第二章 线性表

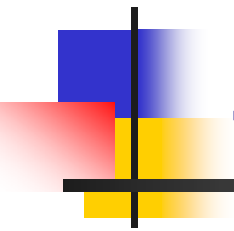
C++语言补充

- Template
- 构造函数
- 析构函数
- new & delete
- 函数重载

什么是线性表

线性结构特点：在数据元素的非空有限集中

- 存在唯一的一个被称作“第一个”的数据元素
- 存在唯一的一个被称作“最后一个”的数据元素
- 除第一个外，集合中的每个数据元素均只有一个前继元素
- 除最后一个外，集合中的每个数据元素均只有一个后继元素

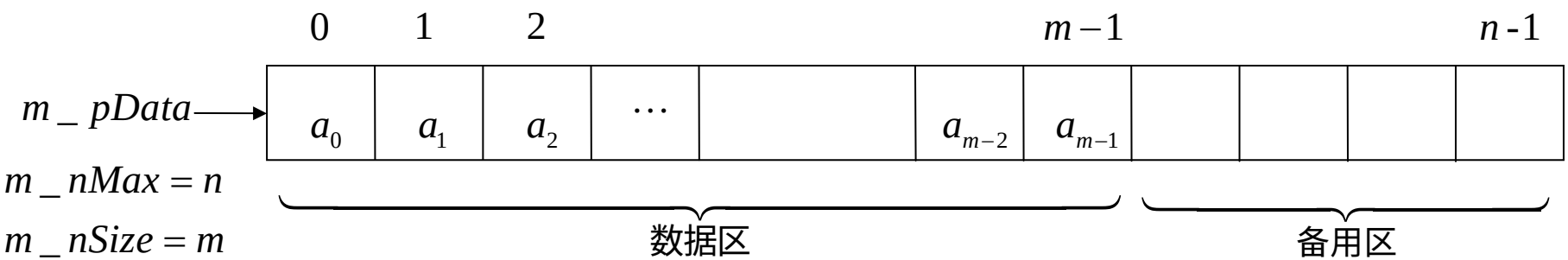


1. 顺序表

顺序表

- 数组
- 向量

class CVector



class

```
Class CVector
{
private:
   DataType *m_pData;
   int m_nMax, m_nSize;
public:
   CVector();
   ~CVector();
};
```

Methods

```
CVector(); CVector(int nMax); // 构造函数
~CVector();             // 析构函数
int Init();             // 初始化
int Free();             // 释放空间
int Clear();            // 数组设空
int IsEmpty();          // 判断数组是否空
int IsFull();           // 判断数组是否满
int Size();             //  （设置）数组的大小
int Resize(int nSize);              // 重设空间大小
DataType *GetData();                // 得到数组数据指针
DataType GetItem(int pos);      // 获取数组某元素
int Locate(DataType item);      // 判断元素位置
int InsertAt(int pos, DataType item);       // 在某位置插入元素
int Pushback(DataType item); // 在末尾插入元素
DataType Popback();                 // 得到末尾元素并删除
int Erase(int pos);                 // 删除某元素
int EraseBetween(int start, int end);       // 删除某范围段所有元素
```

Other Methods

- Purge()
  - 从顺序表中删除重复的数据
- BubbleSort()
  - 冒泡排序
  - 复杂度

练习: 动态字符串

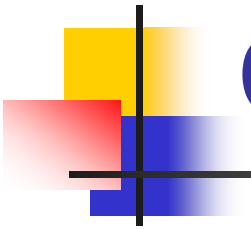
- C语言字符串
  - char str[13]=“Hello, world!”;
  - char *pStr = str;
- 字符串函数
  - gets(char str);
  - puts(char *str);
  - strcpy(char *str1, char *str2);　　//字符串拷贝
  - strcat(char *str1, char *str2);　　//字符串连接
  - strcmp(char *str1, char *str2);　　//字符串比较
  - strlen(char *str);　　//字符串求长

