

C++上机题

张威

南京林业大学信息学院

1 Digital	1
2 Matrix.....	1
3 Cube.....	1
4 名空间util	2
5 POS	2
6 Caesar.....	2
7 Hanoi	3
8 Guess	3
9 Stuff.....	4
10 Poker	5
11 Student.....	6
12 POS + Stuff.....	6

1 Digital

从键盘上输入一个介于[10~99]的整数，按下图所示的数码形式在控制台上用星号（*）显示该整数。每个数字的数码自上而下划分为5段。具体要求如下：

```
*****
      * * *
      * * *
      * * *
*****
*       * *
*       * *
*       * *
*****
```

1. 对输入的整数进行有效性检查；
2. 两个数字的列宽相同，可以根据需要自行调整（常量）；
3. 第2和4段各占3行（例如，数字2的两个竖直段）；
4. 两个数字之间的间隔为1列；
5. 各段的绘制不得硬编码，要求采用循环的形式；函数原型依次为：segment_one(int digit)、segment_two(int digit)、segment_three(int digit)、segment_four(int digit)、segment_five(int digit)，其中，参数digit为数字（0~9）；
6. 主程序的文件名为digital.cpp；辅助函数的名字自行命名。

2 Matrix

用new操作符在堆上分配一个表示3行5列矩阵的二维数组，将该数组的每个元素的值初始化为相应的行数和列数的乘积（例如，第1行、第1列的元素为1），并在控制台上打印输出该矩阵，元素之间用单个空格分隔，一行输出完毕后换行。具体要求：

1. 主程序的文件名为matrix.cpp；
2. 矩阵的初始化要求采用循环进行；
3. 不得存在内存泄露。

3 Cube

编写一个名为cube的函数模版，它计算给定参数的立方；在main()函数中，从键盘上输入一个整数（int）和一个双精度浮点数（double），调用cube函数分别计算其立方。cube函数和main()函数位于相同的cube.cpp文件中。

4 名空间util

编写一个名为square的模版函数，它计算给定参数的平方；将该模版函数与cube函数放在名为util的名空间中；在main()函数中，从键盘上分别输入一个整数和一个双精度浮点数对这两个函数进行测试。具体要求如下：

1. cube和square的声明和实现相分离，头文件为util.h，实现文件为util.cpp；
2. main()函数所在的文件名为driver.cpp。

5 POS

在销售点（POS）终端系统中，当客户携带所选购的商品至收银台处（自助或人工），其商品被扫描仪扫描其通用产品代码（UPC），系统显示商品的名称、单价、制造商等信息；可选地，客户可以输入该商品的实际购买数量。商品的元数据（UPC、商品名称、单价、制造商等）由系统管理员录入。请根据以上描述完成以下任务：

1. 编写名为CCommodity的类；数据成员：m_nId，类型long，在数据库中唯一标识该商品；m_pszUpc，通用产品代码，在整个系统中同样唯一，类型char*；m_pszName，商品名称，字符串char*，允许重复；m_dPrice，商品价格；m_pszManufacturer，商品制造商，C字符串，即类型为char*（现实中应对制造商单独建模）；
2. CCommodity类的所有成员均为private；提供相应的构造函数；提供相应数据成员的访问函数（均可读、可写）；对输出流友好，相应字段之间用逗号分隔；
3. CCommodity类的声明和实现相分离，头文件名为Commodity.h，实现文件为Commodity.cpp；
4. 编写名为pos.cpp的文件，其中包含入口点函数main()；在该函数中，从键盘输入一个商品的信息，并据此创建一个CCommodity对象，并将其输出到控制台。
5. 不得存在内存泄露。

6 Caesar

凯撒加密是最为古老的对称加密算法之一，其密钥为一个整数。密码本为52个英文字母再加上标点符号（简单起见，标点符号仅仅考虑逗号、分号、单引号和双引号）。在加密时将明文中的每个字符在密码本中向后移动密钥所指定的位置，用该位置处的字符替换原来的字符；解密时，将明文中的字符在密码本中向前移动。在加密/解密过程中，密码本中的字符构成环形。请根据以上描述完成以下任务：

1. 编写名为CCaesar的类，头文件为Caesar.h，实现文件为Caesar.cpp；
2. 以饥渴方式实现单例模式，即在整个程序运行期间仅仅存在CCaesar的一个实例，获取该类的唯一实例的函数为：CCaesar& GetInstance()；
3. 密码本全局共享；
4. 实现CCaesar的加密解/密成员函数：char* Encrypt(const char * const)/char*Decrypt(const char * const)；
5. 编写dencryptor.cpp，其中包含入口点main()函数；从键盘上输入一个仅仅包含52个字母和如前所述标点符号的字符串，调用CCaesar的加密函数对其进行加密，并将密文输出；然后对密文进行解密，并将解密结果输出以进行验证。

