或者队列来模拟电影院的排队购票和退票的过程，并实现相应的管理功能。
- 使用快速查找算法（如散列表或者二叉搜索树）来实现电影名称和场次编号的快速查找，并实现相应的管理功能。

- 使用图来表示电影院的座位分布，并实现相应的管理功能。
- 使用排序算法来对电影对象或者场次对象进行排序，并实现相应的管理功能。
- 使用异常处理机制，捕获并处理可能出现的错误情况，如输入非法数据、文件打开失败等。
- 使用命名空间来组织代码，避免命名冲突。
- 使用友元函数或友元类来访问私有成员。
- 使用静态成员变量或静态成员函数来实现一些共享功能。

题目45：模拟餐厅点餐系统

实现以下功能：

- 定义一个餐品类，包含餐品编号、餐品名称、餐品价格、餐品评分等属性，以及构造函数。
- 定义一个订单类，包含订单编号、订单时间、订单金额、订单状态等属性，以及构造函数。
- 定义一个顾客类，包含顾客编号、顾客姓名、顾客电话等属性，以及构造函数。
- 定义一个餐厅类，包含一个餐品对象的容器，一个订单对象的容器，一个顾客对象的容器，以及添加、删除、修改、查询、排序等方法。使用文件输入输出操作，实现数据的持久化存储。

要求实现以下主要功能

- 显示所有餐品的信息和评分
- 根据餐品编号或餐品名称查找餐品的信息和评分
- 添加或删除餐品的信息和评分
- 修改餐品的信息和评分
- 显示所有订单的信息和状态
- 根据订单编号或订单时间查找订单的信息和状态
- 添加或删除订单的信息和状态
- 修改订单的信息和状态
- 显示所有顾客的信息和电话
- 根据顾客编号或顾客姓名查找顾客的信息和电话
- 添加或删除顾客的信息和电话
- 修改顾客的信息和电话
- 使用异常处理机制，捕获并处理可能出现的错误情况，如输入非法数据、文件打开失败等。
- 使用命名空间来组织代码，避免命名冲突。
- 使用友元函数或友元类来访问私有成员。
- 使用静态成员变量或静态成员函数来实现一些共享功能。

题目46：八皇后问题

编写程序对八皇后问题进行求解：在 8 行 8 列的棋盘上放置 8 个皇后，使任一个皇后都不能吃掉其他的 7 个皇后（注：皇后可吃掉与她处于同行或同列或同一对角线上的其他棋子），并将结果以某种方式显示出来。

例如，当求出下述的一个解时，可输出如下信息来表示该解（输出了表示摆放皇后的坐标位置以及“棋盘状态”——棋盘中有皇后的位置放一个“Q”字符，其他位置为“+”字符）。

(1, 1) (5, 2) (8, 3) (6, 4) (3, 5) (7, 6) (2, 7) (4, 8)

```
Q + + + + + + +
+ + + + + Q +
+ + + + Q + + +
```

```

+ + + + + + Q
+ Q + + + + +
+ + + Q + + +
+ + + + + Q +
+ + Q + + + +

```

提示:

(1) 通过 “`int LineNum[9]; bool a[9], b[15], c[15];`” 说明具有全局作用域的 4 个数组。其中的: `LineNum[i]` 表示第 i 列的皇后要放的行位置 (只用其中的列号 1 到 8); `a[i]` 为 `true` ($i = 1, 2, \dots, 8$) 表示第 i 行上尚未放皇后; `b[i]` 为 `true` ($i = 0, 1, 2, \dots, 14$) 表示第 i 条斜对角线上尚未放皇后 (斜对角线指的是 “/” 状对角线, 该对角线上各点的行列号之和 $i+j$ 为一个常数);

`c[i]` 为 `true` ($i = 0, 1, 2, \dots, 14$) 表示第 i 条反斜对角线上尚未放皇后 (反斜对角线指的是 “\” 状对角线, 该对角线上各点的行列号之差 $i-j$ 为一个常数)。

从而当使用语句 “`if (a[j] && b[i+j-2] && c[i-j+7]) LineNum[i]=j;`” 时, 可用于判断并实现: 如果在第 j 行的第 i 列上放置皇后安全的话, 则将一枚皇后放置到那儿。

(2) 编制一个具有如下原型的递归函数 `solve`, 它负责往第 i 列开始的连续 $8-i+1$ 列上均放上皇后, 若成功则通过引用参数 `ok` 返回 `true` (否则返回 `false`)。

```
void solve(int i, bool& ok);
```

摆放皇后之后, 若 $i=8$ 即已放满时则递归出口; 否则通过 `solve(i+1, ok)` 进行递归调用。

(3) 编制主函数, 首先初始化一个 “空棋盘”, 即将 `a`、`b`、`c` 数组的各元素均置为 `true` (表示当前棋盘的 8 个行、15 条斜对角线以及 15 条反斜对角线上都尚未摆放皇后)。而后执行调用语句 “`solve(1, ok);`”, 它负责往第 1 列开始的连续 8 列上均放上皇后, 若成功则通过引用参数 `ok` 返回 `true` (否则返回 `false`)。

点评:

(1) 可改用非递归方法设计并编写 `solve` 函数, 那样的话, 通常要设置数组来记录皇后的摆放位置信息, 还要记录这些皇后所产生的 “影响面” (所建立的 “势力范围”) — 使得哪些行列位置不可再摆放皇后。当在新的行列位置摆放了皇后、但此时又无法进一步摆放其他的皇后时, 要回退 (回溯) 到上一个位置接着去考虑另外的 “行走” 方法 (若还有的话) 等等。但注意, “回退” 一步后, 要同时 “撤销” 由于该步的回退而关联的那些 “影响面” (释放 “势力范围”)。

(2) 本程序只是找到了某一种 “摆放方案” 而终止, 还可进一步考虑寻找其他各种不同的 “摆放方案” (实际上共有 92 种)。

(3) 也可用同样的方法去处理其他 “阶数” 的皇后问题, 如求解四皇后问题等。

题目47: 设计一个外卖系统

定义一个外卖品类, 包括外卖的价格, 外卖的编号, 外卖的名称, 外卖的归属店铺, 外卖接送的外卖员, 该品外卖员的收益

定义一个外卖员类, 包括外卖员的编号, 外卖员的名称, 外卖员的收益, 外卖员拥有的外卖品, 外卖员的路线

定义一个用户类, 包括用户编号, 用户地址, 用户购买的外卖以及时间

定义一个订单类, 包括订单编号, 订单时间, 订单的用户和外卖员, 订单的目的地地址, 订单送达的时间, 订单最晚送达时间

注意:

地址用图表示, 节点之间距离为 1

动态调整外卖员路线, 随时添加订单, 寻找路径, 既可以不迟到送单, 又可以最短路径

规则: 外卖按时送达不扣钱, 延迟扣费

题目48：设计一个高铁系统

定义一个高铁类，具有高铁编号，高铁承载人数，高铁路线节点以及不同节点的到达和停留时间，高铁拥有座位

定义一个乘客类，具有乘客编号，乘客车票，乘客姓名，乘客身份证号，乘客联系方式

定义一个车票类，具有车票编号，车票所属车次，上车节点时间，下车节点时间，座位编号，车票所属用户

定义一个座位类，具有所属车次，该节点或时间的乘客

注意：

高铁路线用图表示

该乘客是否可以上车

该乘客座位是否有人

该乘客上车地点时间是否与车票对应

显示不同时刻不同车次分别的用户以及行驶的节点位置

题目49：栈及其操作

问题描述：

栈(Stack)：是限制在表的一端进行插入和删除操作的线性表。又称为后进先出 LIFO (Last In First Out)或先进后出 FILO (First In Last Out)线性表。

栈顶(Top)：允许进行插入、删除操作的一端，又称为表尾。用栈顶指针(top)来指示栈顶元素。

栈底(Bottom)：是固定端，又称为表头。空栈：当表中没有元素时称为空栈。

设栈 $S=(a_1, a_2, \dots, a_n)$ ，则 a_1 称为栈底元素， a_n 为栈顶元素。栈中元素按 a_1, a_2, \dots, a_n 的次序进栈，退栈的第一个元素应为栈顶元素 a_n 。即栈的修改是按后进先出的原则进行的。

栈的链式存储结构称为链栈，是运算受限的单链表。其插入和删除操作只能在表头位置上进行。链栈的基本形式如下：



实现要求:

- (1)链栈基本操作的实现: 栈的初始化, 生成一个空栈; 压栈, 即元素进栈; 弹栈, 即元素出栈;
- (2)十进制整数N 向其它进制数 d(二、八、十六)的转换是计算机实现计算的基本问题。转换法则: 该转换法则对应于一个简单算法原理:

$$n = (n \text{ div } d) * d + n \text{ mod } d$$
 其中: div 为整除运算, mod 为求余运算

- (3)在文字处理软件或编译程序设计时, 常常需要检查一个字符串或一个表达式中的括号是否相匹配?