动态字符串

```
class CMyString
{
private:
   int m_nSize;
   char *m_pStr;
};
```

Methods

- CMyString();
- ~CMyString();
- int Size();
- int Empty();
- char *GetString();
- int *Compare(CMyString str);
- Concat(), SubString(), Insert(), Delete(), Clear()…

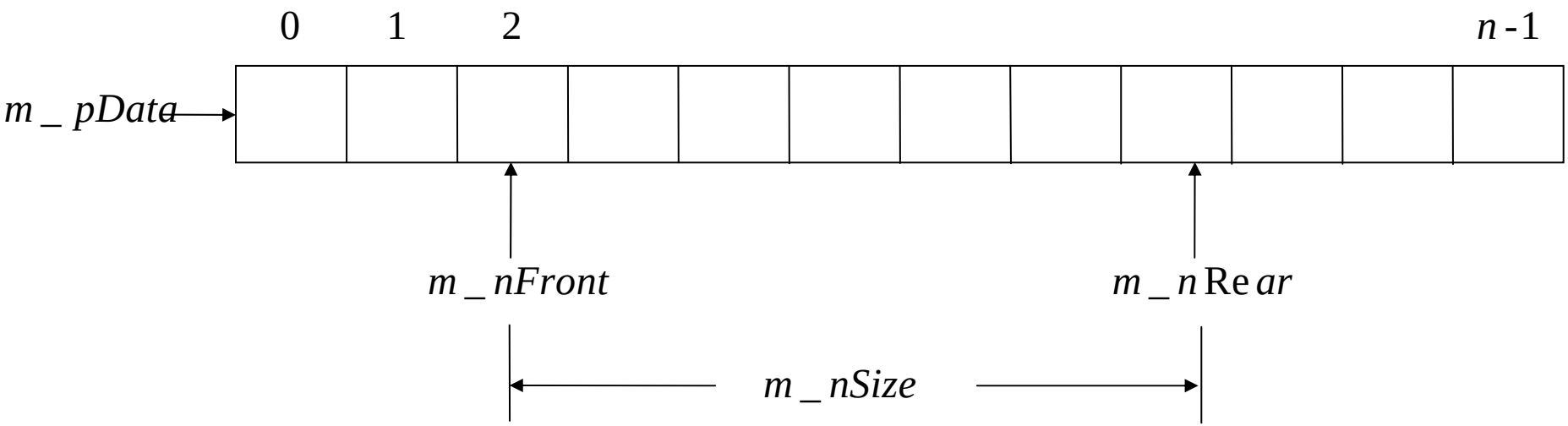


Q&A

2. 顺序队列

顺序队列

- 一种特殊的线性表
  - 只能在表的一端插入，另一端删除，是先进先出的线性表
  - 头指针和尾指针
- First come, first serve
- 优点
  - 循环结构
  - 删除时不需移动元素


0
1
2
n-1
m_pData
m_nFront
m_nRear
m_nSize

class

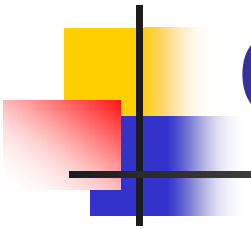
```
Class CQueue
{
   DataType *m_pData;
   int m_nMax;
   int m_nFront, m_nRear, m_nSize;
};
```

Methods

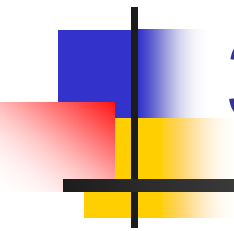
- CQueue(); // 构造函数
- ~CQueue(); // 析构函数
- int SetQueue(int n); // 设置队列大小
- int Free(); // 释放空间
- int Size(); // 队列大小
- int Empty(); // 判断队列是否空
- int Full(); // 判断队列是否满
- DataType GetData(); // 获取数据？
- int Insert(DataType item); // 增加元素
- DataType Delete(); // 删除元素
- void Clear(); // 清空
- int Resize(int nMax); // 重置队列空间大小?

