

자료구조 및 알고리즘 Term Project

임베디드 시스템 공학과
2015146003
김기덕

1. 주제

독서실 예약결재시스템.

자리는 간단하게 10개이며 현금과 카드결재 선택이 가능하다.

예약 후 좌석변경이 가능하며 예약취소도 가능하다.

예약정보확인은 예약할 때 입력한 이름을 통해서 확인이 가능하다.

2. 소스코드

```
#define _CRT_SECURE_NO_WARNINGS
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include <Windows.h>
#define MAX_NAME_LENGTH 50
char seats[10] = { 'o', 'o', 'o', 'o', 'o', 'o', 'o', 'o', 'o', 'o' };

typedef struct
{
    char name[MAX_NAME_LENGTH];
    int seat;
    char pay[5];
}element;

typedef struct treeNode
{
    element key;
    struct treeNode *left;
    struct treeNode *right;
}treeNode;

treeNode* insertKey(treeNode *p, element key)
{
    treeNode *newNode;
    int compare;

    if (p == NULL)
    {
        newNode = (treeNode*)malloc(sizeof(treeNode));
        newNode->key = key;
        newNode->left = NULL;
        newNode->right = NULL;
        return newNode;
    }
    else
```

```

    {
        compare = strcmp(key.name, p->key.name);
        if (compare < 0)
            p->left = insertKey(p->left, key);
        else if (compare > 0)
            p->right = insertKey(p->right, key);
        else
            printf("\n 이미 예약된 이름입니다! \n");
        return p;
    }
}

void insert(treeNode** root, element key)
{
    *root = insertKey(*root, key);
}

void deleteNode(treeNode *root, element key)
{
    treeNode *parent, *p, *succ, *succ_parent;
    treeNode *child;

    parent = NULL;
    p = root;
    while ((p != NULL) && (strcmp(p->key.name, key.name) != 0))
    {
        parent = p;
        if (strcmp(key.name, p->key.name) < 0)
            p = p->left;
        else
            p = p->right;
    }

    if (p == NULL)
    {
        printf("\n 예약된 좌석이 없습니다!");
        return;
    }

    if ((p->left == NULL) && (p->right == NULL))
    {
        if (parent != NULL)
        {
            if (parent->left == p)
                parent->left = NULL;
            else parent->right = NULL;
        }
    }
}

```

```

        }
        else root = NULL;
    }

    else if ((p->left == NULL) || (p->right == NULL))
    {
        if (p->left != NULL)
            child = p->left;
        else
            child = p->right;

        if (parent != NULL)
        {
            if (parent->left == p)
                parent->left = child;
            else
                parent->right = child;
        }
        else
            root = child;
    }

    else
    {
        succ_parent = p;
        succ = p->right;
        while (succ->left != NULL)
        {
            succ_parent = succ;
            succ = succ->left;
        }
        if (succ_parent->left == succ)
            succ_parent->left = succ->right;
        else
            succ_parent->right = succ->right;

        p->key = succ->key;
        p = succ;
    }
    free(p);
}

```

```

treeNode* searchBST(treeNode* root, element key)
{
    treeNode* p;
    int compare;

```

```

p = root;

while (p != NULL)
{
    compare = strcmp(key.name, p->key.name);
    if (compare < 0)
        p = p->left;
    else if (compare > 0)
        p = p->right;
    else
    {
        printf("-----예약정보-----");
        printf("\n\t예약자 명 : %s", p->key.name);
        return p;
    }
}
return p;
}

```

```

void menu()
{
    printf("\n*-----*");
    printf("\n\t1 : 좌석 예약");
    printf("\n\t2 : 예약 조회");
    printf("\n\t3 : 좌석 변경");
    printf("\n\t4 : 예약 취소");
    printf("\n\t5 : 종료");
    printf("\n*-----*\n ");
}

```

```

void view_seats()
{
    int cnt = 0;
    printf("\n\n      KPU 도서관");
    printf("\n-----좌석 배치도-----");
    printf("\n=====");
    printf("1 2  3 4  5 6  7 8  9 10\n");
    for (int i = 0; i < 10; i++)
    {
        printf("%c ", seats[i]);
        cnt++;
        if (cnt == 2)
        {
            printf(" ");
            cnt = 0;
        }
    }
}