匹配思想:

从左至右扫描一个字符串(或表达式), 则每个右括号将与最近遇到的那个左括号相匹配。则可以在从左至右扫描过程中把所遇到的左括号存放于堆栈中。每当遇到一个右括号时, 就将它与栈顶的左括号(如果存在)相匹配, 同时从栈顶删除该左括号。

算法思想:

设置一个栈, 当读到左括号时, 左括号进栈。当读到右括号时, 则从栈中弹出一个元素, 与读到的左括号进行匹配, 若匹配成功, 继续读入; 否则匹配失败, 返回 FLASE。

题目50: 研究生初试录取

问题描述:

研究生考试课程为 4 门, 其中数学、外语、政治为统一命题, 而专业基础课则根据不同专业由招生学校自行命题。国家对初试录取分数有总分要求(如某一年要求4 门课总分应达到310 分), 另外还有对每门课的最低分数要求(如总分为 100 的试卷最低应达到 40 分, 总分为 150 的试卷最低应达到 65 分)。编程统计初试合格的人数, 并按总分由高到低的顺序输出合格考生的信息。

基本要求:

程序运行时首先要求输入: 考生姓名, 准考证号, 报考专业, 是否应届生, 4 门课程(政治、数学、外语、专业基础课)成绩。这些原始数据应保存到一个文件中。然后输入: 录取的总分要求, 各课程的最低分数要求。输出要求: 过线考生的姓名, 准考证号, 报考专业, 是否应届生, 4 门课程(政治、数学、外语、专业基础课)成绩及总分, 这些信息应存放到另一个文件中。

测试数据:

程序应输入不少于 10 名考生的信息, 其中应届生和历届生分别有若干名, 并且都有合格和不合格的情况。

实现提示:

可定义一个考生类存放有关信息和实现相应的操作。分数线数据(总分要求和各门课程的要求)可定义另外的类来存放, 但应能被考生类及其派生类直接访问。

选做内容:

初试合格的考生应经过复试才能决定是否录取, 复试成绩合格(大于一给定分值)可以录取, 否则被淘汰。而录取的顺序假设是按照专业基础课和复试成绩的平均值来确定的(因为

这涉及到是计划内还是委培问题）。因此，应首先输入初试合格考生的复试成绩及复试的合格线分数，然后按上面要求排序输出并标明被淘汰的学生。

题目51：设计 Database 类

问题描述：

设计一个Database类。Database类是一个表的集合，而表又由行和列组成。例如，下面的 雇员信息表包含三个记录，每个记录有四个字段（Employee、Name、Department和Boss）。

雇员	姓名	部门	部门经理
111-11-1234	Cruz	ACC	Warder
213-44-5649	Johnston	MIS	Michaels
321-88-7895	Tom	FIN	Bearskin

基本要求：

能够完成对数据库的基本操作：包括创建数据库，实现对数据库里面的表的添加，删除；以及能够完成对表结构的修改（如添加或删除字段），以及对表中的记录进行添加和删除；能够完成对使用适当的查询语言从一个或多个表中查找相关信息。

测试数据：

可以选取高校人员（教师、学生、职工等）数据库作为测试数据

实现提示：

Database 类的公有接口包含如下数据成员：

- 创建一个表。
- 通过增加或删除字段修改表结构。
- 删除一个表。
- 在表中增加一个记录。
- 从表中删除记录。
- 用适当的查询语言从一个或多个表中查找信息。

选做内容：

高校人员信息通常保存在文件里面，可以从文件里面读取人员信息；然后完成上面操作。

题目52：电煤气管理系统

问题描述：

设计一个水电管理信息系统，能够对高校的水电费用进行管理,包括了登记费用，查询费用，以及住户信息管理等。在设计时要考虑到学生和教工在用水电时的不同，学生可以免费使用一定额度的水电，超过这个额度的随便以后必须自费使用，且自费部分水电费的价格标准要高于教工的收费标准（主要是节约资源）。

基本要求：

- （1）实现对用户信息的录入
- （2）实现水电煤气数据的录入
- （3）计算并查询用户应缴费用
- （4）查询未缴纳费用名单

测试数据：

可选用小区物业管理部的数据，也可采用手工输入不少于 10 个用户信息的数据。

实现提示:

(1) 用户基本信息类, 教工用户信息类, 学生用户信息类

(2) 收费标准类, 该类存储水电煤气标准单位的收费标准, 如: 煤气, 1.0 元/立方米;

(3) 不同类型人员水、电、煤气信息类, 这些类可以包括水表, 电表, 煤气表 ID, 抄表时间, 上次抄表时间, 本次抄表时间, 上次抄表度数, 本次抄表度数, 本次使用度数, 费用, 是否缴费标记等信息。

(4) 在实现的时候考虑继承和多态技术的合理使用。

选做内容:

实现用户查询系统能够找出该用户半年之内的交费记录和本次应交费的数据。

题目53: 数据汇总

问题描述:

在数据处理中经常需要对大量数据进行汇总, 将相同关键字记录的某些数据项的值叠加起来, 生成一个分类汇总表。

假设某超级市场销售有 m 种商品 (假设商品的编号为 1, 2, 3, ..., m), 有 n 台前台收款机 (假设收款机的编号为 1, 2, 3, ..., n) 进行收款, 以记录的形式提供给计算机, 每个记录表示某台收款机的一种商品一次交易的数量和销售额。记录由 4 个域组成: 收款机编号、商品编号、销售数量、销售金额。构造一个结构体类型, 每次销售数据以一个结构体变量保存在一个数据文件中。

实现要求:

(1) 编写实现将数据记录插入到数据文件的最后的函数;

(2) 编写以收款机为单位的数据分类处理函数。构造 n 个单链表, 每个链表保存一台收款机的销售记录, 这 n 个单链表的头指针存放在一个指针数组中, 通过数组的下标就可以知道是哪台收款机。读取数据文件的记录, 将所有的销售记录 (数据文件中的全部记录) 分解插入到 n 个单链表;

(3) 编写以商品为单位的数据分类处理函数。构造 m 个单链表, 每个链表保存一种商品的销售记录, 这 m 个单链表的头指针存放在一个指针数组中, 通过数组的下标就可以知道是哪种商品。读取数据文件的记录, 将所有的销售记录 (数据文件中的全部记录) 分解插入到 m 个单链表;

(4) 统计每台收款机的销售总额;

(5) 以收款机为单位, 将所有收款机按销售总额的非递减顺序构造一个单链表并输出;

(6) 以商品为单位, 统计每种商品的销售总额;

(7) 以商品为单位, 将所有销售的商品按销售总额的非递减顺序构造一个单链表并输出;

(8) 设计一个菜单, 具有上述要求的所有功能、退出系统等最基本的功能。

题目54: 商品管理

某商场有如下的几种货品: 衬衣、鞋子、帽子、裤子、冰箱、电视、立柜、壁橱、沙发。每一种货物都有详细的说明信息。

衬衣: 布料、尺寸、单价、产地、库存量、所属货柜;

鞋子: 皮料、尺寸、单价、产地、库存量、所属货柜;

帽子: 布料、样式 (平顶或尖顶)、尺寸、单价、产地、库存量、所属货柜;

裤子: 布料、尺寸、单价、产地、库存量、所属货柜;

冰箱: 制冷类型、样式 (二门或三门)、颜色、尺寸、单价、产地、库存量、重量、所属货柜;

电视: 样式 (彩色或黑白)、颜色、尺寸、单价、产地、库存量、重量、所属货柜;

立柜: 木料、颜色、尺寸、单价、产地、库存量、所属货柜;

壁橱: 木料、颜色、尺寸、单价、产地、库存量、所属货柜;

沙发: 木料、皮料、颜色、尺寸、单价、产地、库存量、所属货柜;

对这些商品的操作有：

新商品的录入，商品的进库，商品的出库，商品的调价，所属货柜的管理，库存的统计，总价的计算，产地的统计。

要求自行设计数据结构，用类结构将上述的货品表示出来。在上一步的基础上，将上述的商品管理计算机化，完成操作要求的功能。

题目55：设计一个排课系统

问题描述：

每位教师都有教学工作量，教师对他所希望讲授的课程表达为一个期望值， $1, 2, \dots, n$ ，其中 1 位最高的期望值。课程也有优先级， $1, 2, \dots, n$ ，用来决定将课程分给教师的顺序，其中 1 是最高的优先级。设计一个程序针对某些课程给某些教师进行排课。

基本要求：

程序运行时，用户输入教师信息（姓名、教师号、工作量）、课程信息（课程名、课程号、周学时、总学时、优先级）、教师对所希望讲授课程的期望值以及教学工作量等相关信息，所有信息应保存在文件中，程序根据课程的优先级以及教师对课程的期望值进行排课。为了公平起见，程序随机分配课程，如果 10 门课程都有优先级 1，程序以随机的顺序将这些课程分给教师，如果 10 位教师对某门课程的期望值为 1，程序应从 10 位教师中随机选择一个。最后输出排课情况，即某位老师上某门课程，以及某位老师的工作量。

测试数据：

程序应能实现对不少于 20 门课和 10 个教师的排课，并且使每个教师的满意度达到最大，不存在工作量不满的教师以及未分配的课等情况。

实现提示：

可定义一个教师类存放教师信息，所有教师的信息可以用数组或链表存储；定义一个课程类存放课程信息，所有课程的信息可以用数组或链表存储；定义一个排课类进行排课，排课类可以访问课程和教师信息。

选作内容：

对两个同类班级安排一学期（18 周）的课程，程序应能处理某些限制，如一个老师不能同时给两个班上课。

题目56：矩阵的压缩存储

问题描述：矩阵是许多科学与工程计算问题中出现的数学对象。在此，我们感兴趣的不是矩阵本身，我们所关心的是研究表示矩阵的方法，以使对矩阵的各种运算能有效地完成。一个矩阵一般由 m 行和 n 列元素组成，一般的 $m \times n$ 阶矩阵，可表示成一个 $m \times n$ 的二维数组，例如 $\text{matrix}[m][n]$ ，需要的存储空间是 $m \times n$

实现要求：

（1）若矩阵中的元素是对称的，即矩阵中第 i 行第 j 列与第 j 行第 i 列元素的值相等，即 $\text{matrix}[i][j] = \text{matrix}[j][i]$ ，我们把这种矩阵称为对称矩阵。对于 $n \times n$ 阶对称矩阵，我们可以为每一对对称元素分配一个存储空间，即只需要存储其下三角（包括对角线）或上三角中的元素即可。这样，就可将 n^2 个元素压缩存储到 $n(n+1)/2$ 个存储单元中。请实现该功能

（2）当一个 $n \times n$ 阶矩阵的主对角线上方或下方的所有元素皆为零时，称该矩阵为三角矩阵。对于三角矩阵，我们同样也可采用对称矩阵的压缩存储方式将其上三角或下三角的元素存储在一维数组中，达到节约存储空间的目的。请实现该功能

除了对称矩阵和三角矩阵等特殊矩阵外，在实际应用中我们还经常遇到这样一类矩阵，存储在矩阵中的大量元素值为零，而且零元素的分布没有规律，这样的矩阵称为稀疏矩阵。对于稀疏矩阵，采用二维数组表示既浪费大量的存储单元来存储零元素，又要花大量的时间进行零元素的运算。为此，我们对稀疏矩阵采取三元组法进行存储。请实现该功能。

题目57：队列及其操作

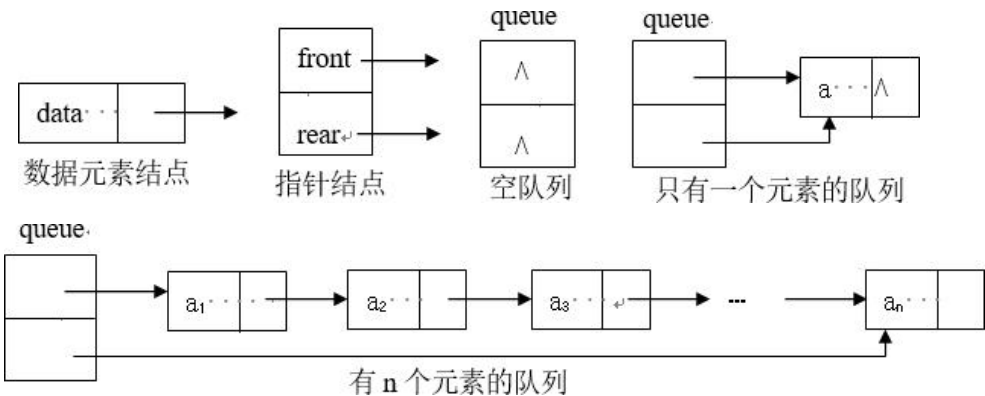
问题描述：

队列(Queue)：也是运算受限的线性表。是一种先进先出(First In First Out，简称 FIFO)的线性表。只允许在表的一端进行插入，而在另一端进行删除。队首(front)：允许进行删除的一端称为队首。队尾(rear)：允许进行插入的一端称为队尾。

队列中没有元素时称为空队列。在空队列中依次加入元素a1, a2, …, an 之后，a1 是队首元素，an 是队尾元素。显然退出队列的次序也只能是a1, a2, …, an，即队列的修改是依先进先出的原则进行的。

队列的链式存储结构简称为链队列，它是限制仅在表头进行删除操作和表尾进行插入操作
 的单链表。

需要两类不同的结点：数据元素结点，队列的队首指针和队尾指针的结点，链队的基本形式如下：



实现要求：

- (1) 链队列基本操作的实现：链队列的初始化，生成一个空链队列；链队列的撤消，即删除队列中的所有结点，仅留下指针结点；
- (2) 链队列的入队操作，即在已知队列的队尾插入一个元素e，即修改队尾指针；
- (3) 链队列的出队操作，即返回队首结点的元素值并删除队首结点；
- (4) 设计一个菜单，具有上述要求的所有功能、退出系统等最基本的功能。

题目58：交通处罚单管理

需求分析：

交通处罚单信息用文件存储，因而要提供文件的输入输出操作；要求可以输入、删除、浏览交通处罚单信息，因而要提供信息的输入、删除和显示操作；要实现按车辆、驾驶员和开单交警查询，则需提供查找操作；另外提供统计操作和选择菜单以实现功能选择。

总体设计：

整个管理系统被设计为信息输入模块、信息删除模块、信息浏览模块、信息查询模块和信息统计模块。

详细设计：

数据结构采用结构体，设计交通处罚单结构体：

```

    Struct TranficPunishBill
    {
    Char carNo[10]; //车牌
    号Char driverNo[20];
    //驾照
    号
    
```

```

Char policeNo[10];          //
开单交警号码Char billNo[20]; //
处罚单号码

Char time[12]; //处罚时间 (yyyymmddhhmm, 年月日时分)
}TPB;

```

(1) 主函数提示用户选择功能：输入 删除 浏览 查询 统计

(2) 信息输入模块

用文件保存信息，把处罚单信息写入处罚单信息文件。

(3) 信息删除模块

输入处罚单号码，然后在处罚单信息文件中查找该条处罚单，删除之。具体删除操作：打开该文件，从文件中读一条记录，判断是否要删除，如果是，则舍弃；否则重新写入文件，读下一条记录。

(4) 信息浏览模块

读处罚单信息文件，分屏输出，每屏 10 条记录。

(5) 信息查询模块

通过菜单选择查询字段：车辆、驾驶员、开单交警，分别按照车牌号、驾照号、开单交警号采用基本的查找算法查找交通处罚单信息文件，如果找到相应得记录则输出处罚单信息，否则输出“您所查找的信息不存在！”。

(6) 信息统计模块

提示输入驾驶员的驾照号和时间段。设置一个计数器，初始化为 0。采用基本算法查找交通处罚单信息文件，比较驾照号，如果驾照号匹配，再看处罚单时间是否出于查询时间段内，如果是，则计数器加 1，并输出该处罚单号。文件查找结束后，输出计数器的值。

附录：

Visual C++编程简介

1. Visual C++创建应用程序框架

在可视化开发环境下，生成一个应用程序要做的工作主要包括编写源代码、添加资源和设置编译方式。向导实质上是一个源代码生成器，利用应用程序向导可以快速创建各种风格的应用程序框架，自动生成程序通用的源代码，这样大大减轻了手工编写代码的工作量。

创建一个应用程序时，首选要创建一个项目。项目用于管理组成应用程序的所有元素，并由它生成应用程序。Visual C++集成开发环境包含了创建各种类型应用程序的向导，执行 File 菜单中的 New 命令即可看到应用程序向导类型的列表如下：

- ◆ ATL COM AppWizard:创建包含ActiveX 控件的ATL(active template library)
- ◆ Cluster Resource Type Wizard:创建能够在微软群服务器上模拟和管理的项目
- ◆ Custom AppWizard:以用户定制的模板向导创建项目
- ◆ Database Project:创建数据库项目
- ◆ DevStudio Add_in Wizard:创建用C++或汇编语言编写的类似于Developer Studio 形式的外接程序
- ◆ Extended Stored Proc Wizard:在 SQL Server 上创建一个建立扩展存储的程序
- ◆ ISAPI Extension Wizard:利用 ISAPI(Internet server API)创建网页浏览程序
- ◆ Makefile:制作 makefile 文件，定制自己项目的集成开发环境
- ◆ MFC ActiveX Control Wizard:创建基于MFC 的ActiveX 控制
- ◆ MFC AppWizard[dll]:创建基于MFC 的动态链接库
- ◆ MFC AppWizard[exe]:创建基于 MFC 的应用程序
- ◆ New Database Wizard:在 SQL 服务器上创建一个SQL Server 数据库
- ◆ Utility Project:创建自定义编译规则的项目
- ◆ Win32 Application:创建 win32 应用程序
- ◆ Win32 Console Application:创建 DOS 下的 win32 控制台应用程序
- ◆ Win32 Dynamic-link Library:创建 win32 动态链接库
- ◆ Win32 Static Library: 创建 win32 静态链接库

从上面看出，Visual C++中程序向导种类很多，以下将重点介绍 MFC AppWizard[exe].

