//5.5 节 树类 Tree 的定义

```
/*树结点类 TreeNode 声明*/
template<class T>
class TreeNode
{
private:
    T data ;
    TreeNode<T> *firstChild, *nextBrother ;
public:
     TreeNode ( T value = 0, TreeNode<T> *L = NULL , TreeNode<T> *R = NULL)
          : data (value ) , firstChild ( L ) , nextBrother( R ) { }          //  构造函数
    TreeNode<T> * GetFirstChild( )const { return firstChild ; }
    void SetFirstChild(TreeNode<T> *t){ firstChild = t ; }
    TreeNode<T> * GetNextBrother( )const { return nextBrother ; }
    void SetNextBrother(TreeNode<T> *t){ nextBrother = t ;}
    T& GetData() {    return    data;   }
    void SetData(const T & item){ data = item ; }
} ;
/*树类 Tree 的声明*/
template<class T>
class Tree
{
private :
    TreeNode<T> *root ;   //  根结点
    T stop;
public :
    Tree ( ){ root =NULL ; } //  构造函数
    TreeNode<T> * FirstChild (TreeNode<T> *p);          //返回结点 p 的大儿子结点
    TreeNode<T> * NextBrother ( TreeNode<T> *p);     //返回结点 p 的下一个兄弟结点

    void PreOrder (TreeNode <T>    * t);              //先根遍历以 t 为根结点的树
    void NorecPreOrder (TreeNode<T> * t    );     //非递归先根遍历以 t 为根结点的树
    void LevelOrder ( TreeNode<T>    * t );   //层次遍历以 t 为根结点的树

    TreeNode<T> * FindTarget ( TreeNode<T> *t , T target ); //在以 t 为根结点的树中搜索值为 target 的结点
    TreeNode<T> * FindFather ( TreeNode<T> *t , TreeNode<T> *p ); //在树 t 中搜索 p 的父结点
    void DelSubtree ( TreeNode<T> *t ,TreeNode<T> *p );      //在以 t 为根的树中删除以 p 为根的子树
    void Del ( TreeNode<T> *p );
    //其他操作
    T GetStop() { return stop ; }
    void SetStop(T tostop) {    stop=tostop ; }
    TreeNode<T> * GetRoot(){ return root; }
    void SetRoot( TreeNode<T> * t){ root=t; }
    TreeNode<T> *    Create() ;          //  创建一棵树
    void    CreateTree(T tostop) ;
    void PostOrder (TreeNode <T>    * t);              //后根遍历以 t 为根结点的树

} ;
```

//5.5.2 节 算法 GFC

```
//搜索指针 p 所指结点的大儿子结点
template<class T>
TreeNode<T> *    Tree<T> :: FirstChild (TreeNode<T> *p)
{
    if ( p!=NULL && (p->GetFirstChild()) != NULL)
          return p->GetFirstChild() ;
    return NULL ;
}
```