```

```

    }
    printf("\n=====\\n");
}

const char* payment(int i)
{
    char cash[5] = "CASH";
    char card[5] = "CARD";
    if(i == 1) return cash;
    else if (i == 2) return card;
}

void main()
{
    system("title 2015146003_Kimkideok");

    element e;
    treeNode *root = NULL, *temp = NULL, *temp1 = NULL;
    int choice, pay, pos, pos1, rm, chg;

    do {
        view_seats();
        menu();
        scanf("%d", &choice);

        switch (choice)
        {
            case 1:
                printf("\n-----좌석예약-----\\n");
                while (1) {
                    printf("\n좌석을 선택하세요\\n(o : 사용가능, x : 사용중) : ");
                    scanf("%d", &pos);
                    if (seats[pos - 1] == 'x') printf("이미 사용중인 좌석입니
다.\\n다시 선택해주세요.\\n");
                    else if (pos > 10) printf("없는 좌석입니다.\\n다시 선택해주세
요.\\n");
                    else if (pos < 1) printf("없는 좌석입니다.\\n다시 선택해주세
요.\\n");
                    else break;
                }
                seats[pos - 1] = 'x';
                e.seat = pos;
                printf("\n예약자 명을 입력하세요 : ");
                scanf("%s", e.name);
                while (1)
                {

```

card\n");

```
printf("\n결제수단을 선택하세요 : \n\t1 : cash \n\t2 :  
card\n");  
  
scanf("%d", &pay);  
if (pay == 1)  
{  
    printf("\n계좌이체를 선택했습니다\n\n");  
    break;  
}  
else if (pay == 2)  
{  
    printf("\n카드결제를 선택했습니다\n\n");  
    break;  
}  
else printf("\n잘못입력하셨습니다.\n");  
}  
strcpy(e.pay, payment(pay));  
insert(&root, e);  
break;  
case 2:  
printf("\n-----예약조회-----\n");  
printf("\n예약자 명을 입력하세요 : ");  
scanf("%s", e.name);  
temp = searchBST(root, e);  
if (temp != NULL)  
{  
    printf("\n\t예약좌석 : %d", temp->key.seat);  
    printf("\n\t결제종류 : %s\n", temp->key.pay);  
    printf("-----\n");  
}  
else  
    printf("\n조회할 수 없는 이름입니다.\n");  
break;  
case 3:  
printf("\n-----좌석변경-----\n");  
printf("\n예약자 명을 입력하세요 : ");  
scanf("%s", e.name);  
temp = searchBST(root, e);  
if (temp != NULL)  
{  
    printf("\n\t예약좌석 : %d", temp->key.seat);  
    printf("\n\t결제종류 : %s\n", temp->key.pay);  
    printf("-----\n");  
    printf("\n변경할 좌석을 입력하세요 : ");  
    pos1 = temp->key.seat;  
    scanf("%d", &pos);  
    printf("\n%d번 자리로 변경하시겠습니까? : \n\t1: 예\n\t2: 아
```

```

        니요\n", pos);

        scanf("%d", &chg);
    }
    else
        printf("\n조회할 수 없는 이름입니다.\n");
    if (chg == 1)
    {
        temp->key.seat = pos;
        seats[pos1 - 1] = 'o';
        seats[pos - 1] = 'x';
    }
    else:
    break:
case 4:
    printf("\n-----예약취소-----\n");
    printf("\n예약자 명을 입력하세요 : ");
    scanf("%s", e.name);
    temp1 = searchBST(root, e);
    if (temp1 != NULL)
    {
        printf("\n\t예약좌석   : %d", temp->key.seat);
        printf("\n\t결제종류   : %s\n", temp->key.pay);
        printf("-----\n");
        printf("\n예약을 취소하시겠습니까? \n\t1: 예\n\t2: 아니요\n");

        scanf("%d", &rm);
    }
    else
    {
        printf("\n조회할 수 없는 이름입니다.\n");
    }
    if (rm == 1)
    {
        pos = temp->key.seat;
        seats[pos - 1] = 'o';
        deleteNode(root, e);
    }
    else:
    break:
    }
    while (getchar() != '\n');
}
while (choice != 5);
}

```


3. 코드설명

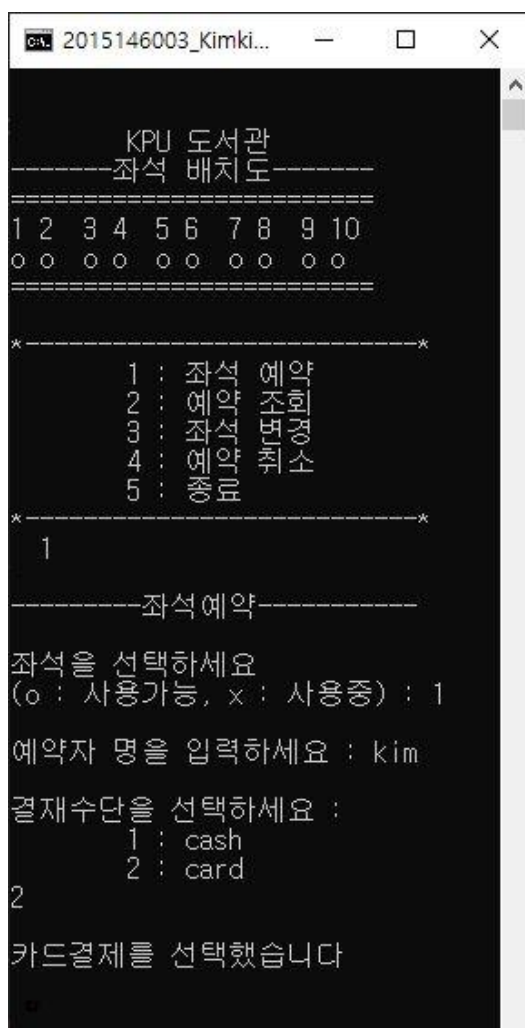
기본구조는 예제의 영어사전 코드에서 가져왔다.

insertNode로 노드 삽입, deleteNode로 노드삭제, searchBST로 노드탐색을 한다.

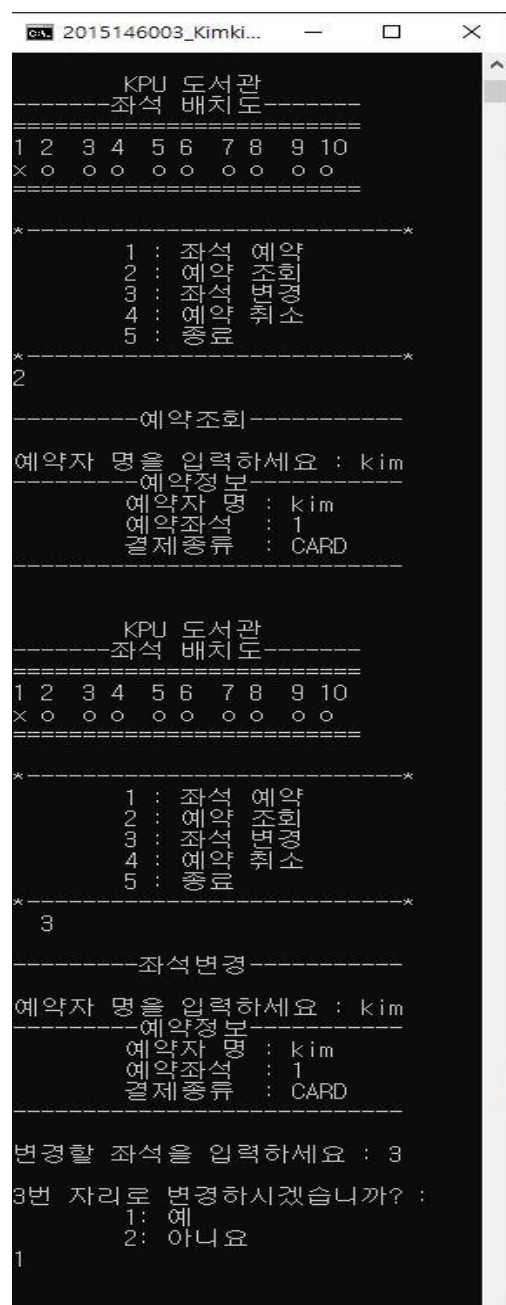
menu함수는 메뉴를 보여주고 view_seats함수는 현재 자리현황표를 보여준다.