7 Hanoi

以下是对梵塔问题的描述：在A座上有自上而下、从小到大堆叠的n个圆盘，以B座为中转，将这n个圆盘从A座移动到C座，每次只能移动一个，并且在移动过程中不能出现大盘压小盘的现象。完成以下任务：

1. 编写名为CHanoi的类，以单例模式实现，采用惰性方式创建其全局唯一的实例（不考虑多线程环境），获取该唯一实例的函数为CHanoi* GetInstance()；
2. 在CHanoi类中，实现公有实例成员函数void Move(int n, char source, char station, char target)，其中，n表示圆盘数量，source表示源座，即n个圆盘初始所在的座，station表示中转座，初始为空；target表示目标座，即n个圆盘应该被移动至的座；
3. 在CHanoi类中，实现私有辅助成员函数void Show(int n, char source, char target)，其中，n表示第n号圆盘，source表示源座，target表示目标座；显示格式为：#n: source -> target；
4. 编写HanoiDriver.cpp，其中包含入口点main()方法，调用CHanoi的函数Move()，将5个圆盘，从'A'座以'B'座为中转，移动到'C'座上。

8 Guess

简单猜数值游戏。编写一个名为CRandom的类。其构造函数用系统时间对随机数生成器种子化以防止每次生成的随机数序列相同；成员函数Result Judge(int nValue)对用户从键盘上输入的整数值进行判断并返回结果。编写guess.cpp文件，在入口点main()中实例化CRandom的一个对象，提示用户输入一个整数。具体要求如下：

1. CRandom的构造函数接收两个参数：nBound，表示生成随机数的上界，默认值为10，即所生成的整数在范围[0, 10)中；nLimit，表示允许用户猜测的最大次数，默认值为10；在构造函数中用系统时间作为随机数生成器的种子（提示：time_t time(NULL)，获取系统时间，头文件为ctime）；要求声明和实现相分离，头文件Random.h，实现文件Random.cpp；

2. 在成员函数void Start()中，生成随机数并将其保存在成员变量m_nValue中，并将表示当前已猜测次数的成员变量m_nCount复位（提示：int rand(void)生成下一个随机数，范围[0, RAND_MAX)，头文件cstdlib。注意不要超过构造函数中所给定的上界）；

3. 在成员函数Result Judge(int nValue)中，递增用户已猜测的次数m_nCount，对用户所输入的整数进行判断，并将判断结果以枚举Result（位于CRandom类中）返回（EQUAL, nValue == m_nValue; LESS, m_nValue > nValue; GREAT, m_nValue < nValue; EXCEED, m_nCount >= m_nLimit）。

4. 在guess.cpp中，编写一个名为void Guess(CRandom& hRandom)的函数，提示用户输入一个整数，并调用CRandom的Judge()成员函数对用户所输入的整数进行判断，根据其返回值在终端（控制台）上输出：

- ① 正确（EQUAL）：“congratulation!”
- ② 太小（LESS）：“Sorry, too small, remaining: XXX”
- ③ 太大（GREAT）：“Sorry, too large, remaining: XXX”
- ④ 超限（EXCEED）：“Sorry, exhausted, value: YYY”

其中XXX表示剩余的猜测次数（提示：在CRandom中提供成员函数以访问其私有数据成员）而YYY表示未被猜中的随机值。在太小或太大情况下，允许用户继续猜测。

5. 在main()函数中创建CRandom的一个对象，在一个循环中，开始游戏，并在每一轮游戏之后询问用户是否继续，如果用户回答否，游戏结束。

9 Stuff

假设开发教务管理系统，需要对学生以及教职员工进行建模。用长整数long唯一地标识学生和教职员工。请根据以下要求编写表示学生和教职员工的类并在入口点中进行测试：

- 1. CStuff，抽象类。数据成员：m_nId, long，标识ID；m_pszNo，编号，char*；m_pszName名称，char*；m_nAge, int，年龄；m_nGender, char，性别（男：m；女：f）；m_nType，类型，枚举StuffType（STUDENT，学生；TEACHER，教师；EXECUTIVE，行政人员）；位于CStuff中；抽象成员函数void Print()；
- 2. CStudent，学生，CStuff子类。数据成员：m_pszMajor, char*，专业；