使用 MFC AppWizard 创建一个简单程序

区别于DOS 程序，即使一个最简单的windows 程序，它也必须显示一个程序运行窗口，

这就需要编写比较复杂的程序代码。而同一类型应用程序的框架窗口风格是相同的，如相同的菜单栏、工具栏、状态栏和客户区。并且基本菜单命令的功能也是一样的。所以同一类型应用程序建立框架窗口的基本代码都是一样的。为了避免重复编写这些代码，一般的可视化软件开发工具都提供了创建windows 应用程序框架的向导。

MFC AppWizard[exe]是一个创建基于 MFC 微软基础类的windows 应用程序的向导，它是 visual c++ 集成开发环境最常用的向导工具。当利用 MFC AppWizard[exe]创建一个项目时，它能够自动生成一个MFC 应用程序的框架。MFC 应用程序框架将那些每个应用程序都共同需要使用的代码封装起来，如完成默认的程序初始化功能、建立应用程序界面和处理基本的 windows 消息，使程序员不必浪费时间去做那些重复的工作，而把精力放在编写实质性的代码上。

MFC AppWizard[exe]向导提供了一系列对话框，在对话框中提供了一些不同的选项，程序员通过选择不同的选项，可以创建不同类型和风格的 MFC 应用程序，并可定制不同程序界面窗口。例如程序是单文档、多文档应用程序。

下面以建立一个 SDI 单文档应用程序为例说明 MFC AppWizard[exe]应用程序向导的使用方法和每个操作步骤对话框中各选项的含义。

例：编写一个单文档应用程序，程序运行后在程序视图窗口显示信息“单文档程序！”

- 1) 在 Visual C++ 中选择 File | New 菜单项，出现 New 对话框，如图 1 所示：

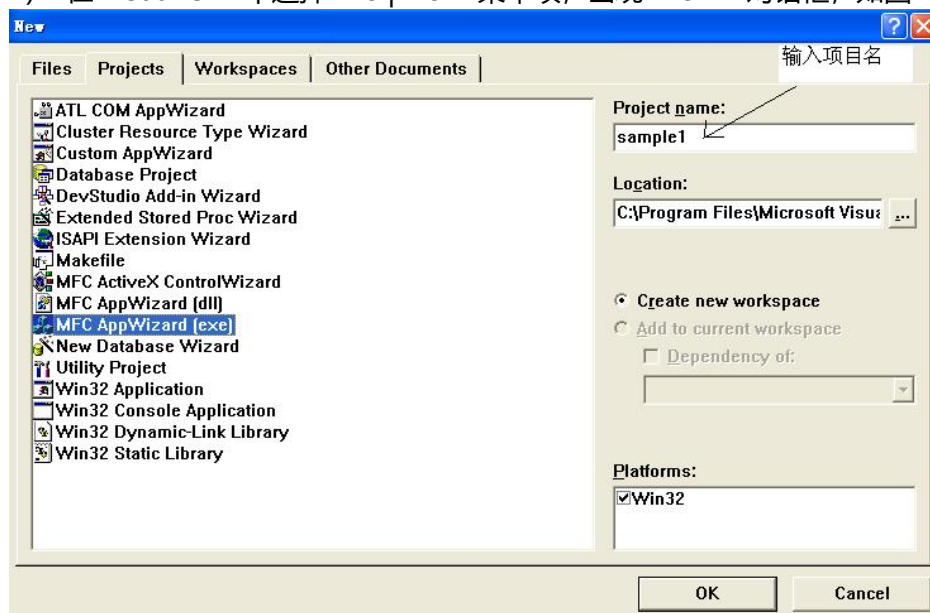


图 1 New 对话框

- 2) 确认 New 对话框的当前页面为 Project，在左档的项目类型列表框中选择 MFC AppWizard[exe]项，在 Project Name 框输入要创建项目的名称。这里输入的名称是 sample1,在 Location 档中输入项目所在路径。单称 OK 按钮，出现 MFC AppWizard - Step 1 对话框，如图 2 所示：

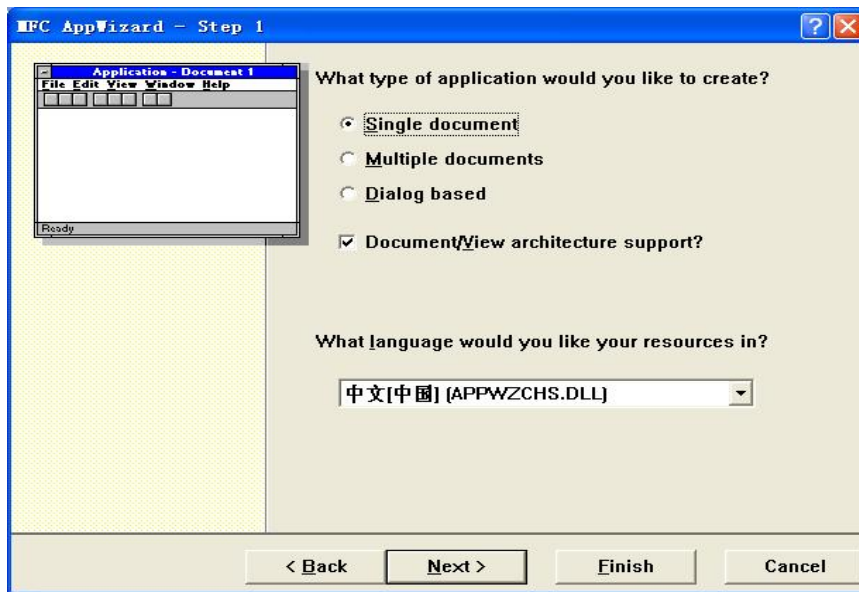


图 2 MFC AppWizard - Step 1

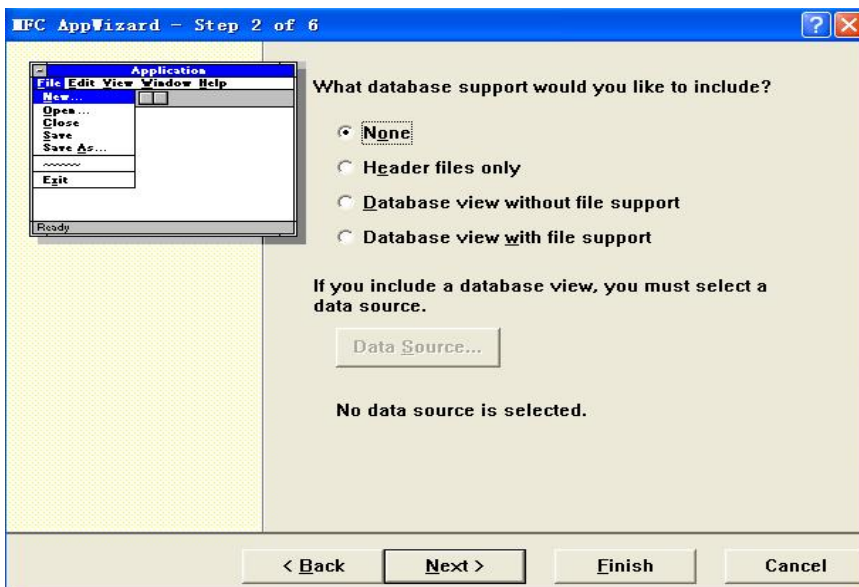


图 3 MFC AppWizard - Step 2

3) 图 2 中的 MFC AppWizard - Step 1 对话框中主要选择应用程序的类型，向导提供以下 3 种类型的应用程序：

- ◆ Single document: 单文档界面应用程序，程序运行后出现标准的 windows 界面，它由框架和客户区组成。并且程序运行后一次只能打开一个文档。
- ◆ Multiple document: 多文档界面应用程序，程序运行后出现标准的 windows 界面，并且可以打开多个文档。
- ◆ Dialog based: 基于对话框的应用程序，程序运行后首先出现一个对话框界面。先选择 Single document 选项后，再单击 Next 按钮，出现 MFC AppWizard -

Step 2 对话框，如图 3 所示。

4) 在 MFC AppWizard - Step 2 对话框中选择应用程序所支持的数据库方式，其中包括对下选项：

- ◆ None：向导创建的应用程序不包括任何对数据库的操作功能，但以后可以手工添加对数据库的操作代码。
- ◆ Header files only:提供了最简单的数据库支持，仅在项目的 stdafx.h 文件中使用#include 指令包含 afxdb.h 和 afxao.h 两个用于定义数据库类的头文件，但并不创建与数据库相关的类，用户需要时可以自己创建。
- ◆ Database view without file support :包含了所有的数据库头文件，并创建了相关的数据库和视图类，但不支持文档的序列化。
- ◆ Database view with file support: 包含了所有的数据库头文件

单击 Next 按钮，出现MFC AppWizard - Step 3 对话框如图 4：

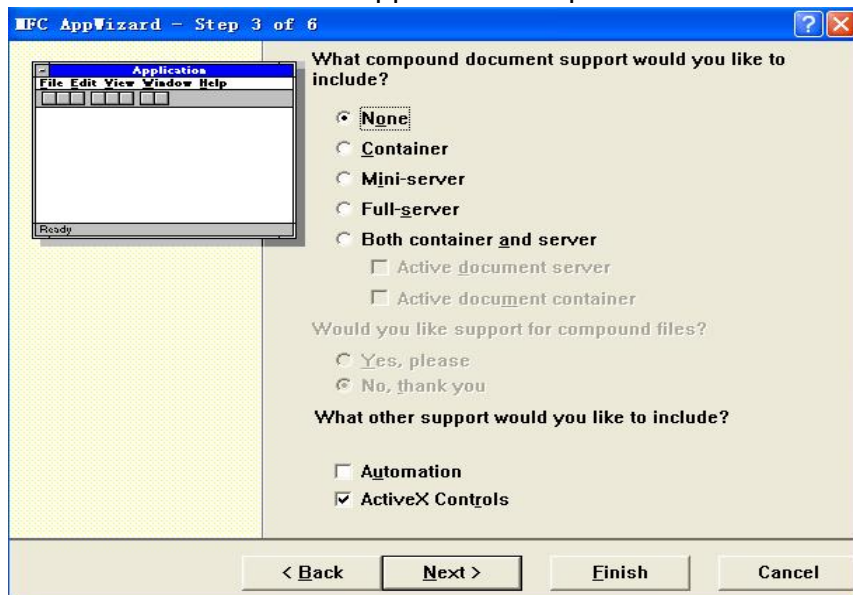


图 4 MFC AppWizard - Step 3

5) 在 MFC AppWizard - Step 3 对话框中选择应用程序支持的复合文档类型。

None:应用程序不支持任何复合文档，该项是默认选项。

Container:应用程序作为复合文档容器，能容纳所嵌入或链接的复合文档对象

Mini-server:微型复合文档服务器，应用程序可以创建和管理复合文档对象，但对于它所创建的复合文档对象，集成应用程序可以嵌入，但不能链接。

Full-server:完全复合文档服务器。

Both container and server: 应用程序既可作为一个复合文档容器，又可作为一个可单独运行的复合文档容器。

Automation:应用程序支持自动化，应用程序可以操作其他程序所创建的对象，或提供自动化对象给自动化客户访问。

ActiveX Controls:应用程序可使用ActiveX 控件。

本例中选择 None 和 ActiveX Controls 选项扣, 单击 Next 按钮, 出现图 5:

