# 作业 第七次

1. 分数相加，两个分数分别是 1/5 和 7/20, 它们相加后得 11/20。方法是先求出两个分数分母的最小公倍数，通分后，再求两个分子的和，最后约简结果分数的分子和分母。（如果两个分数相加的结果是 4/8, 则必须将其约简成最简分数的形式 1/2)，即用分子分母的最大公约数分别除分子和分母。求最大公约数的一种方法为：将 m、n 较小的一个数赋给变量 k, 然后分别用{k, k-1，k-2, ..，1}中的数（递减）去除 m 和 n, 第一个能把 m 和 n 同时除尽的数就是 m 和 n 的最大公约数。假定 m、n 的最大公约数是 v，则它们的最小公倍数就是 m\*n/v。试建立一个分数类 Fract，完成两个分数相加的功能。具体要求如下：
   1. 私有数据成员

* Int num，den ；num 为分子，den 为分母。
  1. 公有成员函数
* Fract ( int a=0,int b=l )：构造函数，用 a 和 b 分别初始化分子 num、分母 den。
* int ged ( intm，int n )：求m、n的最大公约数。此函数供成员 add()函数调用。
* Fract add( Fract f): 将参数分数 f 与对象自身相加，返回约简后的分数对象。
* void show()：按照 num/den 的形式在屏幕上显示分数。
  1. 在主程序中定义两个分数对象 fl 和 f2, 其初值分别是 1/5 和 7/20, 通过 fl 调用成员函数 add 完成 fl 和 f2 的相加，将得到的分数赋给对象 f3，显示分数对象 f3。

1. 试定义一个类ID，将 15 位的旧版身份证号扩充为 18 位。在 15 位身份证号中，第 7、8 两位为出生年份，例如，1980 年出生的人，身份证号码的第 7、8 位的值是 80, 在 18 位身份证号中，将 7~10 四位的值改为 1980, 并将原身份证号码第 9 位开始以后的所有数字依次向右平移 2 位。在 18 位身份证号中，最后增加一位校验码，校验码的计算方法如下（只考虑 20 世纪出生的公民)：
   1. 将己扩展出的 17 位身份证号按各位上的数字进行加权求和，结果为 S。自左到右各位上的数字的权值依次为：{7，9，10，5，8，4，2，1，6，3，7，9，10，5，8，4，2}。
   2. 将 S 对 11 取余运算，结果为 Y。
   3. 根据 Y 的取值大小顺序{0，1，2，3，4，5，6，7，8，9，10}取对应的校验码{1，0，X，9，8，7，6，5，4，3，2}。

例如，当 15 位身份证号为 340524800101001 时, 扩展年份后成 17 位：34052419800101001, 各位的数字的加权和S=3\*7+4\*9+0\*10+5\*5+2\*8+4\*44-1\*2+9\*1+8\*6+0\*3+0\*7+1\*9+0\*10+1\*5+0\*8+0\*4+1\*2=189，S 对11 取余，即S%11=189%11=2, 则校验码为“X”。具体要求如下：

* 1. 私有数据成员
* char s[19],x[11]；s数组存放身份证号，x数组按顺序存放待校验码{1，0，X，9，8，7，6，5，4，3，2}。
* int w[17]；存放自左向右的身份证号前17位数字的权值。
  1. 公有成员函数
* ID(char \*str)：构造函数，用str初始化数据成员（假设这里给定的是15位身份证号），并根据题意初始化数据成员x和w。
* void fun()：按题意将存放在s中的15位身份证号扩展为18位。
* void print()：输出身份证号。
  1. 在主函数中用一个15位身份证号对类ID进行测试。

输出示例：

原身份证号：340524800101001

扩展后的身份证号：34052419800101001X

1. 定义一个字符串类 String，具体要求如下:
   1. 私有成员数据：

* char \* str1,\* str2;
* char \*str; //str 为 str1 与 str2 之和，如 str1 为 “db a”， str2 为“4 1”，则 str 为“db a4 1”；（“ ”，为“空格”字符)
  1. 公有成员函数：
* 构造函数：根据给定的两个字符串初始化成员数据 str1 和 str2，由 str1 和 str2 得到 str。
* void dele( )：刪除 str 中的空格字符；
* void sort( ): 对 str 中的字符按升序进行排列；
* void show( )：输出所有成员数据：
* 析构函数。
  1. 设计一个完整的程序对该类进行测试。
* 以字符串“db a”和“4 1”初始化 strl 和 str2；
* 输出删除空格和排序前的字符串 strl 、str2 和 str;
* 输出刪除空格和排序后的字符串 strl 、str2 和 str。

1. 试定义 RECT 类（长方形）及其派生类 CUB ( 长方体），具体要求如下：
   1. 类 RECT 的成员如下：
      1. 保护数据成员
      * double x,y; 分别表示长方形的长和宽。
      1. 公有成员函数
      * RECT ( double x1,double y1 ); 构造函数，分别用 x1 、y1初始化 x 和 y。
      * virtual double area( ;) 虚函数，计算长方形的面积，计算公式：面积=长\*宽。
      * double peri( ;) 计算长方形的周长。计算公式：周长=2\*长+2\*宽。
      * virtual int isSquare( ); 虚函数，判断是否为正方形，如是，返回 1; 否则返回 0。
   2. 类 CUB 为类 RECT 的公有派生类，其成员如下：
      1. 私有数据成员
      * double heigh;t 表示长方体的高度。
      1. 公有成员函数
      * CUB( ); 构造函数，用 h、x、y 分别初始化 height 及其基类成员 x 和 y。
      * double volume( ;) 计算长方体的体积。计算公式：体积=底面积\* 高，其中底面积通过调用基类成员函数 area( )计算。
      * double area(); 计算长方体的表面积。计算公式：表面积=2 \* 底面积+底面周长\*高度。底面积和底面周长分别调用基类成员函数 areaO 和 peri 0计算。
      * int isSquareO;判断是否为正方体，如是，返回 1，否则返回 0。在判断过程中，首先调用基类的成员函数 isSquareO 判断底面是否为正方形。
   3. 在主函数中，创建一个 CUB 对象 cu 和一个 RECT 指针\*re，并使 re 指向 cu。通过 cu 调用 volume(),计算并输出长方体的体积：通过 re 调用 area()和 isSquare()，计算并输出长方体的表面积，并判断是否为正方体。
2. 设计评选优秀教师和优秀学生候选人的程序。如果学生的分数大于 90, 则可评为优秀生；如果教师发表的论文数大于 3, 则可评为优秀教师。具体要求如下：
   1. 定义基类Base:
      1. 保护数据成员
      * char name[8]：//存放姓名
      * int num：//存放分数或论文数
      1. 公有成员函数
      * Base( )：构造函数，输入姓名；
      * void print( )：功能函数，输出数据成员：
      * virtual int Isgood( )=0：纯虚函数，用于判断是否满足条件；
   2. 由基类派生学生类Student, 定义公有成员函数：
      * Student( )：构造函数，输入分数；
      * int Isgood()：根据优秀学生的标准，满足条件返回1，否则返回0;
   3. 由基类派生教师类Teacher，定义公有成员函数：
      * Teacher( )： 构造函数，输入论文数；
      * int Isgood( )：根据优秀教师的标准，满足条件返回1，否则返回0;
   4. 在主函数中定义学生数组和教师数组，输入一系列教师或学生的记录后，将优秀教师及学生候选人的情况列出来。
3. 把描述直角坐标系上的一个点的类作为基类，派生出描述一条直线的类和描述一个三角形的类。定义成员函数求出两点间的距离和三角形的面积。

提示：先定义描述点的类 Point; 类 Point 的派生类Line 为直线类，直线有两个端点，所以它在点类的基础上新增一组点的坐标(x2,y2); 三角形类 T 在直线的基础上再新增一组点的坐标(x3,y3), 求出三角形的面积。具体要求如下：

