

2014 자료구조 실습과제 11

[실습 11] 이진 탐색 트리의 구현

강의시간에 구현하던 이진 탐색트리의 다음 기능들을 완성하시오.

1) 이진 탐색 트리의 Node 클래스를 완성하시오.

- 노드 삽입 기능
- 트리의 높이 계산
- 트리의 노드수 계산
- 순회 (Preorder, Inorder, Postorder)

2) 이진 탐색 트리 클래스를 완성하시오.

- 노드 삽입 기능
- 트리의 높이 계산
- 트리의 노드수 계산
- 순회 (Preorder, Inorder, Postorder)

3) 각 구현된 기능들이 잘 동작하는지를 보이시오. (탐색, 노드 삭제 기능 제외)

<main() 함수 구현 예>

```
#include "BinTree.h"
#include <conio.h>
#include <stdlib.h>

void Usage()
{
    system("cls");
    printf("=====\n");
    printf(" Binary Tree Test Program Usage:\n\n");
    printf(" a num : append num node.\n");
    printf(" d num : delete num node.\n");
    printf(" s num : search num node.\n");
    printf(" i :   inorder print.\n");
    printf(" p :   preorder print.\n");
    printf(" t :   postorder print.\n");
    printf(" h :   print this message.\n");
    printf(" q :   exit.\n");
    printf("=====\n");
}

void main(void)
{
    CBinTree tree; //트리객체 생성
    char    ch;
    int     num, nmatch;
    char    str[100];

    Usage();
    do {
        printf(">>> ");
        gets(str);
        nmatch = sscanf(str, "%c%d", &ch, &num);
        switch (str[0]) {
            case 'h': Usage(); break;
            case 'p': tree.Print( PRE_ORDER ); break;
            case 't': tree.Print( POST_ORDER ); break;
            case 'i': tree.Print( IN_ORDER ); break;
            case 'a': tree.Insert(num); tree.Print( IN_ORDER ); break;
            case 'd': tree.Delete(num); tree.Print( IN_ORDER ); break;
            case 's': tree.Search(num); tree.Print( IN_ORDER ); break;
        }
    } while (str[0] != 'q');
}
```