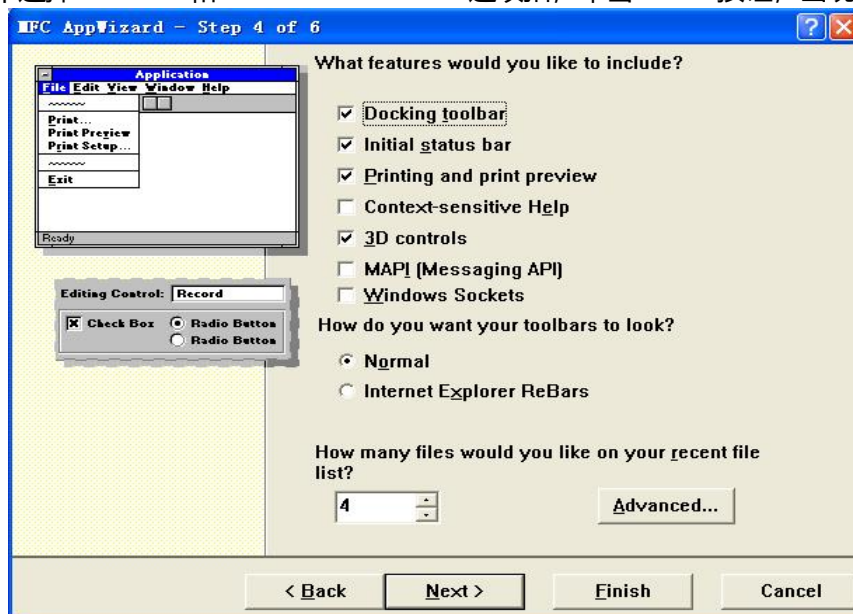


图 5 MFC AppWizard - Step 4

- 6) 在 MFC AppWizard - Step 4 对话框中设置应用程序的界面特征, 如工具栏和状态栏的设置, 其中包括以下选项:
- ◆ Docking toolbar:默认选项, 为应用程序添加一个标准的工具栏
 - ◆ Initial status bar: 默认选项, 为应用程序添加一个标准的状态栏
 - ◆ Printing and print preview: 默认选项, 应用程序支持打印和打印预览的功能
 - ◆ Context-sensitive Help:应用程序具有上下文相关帮助功能
 - ◆ 3D controls: 默认选项, 应用程序界面具有三维外观
 - ◆ MAPI (Message API) :应用程序能使用邮件API, 具有发送电子邮件的功能
 - ◆ Windows Sockets:应用程序能使用WinSock 套接字, 支持TCP/IP 协议
- 使用所有的默认选项, 单击 Next 按钮, 出现 MFC AppWizard - Step 5 对话框如图 6:

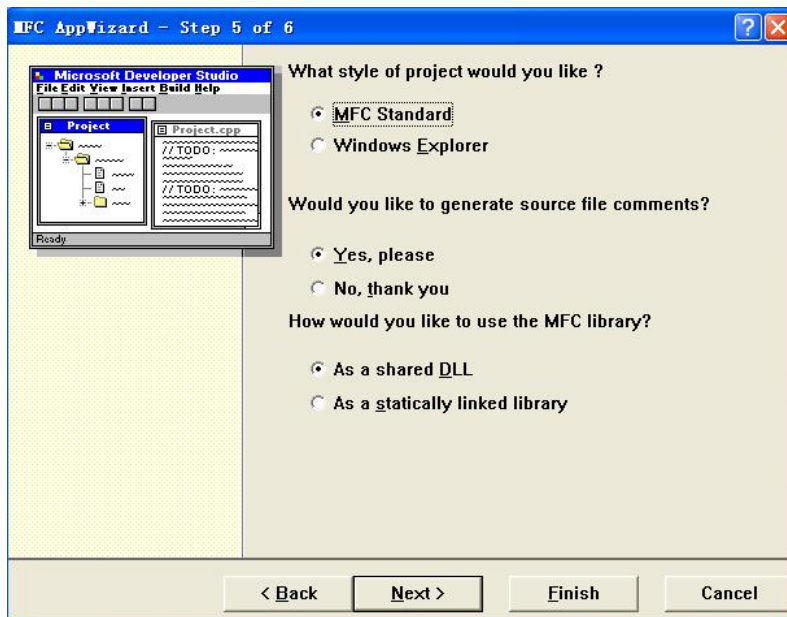


图 6 MFC AppWizard - Step 5

7) 在 MFC AppWizard - Step 5 对话框中设置项目的风格，其中包括以下选项：

- ◆ MFC Standard:应用程序采用MFC 标准风格
- ◆ Windows Explorer: 应用程序采用windows 资源管理器风格

对话框中还设置MFC 库与应用程序的链接方式：

As a shared DLL: 采用共享动态链接库的方式

As a statically linked library :采用静态链接库的方式

单击 Next 按钮，出现 MFC AppWizard - Step 6 对话框如图 7：

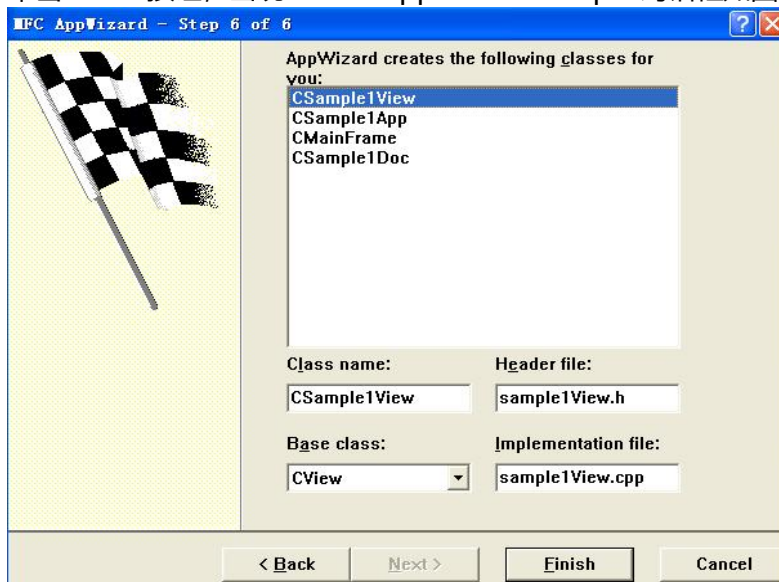


图 7 MFC AppWizard - Step 6

8) 在 MFC AppWizard - Step 6 对话框中列出了向导将创建的类，用户可以修改一

些默认的类型名和对应的头文件名。单击Finish 按钮，出现对话框如图 8：

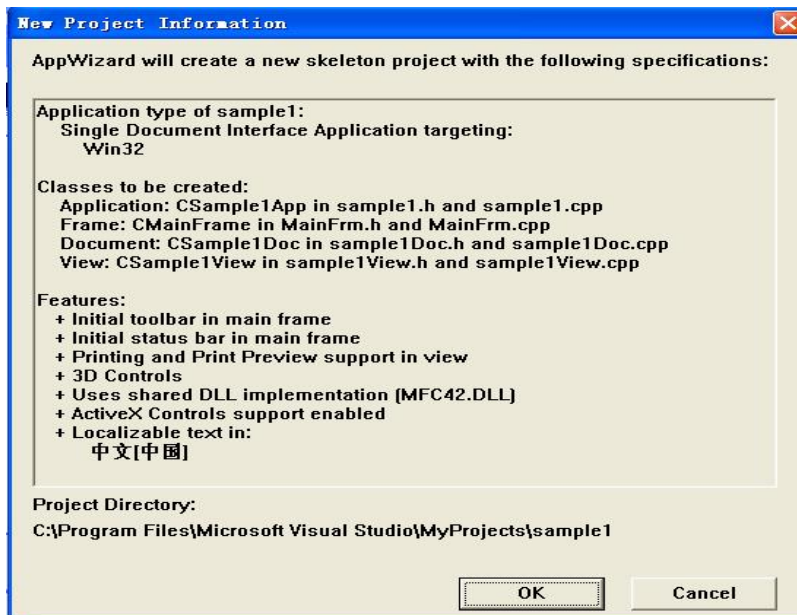


图 8 生成项目的信息

在生成对话框中，单击 OK 按钮，MFC AppWizard 将开始创建应用程序框架。当应用程序框架创建成功后，Developer Studio 将装入应用程序项目，并在项目工作区窗口打开这个项目。利用MFC AppWizard 向导创建应用程序sample1 的框架后，用户无需添加任何代码就可以对程序进行编译、链接，生成一个应用程序。在本例中，需要在成员函数CSample1View::OnDraw(CDC* pDC)中添加显示文本“单文档程序!”的代码。在Workspace 窗口单击ClassView 标签，单击CSample1View 类左边的“+”展开该类，双击其中的成员函数 OnDraw()，在编辑窗口出现该成员函数的代码，在指定位置添加如下粗体所示的代码：

```
void CSample1View::OnDraw(CDC* pDC)
{
    CSample1Doc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    pDC->TextOut(200,200,"单文档程序! ");//在坐标(200,200)处显示文本
}
```

