

//5.2.6节 表达式树类ExprTree的定义

/*表达式树类ExprTree的声明*/

```
class ExprTree
{
private:
    BinTree<string> *tree;           // 二叉树
public:
    void CreateExprTree( );          //创建一棵表达式树
    void GetValue(BinTreeNode<string> *t,int &value); //计算表达式值
    BinTree<string> *GetTree() { return tree ; }
    void SetTree(BinTree<string> * t) { tree=t ; }
};
```

//5.2.6节 算法 CreatExpressTree

//利用辅助堆栈S构造表达式对应的二叉树，算法结束时指针tree指向二叉树根结点

```
void ExprTree::CreateExprTree()
{
    tree=new BinTree<string>();
    BinTreeNode<string> *root;
    string op;
    cin >> op ;    //读入后缀表达式
    AStack<BinTreeNode<string>*> s;
    BinTreeNode<string> * lpr,*rpr,*p;
    while( op !="#" )
    {
        if(op=="+" || op=="-" || op=="*" || op=="/" )
        {
            s.Pop(rpr);
            s.Pop(lpr);
            p=new BinTreeNode<string> (op);
            p->SetLeft(lpr);
            p->SetRight(rpr);
            s.Push(p);
        }
        else
        {
            p=new BinTreeNode<string> (op);
            p->SetLeft(NULL);
            p->SetRight(NULL);
            s.Push(p);
        }
        cin >> op ;
    }
    s.Pop(root);
    tree->SetRoot(root);
};
```