**数据结构实验报告**

**学号：\_\_\_\_CST17037\_\_\_\_\_ 姓名：\_\_\_陈锟\_\_\_**

# 实验题目

使用顺序表完成学生信息管理系统

# 实验要求

理解线性表的概念

理解顺序表存储结构概念和特点

掌握顺序表存储结构的建立、插入、删除、查询和输出等基本操作算法

# 方案设计

程序分为多个模块:

性别枚举:Gender{male,female}

数据类 Date,描述数据个体的结构

顺序表模板类 InitList,实现顺序表的算法模板类

学生信息表 StudentList,继承自线性表类,对线性表的拓展及功能包装

主函数 组织调用 StudentList类函数

# 方案实现

数据类 Date:

学号 姓名 string num,name;

性别 Gender gender ;

分数 int score ;

输出函数 printDate();格式化输出数据所有信息

顺序表模板类InitList:

保护型成员:

顺序表最大长度 unsigned int MAX\_LENGTH;

顺序表当前使用长度unsigned int length;

模板型date一维指针 T \*date;

公有型成员:

InitList();初始化数据,设定顺序表最大长度100

InitList(unsigned int maxLength);初始化数据,设定顺序表最大长度为maxLength

~InitList();释放数组内存空间,设定指针\*date=nullptr

void printList();循环输出所有date成员信息

void setEmpty();设置长度为零,"清空"顺序表

bool modifyNodeByNum(unsigned int num, T d);将第num位节点修改为 d,修改成功返回true

void invert();逆置顺序表

bool insertByNum(unsigned int num, T d);将节点d插入到第num位,原num位元素及之后原素向后一位,成功返回true

unsigned int findListByDateNum(Date d);根据节点d信息查找节点,并返回对应编号,未找到返回 0

bool delNode(unsigned int num);删除节点num,原num位元素及之后原素向前一位,成功返回true

unsigned int getLength();获取实际长度length

unsigned int getMaxLength();获取最大长度MAX\_LENGTH

学生信息表类StudentList:

私有型成员:

Date inputStudentInfo();输入学生信息,包含异常数据处理

StudentList();构造函数,调用父类构造函数创建最大长度100的线性表

void creatStudentList();创建学生信息表,输入数据,创建学生信息表

void printStudentList();输出所有学生信息

void prinStudentListTitle();打印学生信息表表头

void modifyStudnetNode();修改某学生信息

void invertStudentList();逆置学生信息表

void insertStudentList();插入学生信息表

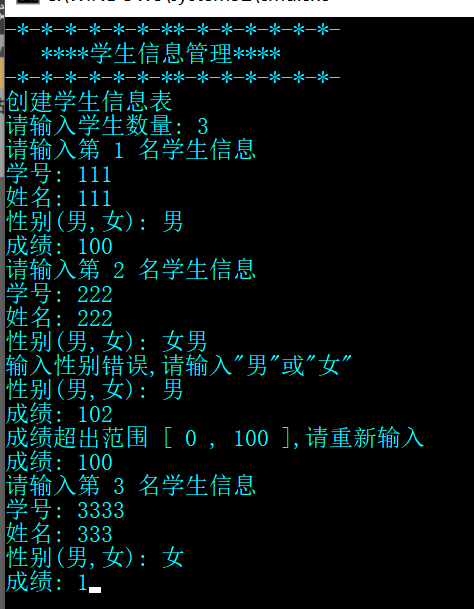
unsigned int findStudentList();查找学生信息

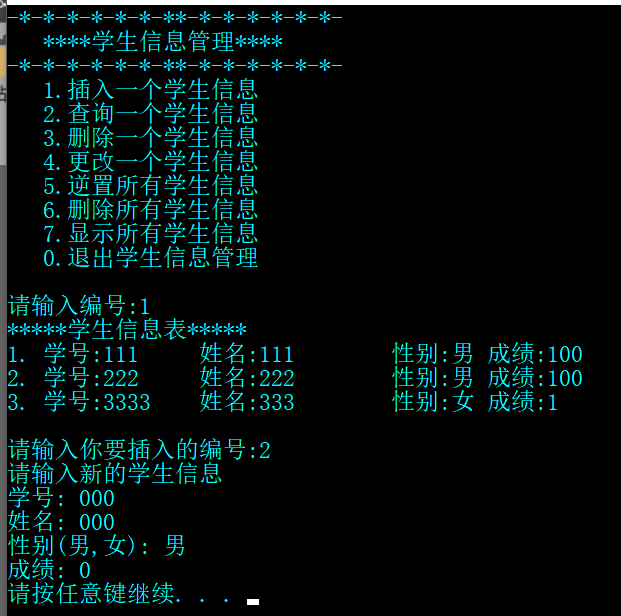
void delStudentList();删除学生信息

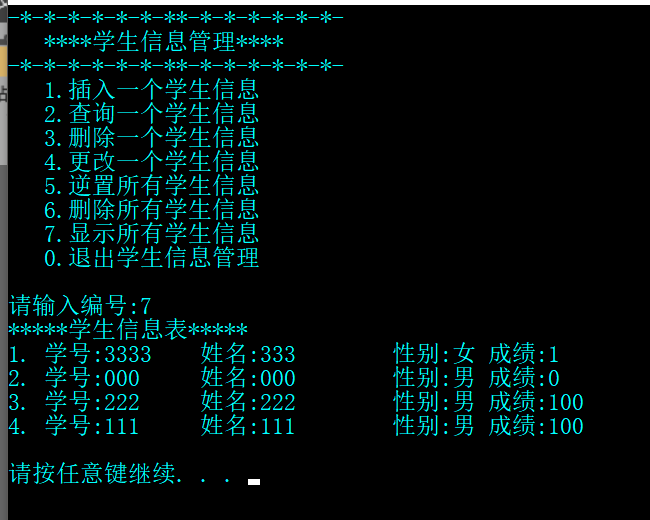
全局函数:

int inputNumber(string info);输入数字,以info提示输入数字,函数对输入数据进行异常提示处理,并返回有效数字

# 调试分析







时间复杂度分析:学生信息表使用顺序表实现

插入与删除操作,每次需要移动n-i个元素,故时间复杂度为O(n);

查找与访问操作,时间复杂度为O(1);

# 实验收获

练习了面向对象程序设计,复习了顺序表的知识