函数 TextOut () 是 MFC 类CDC 的成员函数，其功能是在指定位置输出字符串，第 1,2 个参数是坐标位置，第 3 个参数是要输出的字符串。执行 Build 命令编译链接程序，程序运行后将在程序视图区域显示文本串。如图 9 所示：

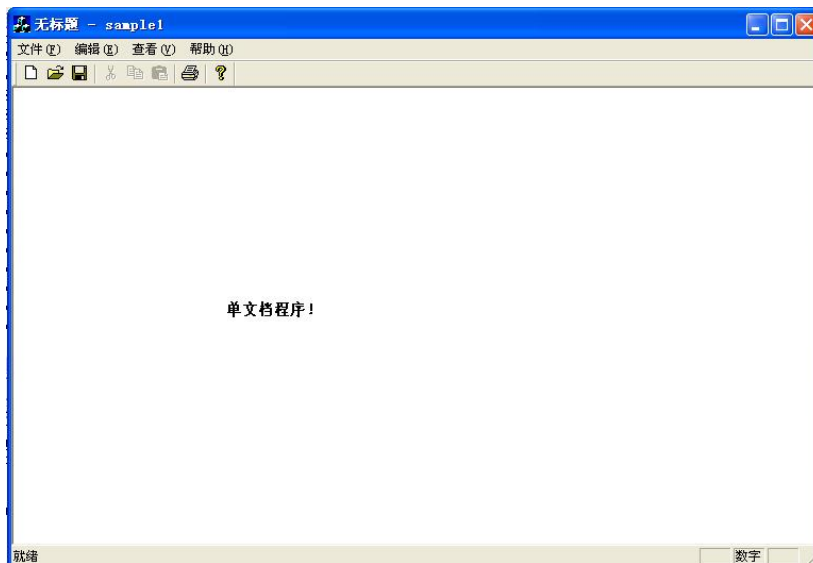


图 9 运行程序 sample1

2. 文档与视图

Windows 应用程序的流行，得益于它的用户界面相容的特点。在 MFC 应用程序中，文档与视图用来将程序的数据本身与数据的显示以及用户对数据的交互相隔离。在这种模式中，文档负责管理和维护数据本身，这些包括从永久介质（如磁盘）文件中取出数据或是将已修改的数据保存到磁盘文件中。而视类则负责从文档类中的数据取来显示给用户，并接受用户的修改，然后将修改的结果反馈给文档类，由文档类将修改保存到磁盘文件中。视类和文档类的交互是通过文档类中的公共成员变量或成员函数实现的。

在通常情况下，视通过 GetDocument 成员函数获得指向相关联的文档对象的指针，并通过该指针调用文档类的成员函数来从文档中读取数据。视把数据显示于计算机屏幕上，用户通过与视的交互来查看数据并对数据进行修改。然后，视通过相关联的文档类的成员函数将经过修改的数据传递给文档对象。

以下简单介绍一个在单文档程序中的绘图例子。

画笔 画笔 (CPen) 是 windows 的应用程序用来绘制直线和曲线的图形工具。绘制应用程序用它来画直线和曲线。画笔分两类：修饰类和几何类。修饰画笔适用的应用程序需要绘制固定宽度的线和可以速刷的线。几何画笔适用的应用程序则要求线可以缩放，具有特定的端点和接合方式，线宽大于一个像素。

◆ 修饰画笔

修饰画笔的尺寸以设备单位表示，因此修饰画笔所绘制的线条通常有固定的宽度。修饰画笔的画线速度通常是几何画笔画线速度的三到十倍。修饰画笔有三种属性：宽度、风格和颜色。

应用程序调用 CreatePen,CreatePenIndirect 或 ExtCreatePen 函数来建立一种修饰画笔。应用程序用 GetStockObject 函数可以由窗口管理程序所管理的三种库存修饰画笔中检取一种。

建立画笔后，调用SelectObject 函数，把该画笔选进应用程序的设备描述表中。以后应用程序在其客户区的所有画笔操作都由这一画笔完成。

◆ 几何画笔

几何画笔的尺寸是以逻辑单位表示的，因此所画的线是可以缩放的。也就是说，它的宽度可宽可窄。应用程序用 ExtCreatePen 函数建立几何画笔，和修饰画笔一样，用 SelectObject 函数把几何画笔选进应用程序的设备描述表中。

```
HPEN ExtCreatePen(DWORD dwPenStyle, //pen style
DWORD dwWidth, //pen width
CONST LOGBRUSH *lpb, //pointer to structure for brush attributes
DWORD dwStyleCount, //length of array containing custom style bits
CONST DWORD *lpStyle //optional array of custom style bits
);
```

参数说明：

dwPenStyle 指定画笔风格和属性。函数提供了许多风格，如果需要同时具有两种以上的风格，用操作符“|”连接即可，如表 1 所示

表 1 画笔风格描述表

类 型	描述
PS_GEOMETRIC	是几何画笔
PS_COSMETRIC	是修饰画笔
PS_ALTERNATE	画笔设置一个像素，仅适用于修饰画笔
PS_SOLID	实心画笔
PS_DASH	虚 线 画 笔
PS_DOT	点 线 画 笔
PS_DASHDOT	虚点线画笔
PS_DASHDOTDOT	交替虚线双点线笔
PS_NULL	空的画笔
PS_INSIDEFRAME	画笔是实心的
PS_USERSTYLE	用户自定义画笔

dwWidth 指定画笔的宽度，如果 dwPenStyle 参数是 PS_GEOMETRIC，则给出的宽度是逻辑宽度，如果是PS_COSMETRIC，宽度必须是 1。

Lpb: 指向 LOGBRUSH 结构的指针，如果 dwPenStyle 参数是 PS_GEOMETRIC，则该结构中的 lbcColor 指出笔的颜色，而 lpStyle 必须是 BS_SOLID；如果 dwPenStyle 参数是

PS_COSMETRIC,则该结构中所有的数据成员都用来指定画笔的刷子属性。
dwStyleCount 指定 lpStyle 数组的长度, 如果 dwPenStyle 不是 PS_USERSTYLE, 那么该值必须为 0
lpStyle 和用户自定义风格相关的数组, 如果 dwStyleCount 不是 PS_USERSTYLE, 那么该值必须为 NULL

类 CPen CPen 对象封装了 windows GDI 笔, 并提供与 CPen 对象一起工作的几种方法。当在设备描述表内绘图时, 创建笔供使用。笔以多种样式和颜色出现, 一般用 CPen::CreatePen()方法创建 windows 笔, 并将它连接到 CPen 对象, 该方法使用以下原型:
BOOL CreatePen (int nPenStyle,int nWidth,COLORREF crColor);
类成员函数如表 2 所示

表 2 CPen 类成员函数说明

类 型	描述
CreatePen	创建具有指定样式、宽度、刷子属性的逻辑笔, 并将它连至 CPen 对象
CreatePenIndirect	创建具有LOGPEN 结构内定义的样式、颜色和宽度的笔、并将它连至CPen 对象
Fromhandle	从 windows HPEN 返回指向CPen 对象的指针
GetExtLogPen	获取笔的基础EXTLOGPEN 结构
GetLogPen	获取笔的基础LOGPEN 结构
Operator HPEN	返回当前连接在CPen 对象上的windows 笔的句柄

CPen 类有三个重载构造函数, 例举如下:
CPen();
CPen (int nPenStyle,int nWidth,COLORREF crColor)
 Throw (CresourceException);
CPen(int nPenStyle,int nWidth,const LOGBRUSH *pLogBrush,int nStyleCount=0,
 Const DWORD* lpStyle=NULL)
 Throw (CresourceException);
如果调用第一个构造函数 (空函数), 它还必须调用 CreatePen,CreatePenIndirect 或 CreateStockObject 成员函数初始化CPen 对象; 如果调用其他两种构造函数, 则无需进一步初始化, 构造函数所使用的参数在表 3 中所列。

表 3 构造函数参数说明

类 型	描 述
nPenStyle	指定画笔样式
nWidth	指定笔划的。如果设置为 0，则宽度默认值为 1
nColor	画笔使用的RGB 颜色值
nLogBrush	LOGBRUSH 结构
NStyleCount	LpStyle 数组所包含的DWORD
LPStyle	DWORD 值的数组。首值指定用户自定义线型的第一短线的长度；第二个值指定第一间空的长度