3. CTeacher, 教师, CStuff子类。数据成员: m_pszTitle, char*, 职称;
4. CExecutive, 行政人员, CStuff子类。数据成员: m_pszPosition, char*, 岗位;
5. 所有类的声明和实现相分离, 其头文件和实现文件的主文件名将为将相应类名去掉C前缀, 例如Student.h和Student.cpp; 所有数据成员均为私有, 提供相应的访问函数(可读、可写);
6. 编写university.cpp文件, 在入口点中分别创建CStudent、CTeacher和CExecutive类的一个实例, 其变量类型为CStuff *, 调用Print()成员函数输出相应的信息, 具体格式为:
(XXX: ID号, 编号, 姓名, 年龄, 性别, 类型, ...)
7. 不得存在内存泄露。

10 Poker

假设开发斗地主纸牌游戏。一副牌总共54张, 顺序为A、2~10、J~K、小王、大王; 除了大小王之外, 其它依次为红桃、黑桃、方块和梅花; 请完成以下任务:

1. 编写表示一张纸牌的类CCard, 它包含数据成员: m_nFlush, 表示花色; m_nNumber, 范围[1, 13] (大小王除外)。数据成员均为私有。花色由枚举Flush定义(位于CCard的头文件中, 但独立于CCard类): HEART, 0, 红桃; SPADE, 1, 黑桃; DIAMOND, 2, 方块; CLUB, 3, 梅花; JOKER, 4, 大小王(大王的m_nNumber为1, 小王的m_nNumber为0); 要求声明和实现相分离, 头文件Card.h, 实现文件Card.cpp
2. 类CCard的对象一旦被创建(表示发牌), 其数据成员不能被更改, 但可以获取其花色及其编号。花色默认为红桃, 编号为1 (A, ace);
3. 类CCard对输出流友好, 即其对象可以直接在输出流上输出, 形式为: “花色: 编号”;
4. 编写表示一手牌的类CDeck, 其私有数据成员std::vector<CCard*> m_hCards, 包含所有的纸牌; 成员函数bool Add(CCard* pCard)和bool Add(Flush nFlush, int nNumber)向其中添加一张牌, 如果已经有了18张牌, 返回false; 否则, 返回true; 成员函数bool Contain(Flush nFlush, int nNumber)确定是否包含参数给定的纸牌; 成员函数bool IsFull()确定是否已经包含了18张牌; 成员函数void Reset()执行复位, 即清空所有的纸牌; 类CDeck对输出流友好, 要求输出操作符可以直接访问CDeck的私有成员; 声明和实现相分离, 头文件为Deck.h, 实现文件为Deck.cpp;
5. 编写类CPoker, 私有数据成员: m_hDeckA、m_hDeckB、m_hDeckC, 类型为CDeck, 表示发给三个玩家的牌, 要求能够获取它们; 公有成员函数void Deal()实现为三个玩家随机发牌并确保在发牌之前, 将m_hDeckA、m_hDeckB、m_hDeckC复位; 可根据情况添加辅助函数;

要求声明和实现相分离，头文件Poker.h，实现文件Poker.cpp（提示：随机数的生成参考第8题的猜数）；

6. 编写PokerGame.cpp文件，在其入口点main()中创建CPoker的一个实例，调用其Deal()函数随机发牌，获取三个玩家的牌，并输出到控制台（终端）窗口。

7. 不得存在内存泄露。

11 Student

参见第9题，将CStuff的所有成员合并到CStudent类中，即CStudent不再从CStuff派生，为包含学号、姓名、性别、年龄和专业的独立类。请完成以下任务：

1. 类CStudent的声明和实现相分离，头文件为Student.h，实现文件为Student.cpp；该类对输出流类友好，即可以通过>>操作符输出学生信息；

2. 编写StudentMain.cpp文件，除入口点main()函数之外，编写void InputStudent(CStudent *pStudent)函数，在其中从控制台（终端窗口）输入学生的所有数据成员，每个数据均应给出相应的提示，例如，“Enter name: ”。在入口点main()中，创建std::vector<CStudent*> hVector变量，然后在一个循环中，调用InputStudent()输入学生信息，并将其添加到hVector中，之后提示用户是否继续，如果用户选择是，继续录入学生信息；否则，退出循环。录入结束后，打印输出每个学生的信息。

12 POS + Stuff

重写第5题和第9题中的相关类，对于字符串，用std::string替换，其它相同。