数据结构说明和

数据字典

|  |  |  |  |
| --- | --- | --- | --- |
| 对象名称 | 简要说明 | 对象成员/函数 | 功能说明 |
| myTextEdit | 文本及其编辑相关操作对象 | lineheAD\* getFirstLine(void), void setFirstLine(lineheAD\* lp) | 获取第一行行链表结点，设置第一行行链表结点 |
| lineheAD\* getNowLine()，void setNowLine(lineheAD\* lp) | 获取当前坐标所在行链表结点，设置当前行链表 |
| int getLineN() | 获取文本行数 |
| void getAxis()，void setAxis(int tR,int tC) | 输出当前光标所在坐标，设置光标位置，并且移动nowline行链表标记 |
| int getRow(),int getCol() | 返回行、列 |
| lineheAD\* axisToPtr(int row); | 用一个整形的行数获取对应的行链表结点的指针 |
| void insertStr(std::string str); | 在当前的坐标位置插入文本 |
| void delFULL(void) | 删除所有文本内容 |
| void delNL(int type); | 清空1/删除10当前行 |
| void delC(int isbackspace); | 删除当前坐标前/后的单字符 |
| std::string cutBlock(int r2,int c2); | 剪切操作，将从当前坐标到指定坐标为止的文本从内存中删除，并返回给用户 |
| void delBlock(int r2,int c2); | 将从当前坐标到指定坐标为止的文本删除， |
| std::string copyBlock(int r1,int c1,int r2,int c2); | 复制，获取指定区间的文本并返回 |
| void moveArray(lineheAD\* t,int index,int n); | 用移动数组内容的方式，删除指定的行链表中的块内容 |
| lineheAD\* firstLine; | 保存首行行链表结点 |
| lineheAD\* nowLine; | 记录当前行行链表 |
| temText\* WHOLETEXT; | 用于保存整个文本，行链表之间是一个双向单循环链表 |
| Int row, int col | 从1开始表示的行、列 |
| temText | 文本对象在内存中的存储，以lineheAD行链表结点为基本单位 | lineheAD\* newLine(void); | 在文本的最后插入一个行链表 |
| lineheAD\* insertLine(lineheAD\* preLine,CTYPE\* str,int n); | 在指定行的后面插入新的一行，同时可以指定初始化的字符串内容 |
| int linecounter; | 记录行数 |
| lineheAD \* firstLine; | 保存第一行行链表 |
| bool changeMode; | 记录模式是否自动换行 |
| bool deleteLine(lineheAD\* target); | 删除指定的行链表 |
| lineheAD | 文本中的行对象，行链表结点为双向链表，头结点的前缀链表为尾结点，尾结点的后继链表为空 | lineheAD \* getNext(void); void setNext(lineheAD\* np) | 获取后继结点；设置后继结点 |
| lineheAD\* getPre(void);  void setPre(lineheAD\* np) | 获取前缀结点；设置前缀结点 |
| int getSize(void); void setSize(int n); | 获取行链表内字符串的长度；设置当前行链表内字符串的长度 |
| CTYPE \*chs; | CTYPE为宏定义的字符类型，目前设置为char型， chs为字符类型的数组指针 |
| int arySIze; | 记录当前行的字符串数组大小（已经申请的空间大小） |
| int strSize; | 保存行内字符串的长度 |
| lineheAD \* next; | 保存行链表的后继结点指针 |
| lineheAD \* pre; | 保存行链表的前缀结点指针 |
| searchResult | 搜索对象，包含了替换功能 | searchResult(myTextEdit\* TEXT,std::string toFind,bool IC); | 重载的构造函数：指明搜索的文本对象（搜索范围），规定搜索的内容，记录是否忽略大小写 |
| int getNumber(void) | 获取搜索结果的数量 |
| int\* getSpecificOne(int INDEX) | 获取指定序号的搜索结果的坐标 |
| std::vector<int\*> allResults; | 保存所有的搜索结果，每个搜索结果用int[3]保存，内容为行序号，开始列坐标，结束列坐标 |
| bool ignoreCap; | 记录当前搜索是否忽略大小写 |
| void replace(int N,std::string newstr); | 将指定序号的搜索结果替换为新的指定字符串 |
| void search(void); | 搜索函数，获取所有的搜索结果坐标并保存 |
| nt Index\_KMP(std::string S,std::string T,int pos,const int\* const next, bool ignCap); | KMP搜索算法，使用时对某指定行链表中的字符串对指定目标字符串进行搜索 |
| void getNextVal(std::string substr,int\* next); | 计算KMP算法中需要的子串nextvalue数组 |
| myTextEdit\* TEXT; | 记录搜索的文本对象 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |