**자료구조 HW1**

**2014104272 이준영**

**1. main.c 코드 설명 (볼드체 주석 부분)**

// ADT Stack

#include <stdio.h>

#include "ADT\_stack.h"

#include "chart.h"

#define REVERSE 0 // reverse

int main() {

// new type definition

typedef struct {

char name[20];

int score;

} STD;

// prepare 5 data

STD student[5] = {

{"James", 95},

{"Yoosoo", 87},

{"Paul", 93},

{"Peter", 76},

{"Park", 100},

};

#ifndef REVERSE

generate\_chart\_header();

// me my boss my score

generate\_chart\_node(student[0].name, student[0].name, student[0].score);

generate\_chart\_node(student[1].name, student[0].name, student[1].score);

generate\_chart\_node(student[2].name, student[1].name, student[2].score);

generate\_chart\_node(student[3].name, student[2].name, student[3].score);

generate\_chart\_node(student[4].name, student[3].name, student[4].score);

generate\_chart\_footer();

#else

//Stack Creation

STACK\* stack1;

stack1 = create\_stack(); **//stack1을 생성**

//push them

int i;

for(i=0;i<sizeof(student)/sizeof(student[0]);i++){

**//push를 student의 배열 개수만큼 반복하도록 조건을 주었다 (student[6]으로 바뀌면 6번 ...)**

push(stack1,&student[i]);

**//&student[0]부터 stack1에 push함, 즉 &student[0]이 stack1의 가장 바닥에 깔림**

}

//pop them

STD\* boss = NULL;

STD\* std;

generate\_chart\_header();

while(stack1->count!=0){ **//stack1 안에 저장되어 있는 data의 개수가 0이 될 때까지 반복**

if(boss==NULL){ **//while문을 처음 실행할 때 아래의 코드가 실행되도록 조건문을 이용**

std = (STD\*)pop(stack1);

boss = std;

**//std에 stack1의 top인 &student[4]의 값이 들어가고 이 경우 boss도 같은 값을 가짐**

}else{

boss = std;

std = (STD\*)pop(stack1);

**//2번째 반복부터는 boss에 이전의 std값을 넣고 std는 pop을 하여 다음 data를 넣음**

}

**//이 while문의 경우 5번 반복하는데 이 때 std와 boss에 저장된 값을 자세히 살펴보면**

**//1번째: std = &student[4], boss = &student[4]**

**//2번째: std = &student[3], boss = &student[4]**

**//3번째: std = &student[2], boss = &student[3]**

**//4번째: std = &student[1], boss = &student[2]**

**//5번째: std = &student[0], boss = &student[1]**

generate\_chart\_node(std->name,boss->name,std->score);

**//위의 함수를 이용하여 HTML로 출력**

}

generate\_chart\_footer();

#endif

return 0;

}

**2. 보충 설명**

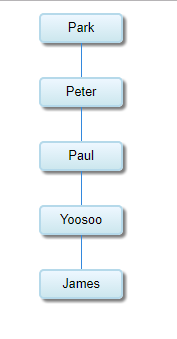
스택을 이용하여 student의 name 부분만 생각하고 표현하면 다음과 같이 설명할 수 있습니다.

stack1에 James(student[0].name)부터 넣으면(push) top은 Park(student[4].name)가 됩니다. 그 다음 stack1에서 데이터를 빼면(pop) 가장 마지막에 넣어서 top에 위치한 Park(student[4].name)부터 나오게 됩니다. 즉, stack1에 넣은 data 순서와 반대로 출력됩니다.

(James -> Yoosoo -> Paul -> Peter -> Park 순서로 stack1에 넣으면 역순(reverse)인

Park -> Peter -> Paul -> Yoosoo -> James 순서로 나옴)

**3. 결과**



**4. 느낀점**

과제를 하면서 수업시간에 배운 스택 내용을 복습하고 구조를 생각하며 코드를 짜는 과정을 통해 몰랐던 부분을 깨닫고 공부할 수 있었습니다. 그리고 과제를 하면서 스택이 어떠한 정보를 넣고 빼면 정보가 역순으로 나오는 성질을 알고 실습해보는 것이 이번 과제의 목적이라고 느꼈습니다. 처음에는 과제에서 원하는 내용이 무엇인지 이해가 잘 되지 않아서 시간을 많이 낭비하였는데 다음 과제부터는 원하는 내용을 명확하게 파악하는데 우선순위를 둬야겠다고 다짐했습니다.