* 1. 定义点类 Point
     + int x1,y1：//保护的数据成员（点坐标）
     + 公有构造函数Point ( int a,int b ): //初始化 x1、y1
  2. 定义直线类 Line
     + int x2,y2; //保护的数据成员（点坐标）。
     + 公有构造函数 Line (int a,int b，int c,int d )：//初始化 x2、y2，以及 x1、y1
  3. 定义三角形类Triangle
     + int x3,y3； //私有的数据成员（点坐标）
     + double area； //私有的数据成员（面积）
     + 公有构造函数 Triangle ( int a,int b,int c,int d, int e,int f)s //初始化 x3、y3，以及x1、y1，x2、y2
     + void f():面积 ：//求三角形面积的功能函数，先求出三条边x、y、z，然后利用公式 s=(x+y+z)/2，area=sqrt(s\*(s-x)(s-y)(s-z))
     + void print(): //输出三个点的坐标和面积
  4. 在主函数中对该类进行测试。定义一个 Triangle 类的对象 tri, 以 1和1，4和1,以及 4和5 作为点的坐标，完成测试工作。程序运行输出：

(1,1) (4,1) (4,5)

area=6

1. 将字符串中指定的单词（字母子串）进行单数变复数的处理。单词之间用非字母分隔。为了简化编程，将单词的单数变复数的规则是：直接在单词后加 s。例如，单词 student 的复数为 students。若有下列字符串：We are student,you are student,too.指定单词 student，将字符串中所有的 student 修改为 students。处理后的字符串为：We are students, you are students,too. 试建立一个类 Modistr，完成以上工作。具体要求如下：
   1. 私有数据成员：
      * char \*str; 存放指定的字符串。
   2. 公有成员函数：
      * Modistr(char \*s); 构造函数，动态申请字符串空间，用 s 指向的串初始化 str。
      * void print();输出字符串 str。
      * void modify(char \*wordp)将 wordp 所指向的单词转换为复数。
      * ~Modistr()析构函数。
   3. 在主函数中完成对该类的测试。定义类 Modistr 的对象 mys,用字符串”We are student,you are student,too.”初始化 mys，调用成员函数完成操作。
2. 用c++实现一个单链表结构。可定义单链表节点结构的简单形式如下：

struct Node  
{  
 int Data;  
 Node\*next;  
};

其中，\*next是指针域，用来指向该节点的下一个节点；Data是一个整形变量，用来存放结点中的数据。

在此基础上，定义一个链表类list，其中包含链表结点的插入，删除，输出等功能的成员函数。（成员函数的功能见后续表格中说明）

class List

{

Node\*head;

public:

List() {head=NULL;} //构造函数，初始化

void CreateList(); //创建链表

void InsertList(int x, int i);//链表结点的插入

void DeleteList(int i);//链表结点的删除

void SearchList(int x);//链表结点的查找

void OutputList();//链表的遍历输出

void InverseList();//链表的逆置

void MergeList( List \* Lb);//链表的合并，功能是将两个升序链表合并为一个升序链表

Node\*Gethead(){return head;}  
};

|  |  |
| --- | --- |
| 描述： | 在单链表存储结构上实现基本操作：初始化、创建、插入、删除、查找、遍历、逆置、合并运算。 |
| 输入： | 输入线性表La的长度:n 输入线性表La中的元素:a1 a2 a3 ...an（数值有序，为降序） 输入要插入到线性表La中的元素x和插入的位置i:x i 输入要删除元素的位置:i 输入要查找的元素:x 输入线性表Lb的长度:m 输入线性表Lb中的元素:b1 b2...bm(数值有序，为升序) |
| 输出： | 创建好的线性表La=a1 a2...an 插入一个元素后的线性表La=a1 a2...an+1 删除一个元素后的线性表La=a1 a2...an 查找一个输入的元素，如果找到，输出"找到，x在第i个位置"；如果没有找到，输出"没找到"的信息。 逆置La后的线性表an an-1...a1 合并La和Lb后的线性表 |

|  |  |
| --- | --- |
| 输入样例： | 6 15 13 10 9 8 5 7 6 4 10 4 1 3 6 9 |
| 输出样例： | 创建好的线性表La=15 13 10 9 8 5 插入一个元素后的线性表La=15 13 10 9 8 7 5 删除一个元素后的线性表La=15 13 10 8 7 5 找到，10在第3个位置 逆置后的线性表La=5 7 8 10 13 15 合并La和Lb后的线性表=1 3 5 6 7 8 9 10 13 15 |