例：用红线虚线绘制菱形。如图 10 所示：

步骤：1.创建一个工程 sample2,用SDI，其余都用默认设置，然后确定就创建了一个单文档的视图窗口。

2.在 CSample2View 类中的OnDraw 函数里加入下面的代码

```
void CSample2View::OnDraw(CDC* pDC)
{
    CSample2Doc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
    CPen penRed;
    penRed.CreatePen(PS_DASH,1,RGB(255,0,0));
    CPen * ppenOld;
    ppenOld=pDC->SelectObject(&penRed);
    CRect rc;
    GetClientRect(&rc);
    pDC->MoveTo(0,(rc.bottom+rc.top)/2);
    pDC->LineTo((rc.right+rc.left)/2,0);
    pDC->LineTo(rc.right,(rc.bottom+rc.top)/2);
    pDC->LineTo((rc.right+rc.left)/2,rc.bottom);
    pDC->LineTo(0,(rc.bottom+rc.top)/2);
    pDC->SelectObject(ppenOld);
}
```

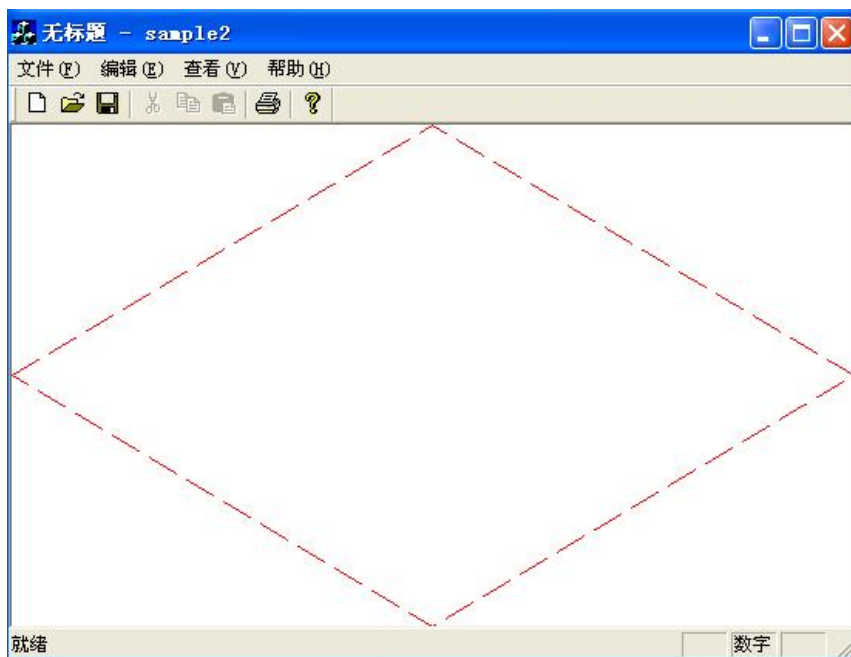


图 10 绘制菱形

3. 对话框

对话框是windows 应用程序中一种常用的资源，其主要功能是输出信息和接收用户的输入数据。在 MFC 中，对话框的功能被封装在 CDialog 类中，而 CDialog 类是 CWnd 类的派生类。作为窗口，对话框具有窗口的一切功能。可以直接利用 MFC AppWizard 应用程序向导创建一个基于对话框的应用程序，这种程序运行后首先出现一个对话框。在 MFC AppWizard 向导的第 1 步选择 Dialog Based 选项，此时，应用程序向导将出现与单文档和多文档程序向导不同的操作步骤，因为对话框应用程序一般不包含文档，故不支持数据库和复合文档。

按照对话框应用程序向导提示的步骤进行操作就得到一个对话框应用程序项目，出现如图 11 所示的对话框编辑器和控件工具栏浮动窗口，它们采用的是一种所见即所得的可视化工作方式。利用向导创建应用程序的框架后，程序员可根据程序具体功能要求添加代码。

执行 File | New 命令，出现 New 对话框，选择 MFC AppWizard [exe] 项，输入程序名 sample3，单击 OK 按钮。在随后出现的 MFC AppWizard - Step 1 对话框窗口中选择 Dialog Based 选项，单击 Finish 按钮就创建了应用程序项目。出现图 11

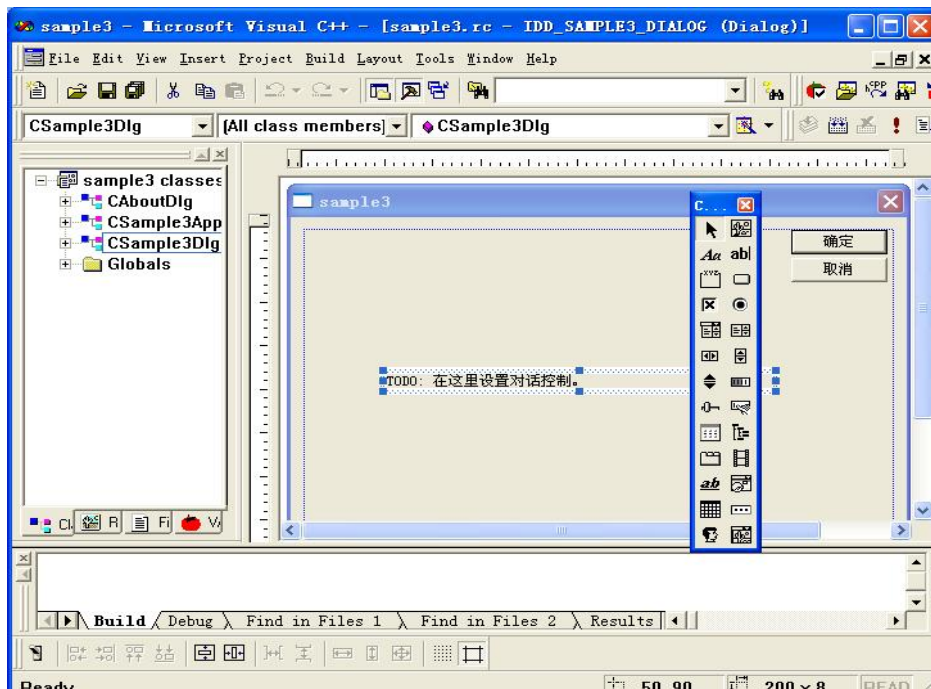


图 11 对话框应用程序的开发窗口

对话框类 为了程序员方便实现对话框功能，MFC 提供了一系列对话框，并实现了对对话框消息响应和处理机制。CDialog 类是对话框中最重要的类，用户在程序中创建的对话框类一般都是 CDialog 类的派生类。CDialog 类还是其它所有对话框类的基类。

CDialog 类从 CWnd 类派生而来，所以它继承了 CWnd 类的成员函数，具有 CWnd 类的基本功能，可以编写代码移动、显示或隐藏对话框，并能根据对话框的特点增加新的成员函数，扩展它的功能。表 4 中所列是 CDialog 类中经常要使用的成员函数，在用户的 CDialog 类的派生类中可以直接调用。大部份的成员函数是虚函数，可以在用户的派生类中重载，以实现特定的目的。除了 CDialog 类成员函数，类 CWnd 和 CWinApp 也提供了一些成员函数用于对话框的管理。

表 4 有关对话框的常用函数

成员函数	功 能
CDialog::CDialog()	通过调用派生类构造函数，根据对话框资源模板定义一个对话框
CDialog::DoModal()	激活模态对话框，显示对话框
CDialog::Create()	根据对话框资源模板创建非模板对话框窗口
CDialog::OnOk()	单击 OK 按钮时调用该函数，接收对话框输入数据，关闭对话框
CDialog::OnCancel()	单击 OK 按钮时调用该函数，不接收对话框输入数据，关闭对话框
CDialog::OnInitDialog()	WM_INITDIALOG 的消息处理函数
CDialog::EndDialog()	用于关闭模态对话框窗口
CWnd::ShowWindow()	显示或隐藏对话框窗口
CWnd::DestroyWindow()	关闭并销毁非模态对话框窗口
CWnd::UpdateData()	通过调用DoDataExchange()设置或获取对话框控件的数据
CWnd::DoDataExchange()	被 UpdateData()调用以实现对话框数据交换，不能直接调用
CWnd::GetWindowText()	获取对话框窗口的标题
CWnd::SetWindowText()	设置对话框窗口的标题
CWnd::GetDlgItemText()	获取对话框中控件的文本内容
CWnd::SetDlgItemText()	设置对话框中控件的文本内容
CWnd::GetDlgItem()	获取控件或子窗口的指针
CWnd::MoveWindow()	用于移动对话框窗口
CWnd::EnableWindow()	使窗口处于禁用或可用状态

添加控件 如果没有控件，对话框完成不了具体功能，对话框与控件有着密不可分的关系。结合对话框编辑器，利用 controls 控件工具栏可给当前正在编辑的对话框模板资源添加控件。控件工具栏上的每个图标代表了一种控件，如果不能确定控件的类型，只要将光标在该控件图标上停留片刻，就会显示说明该控件类型的提示文本。

向对话框添加控件时，先在 controls 控件工具栏单击要添加的控件，再将光标指向对话框空白区域并单击鼠标，就可以将选择的控件加入对话框模板资源。在生成自己的对话框类并添加需要的控件后，可以利用 ClassWizard 类向导在对话框类的定义中为对话框资源上的每一个控件添加一个或多个对应的成员变量。ClassWizard 类向导的 Member Variable 页面主要用来为对话框类添加和删除与对话框控件关联的成员变量，在编写对话框程序时经常同该页面打交道。MFC ClassWizard 类向导的Member Variable 页面如图 12 所示