自学

- pp. 138 模拟排序机
- pp. 140 队列与文件



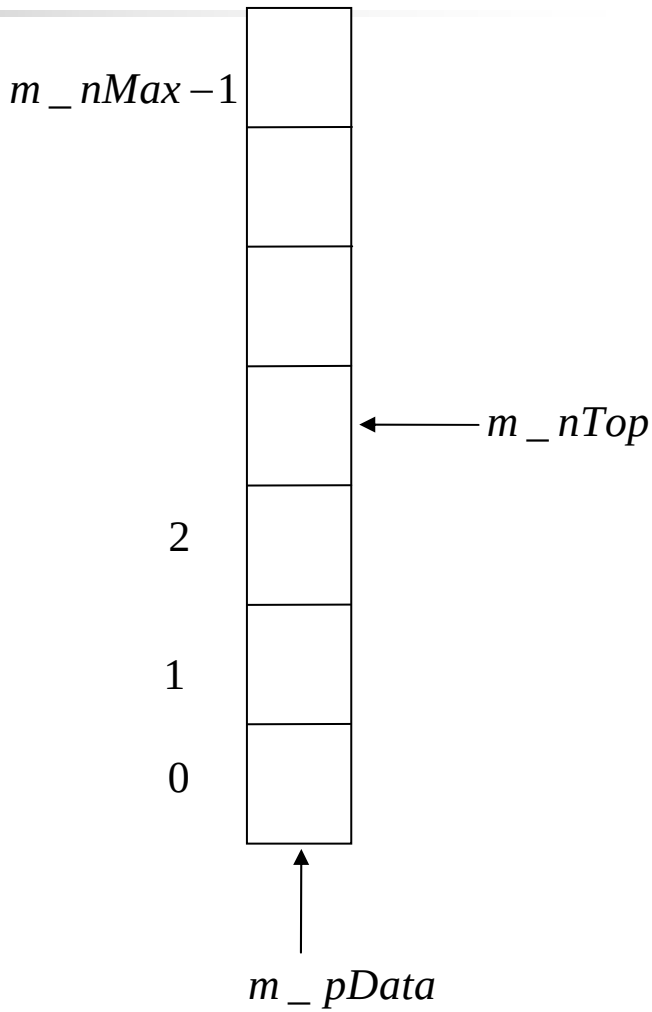
Q&A



3. 顺序栈

顺序栈

- 一种特殊的线性表
  - 只能在表的一端插入和删除，是后进先出的线性表
  - 进栈和出栈



class

```
class CStack
{
   DataType *m_pData;
   int m_nMax;
   int m_nTop;
};
```

Methods

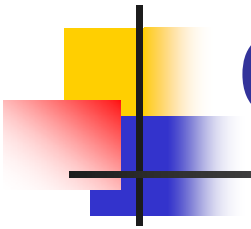
- CStack()/CStack(int n); // 构造函数
- ~CStack(); // 析构函数
- int SetSize(int n); // 设置栈的大小
- int Free(); // 释放空间
- int Size(); // 栈的大小
- int Empty(); // 判断是否空
- int Full(); // 判断是否满
- int Push(DataType item); // 压栈
- DataType Pop(); // 出栈
- DataType GetPeek(); // 取栈顶元素
- int Clear(); // 清空栈

练习

- pp.144 进制数转换
- pp. 145 判断字符串是否中心对称

Project #1

- Dest Calculator
  - pp.146 中缀表达式求值
- Requirement
  - Group members: 3
  - Operators: (,),+,-,*,/,%,^
- Deadline: Monday, Sep. 26
- Samples
  - 3.0*2+9-(1.0+2*3+3/1.1)
  - (3*5*(4+8)%2)



Q&A