payment함수는 결제 선택창을 보여주고 선택하게 한다. 선택한 좌석에는 x표시가 되고 좌석변경을 하면 변경된 좌석에 x표시, 기존좌석에는 o표시가 된다. 예약취소를 하면 예약취소한 좌석이 o가 된다.

4. 실행화면



1. 좌석예약
2. 예약조회 및 좌석변경
3. 예약조회 및 예약취소
4. 예약취소후 확인 가능한 좌석배치도의 변경



```
2015146003_Kimkideok
1
      KPU 도서관
-----좌석 배치도-----
=====
1 2 3 4 5 6 7 8 9 10
o o x o o o o o o o
=====

*-----*
1 : 좌석 예약
2 : 예약 조회
3 : 좌석 변경
4 : 예약 취소
5 : 종료
*-----*

2

-----예약조회-----
예약자 명을 입력하세요 : kim
-----예약정보-----
예약자 명 : kim
예약좌석 : 3
결제종류 : CARD
-----

      KPU 도서관
-----좌석 배치도-----
=====
1 2 3 4 5 6 7 8 9 10
o o x o o o o o o o
=====

*-----*
1 : 좌석 예약
2 : 예약 조회
3 : 좌석 변경
4 : 예약 취소
5 : 종료
*-----*

4

-----예약취소-----
예약자 명을 입력하세요 : kim
-----예약정보-----
예약자 명 : kim
예약좌석 : 3
결제종류 : CARD
-----

예약을 취소하시겠습니까?
1: 예
2: 아니요
1
```

```
2015146003_Kimkideok
1
      KPU 도서관
-----좌석 배치도-----
=====
1 2 3 4 5 6 7 8 9 10
o o o o o o o o o o
=====

*-----*
1 : 좌석 예약
2 : 예약 조회
3 : 좌석 변경
4 : 예약 취소
5 : 종료
*-----*
```

5.자기평가서

1. 본 과제를 통해 무엇을 배웠습니까 ?
이진트리구조와 그 사용법을 확실히 알게 되었다. 추가로 모르는 부분을 따로 찾아가며 공부를 하면서 현장에서 트리구조는 유용하게 사용한다는 것을 알았고 앞으로의 다른 과목 팀 프로젝트를 하면서도 유용하게 사용할 수 있을 거 같다.
2. 본 과제를 해결하기 위해 어떤 방식으로 접근하셨습니까?
맨땅에 헤딩하듯이 백지상태로의 코드작성은 힘들고 비 효율적일거 같아서 영어사전의 코드를 사용했다. 내가 짜고자 하는 코드의 목적에 맞게 구조체의 메소드를 변경하고 그에 맞게 나머지 코드들도 수정하는 식으로 코드작성을 하였다. 하지만 이진트리의 구조에 대해 완벽하게 이해하고 있지 않다면 코드수정 또한 힘들기 때문에 수업을 들으면서 어려웠던 부분을 다시 찾아보고 공부하며 이진트리의 구조에 대해 다시 한번 공부하였고 공부하고 난 뒤에 코드 수정을 하기 위해 기존의 코드를 보니 어디를 어떻게 수정해야할지 감이 잡혔다.
3. 본 과제를 수행하는 과정에서 경험한 성공요인과 실패 요인은 무엇입니까?
성공요인으로서는 기존의 코드를 잘 이용한 것 같다. 원래 대부분의 코드가 백지상태부터 시작하는 것을 많지 않고 여러 코드들의 잘 조합하는 것이 중요한데 이번에는 영어사전 코드를 갖고와서 잘 변형한거 같다. 좌석 변경을 할 때 노드 내의 메소드를 변경해주는 것에 대해 조금 헤맸는데 오류를 잘 잡아내서 코드를 잘 마무리 지은 것 같다.