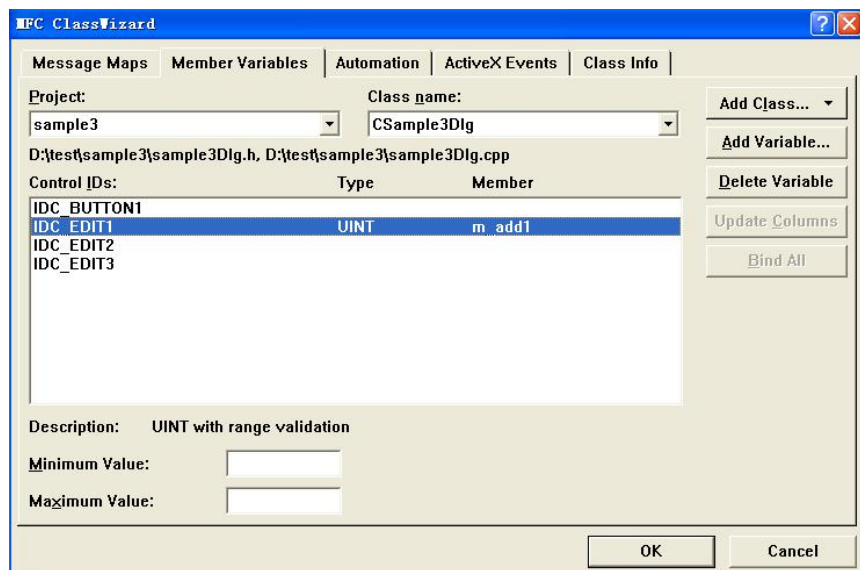


图 12 ClassWizard 类向导Member Variable

在此页面中，Class name 下拉框用于选择要添加成员变量的对话框类；Control IDs 栏用于选择控件，因为要添加的成员变量总是与一个对话框控件 ID 联系在一起，它们代表控件对象本身或控件的某项属性；单击Add Variable 按钮就可创建一个与控件关联的成员变量；单击Delete Variable 按钮就可删除指定控件的某个成员变量。Control IDs 栏列出对话框资源上已有的控件，它是一个列表框，第 1 列 Control IDs 表示控件的 ID，第 2 列 Type 表示变量的类型，第 3 列 Member 表示成员变量名。

选定对话框类和控件 ID 后，单击Add Variable 按钮,将弹出 Add Member Variable 对话框，如图 13 所示

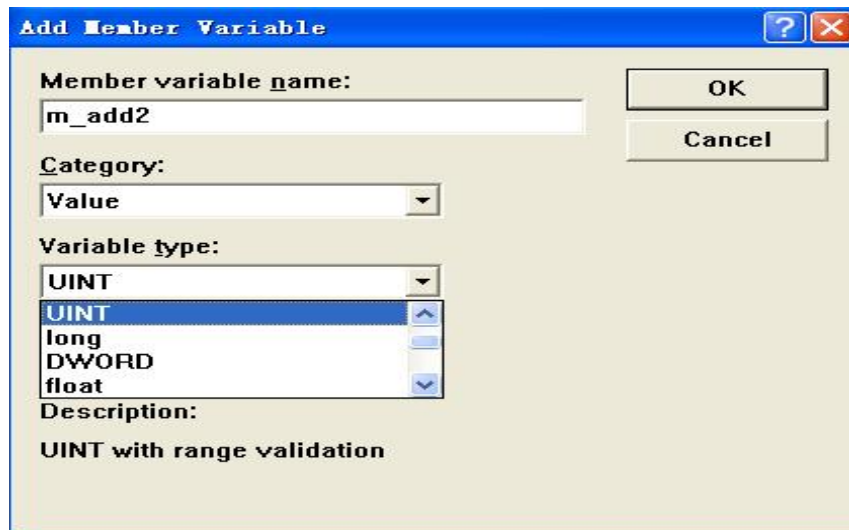


图 13 添加成员变量

如果在Category 下拉框中选择Value 项，表示要为该控件的某项属性定义一个变量，而不是控件本身。这时，还可以通过Variable type 下拉框为变量选择不同的类型，可以选择一般的C++

数据类型。如果在 Category 下拉框中选择 control 项，表示定义的变量代表控件对象本身，control 类型的变量实质是一个控件对象，其类型是 MFC 控件类。例如，对于编辑框控件，此时成员变量的类型为 Cedit。因此，可以通过添加的控件对象访问控件类的成员变量和调用控件类的成员函数，实现对控件行为的控制和管理。

例 编写一个基于对话框的加法程序。

步骤： 1. 创建基于对话框的项目 sample3,向对话框资源添加控件。在 control 工具栏中选择 StaticText 控件，添加三个静态文本控件，其标题(caption)分为“ 加数”，“ 被加数”，“ 两个数之和”。该控件作为输入框的提示文本。采用同样的方法为对话框资源添加三个 Edit Box 编辑框控件，该控件用于接收用户的输入数据。其默认 ID 分别为IDC_EDIT1, IDC_EDIT2, IDC_EDIT3。

2. 添加与控件相关联的成员变量，启动Class Wizard 类向导，单击Member Variable 标签。在class name 下拉框中选择类CSample3Dlg，在control IDs分别选择编辑框IDC_EDIT1, IDC_EDIT2, IDC_EDIT3,如图 12 所示，单击add variable 按钮，通过Add Member Variable 对话框添加对应的成员变量m_add1 , m_add2,m_result,其数据类型都为 UNIT。

3. 向对话框资源添加控件 Button，设置 captioin 为“ 相加”。效果如图 14 所示

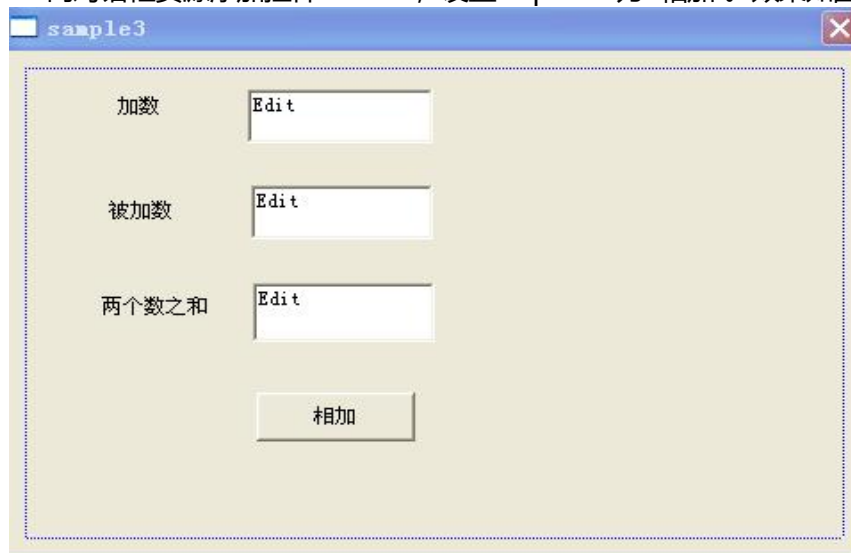


图 14 添加控件后的对话框

双击 Button 按钮，在void CSample3Dlg::OnButton1()函数中手工添加实现代码。

```
void CSample3Dlg::OnButton1()
{
    // TODO: Add your control notification handler code here
    UpdateData(true);
    m_result=m_add1+m_add2;
    UpdateData(false);
}
```

程序运行的结果如图 15 所示：

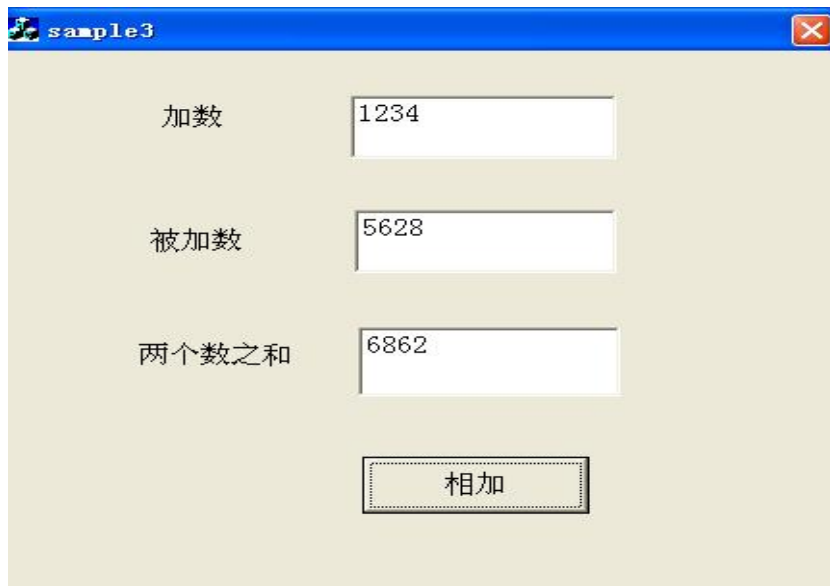


图 15 基于对话框的加法程序

当调用 `UpdateData(true)` 时，MFC 通过调用用户对话框派生类的成员函数 `DoDataExchange()` 函数将数据从控件传递到关联的成员变量。当调用 `UpdateData(false)` 时，MFC 通过调用 `DoDataExchange()` 函数将成员变量传递到相关联的控件。利用 `UpdateData()` 函数实现了在刷新后重新获取成员变量。