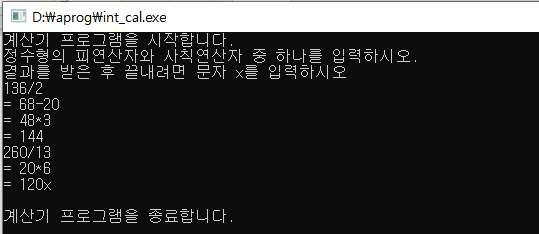
알고리즘 과제09 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

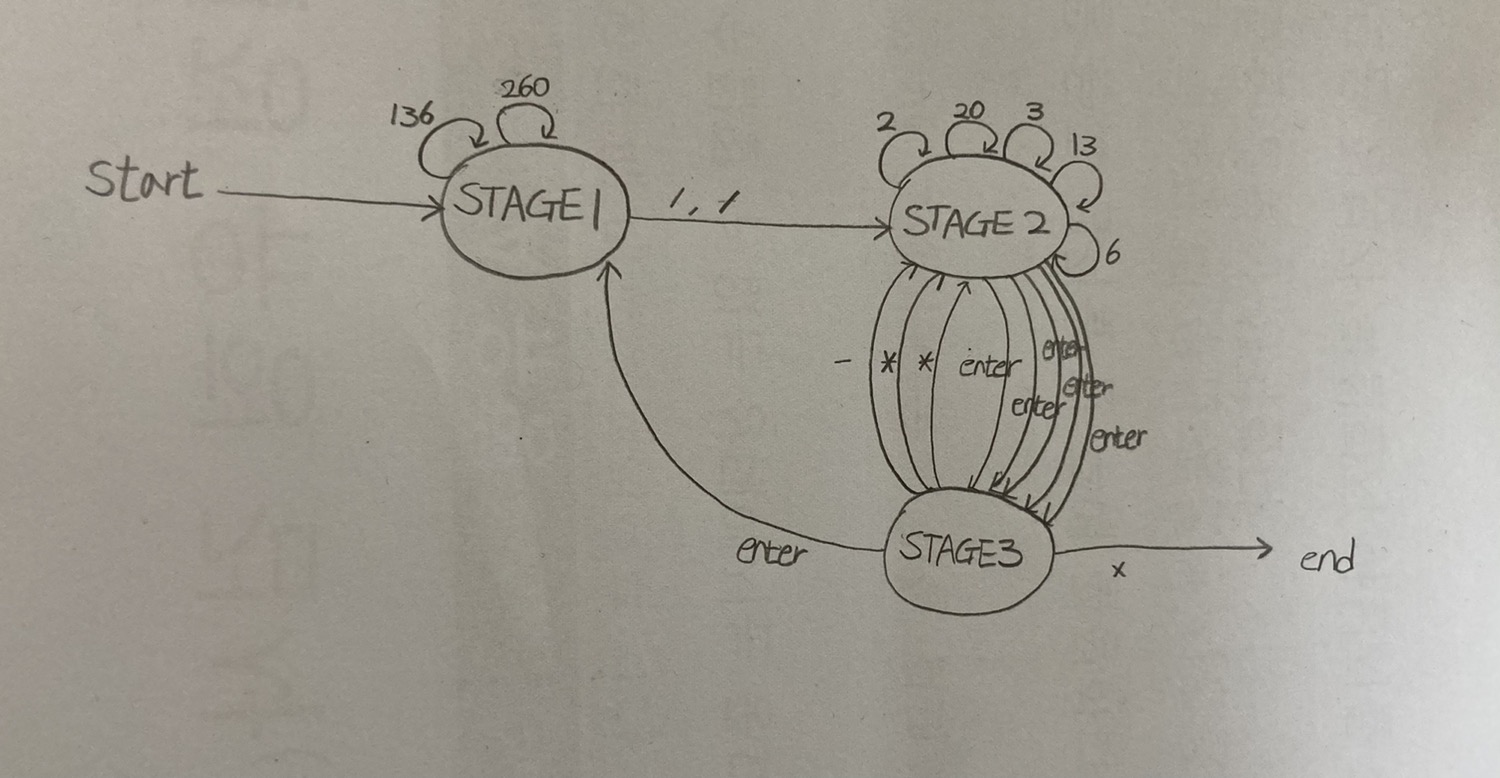
**9주차 수업 정리과제**

1. [정수 단순 계산기 프로그램 분석]

(1) 동영상 수업에서 설명한 정수 단순 계산기 프로그램]의 코드를 이해하고 다음 실행 화면의 입력을 예로 이에 해당하는 STD(상태전이도)를 그리시오

//컴퓨터로 그림을 그리기 어려우면 손으로 그려 사진찍어 제출하시오





(2) 정수 단순 계산기 프로그램에서 피연산자 숫자를 만드는 역할을 하는 다음 함수 get\_int() 정의하는 함수코드를 쓰시오

int get\_int(int current\_operand, char c)

{

int new\_value;

new\_value = c - '0';

current\_operand \*= 10;

current\_operand += new\_value;

return current\_operand;

}

2. [실수 단순 계산기 프로그램 분석] 수업 동영상을 잘 듣고 물음에 답하시오

(1) 정수계산기 프로그램에서 없었던 실수를 처리하기 위하여 실수 단순 계산기 프로그램에서 사용한 status 변수의 역할을 설명하시오

소수점 숫자처리를 위해 status 변수를 활용하는데 status는 소수점의 자릿수를 알려준다. status는 0으로 초기화되는데 소수점을 만나면 status를 1로 변경한다. 그 후엔 반복처리 하면서 소수점의 자릿수만큼 status의 값을 증가시켜준다.

(2) 실수 단순 계산기 프로그램에서 피연산자 숫자를 만드는 역할을 하는 다음 함수 get\_real()을 정의하는 함수코드를 쓰시오

double get\_real(double current\_operand, char c, int status)

{

int part1;

double part2;

if (status == 0) {

part1 = c - '0';

current\_operand \*= 10;

current\_operand += part1;

}

else {

part2 = c - '0';

part2 = part2 \* pow(10, -status);

current\_operand += part2;

}

return current\_operand;

}

[참고1] 정수 단수 계산기 프로그램(int\_cal.c)

#include <stdio.h>

#include <math.h>

#define STAGE1 0

#define STAGE2 1

#define STAGE3 2

void int\_calculator(void);

int int\_operation(char op, int op1, int op2);

int get\_int(int current\_operand, char c);

int is\_operator(char c);

int is\_digit(char c);

main()

{

printf(" 계산기 프로그램을 시작합니다.\n");

printf("정수형의 피연산자와 사칙연산자 중 하나를 입력하시오.\n");

printf("결과를 받은 후 끝내려면 문자 x를 입력하시오\n");

int\_calculator();

printf("\n계산기 프로그램을 종료합니다.\n");

}

void int\_calculator(void)

{

int operand1 = 0, operand2 = 0;

char op, c = '0';

int result;

int current\_stage = STAGE1;

while (c != 'x') {

c = getchar();

if (current\_stage == STAGE1) {

if (is\_digit(c))

operand1 = get\_int(operand1, c);

else if (is\_operator(c)) {

op = c;

current\_stage = STAGE2;

operand2 = 0;

}

}

else if (current\_stage == STAGE2) {

if (is\_digit(c))

operand2 = get\_int(operand2, c);

else if (c == '\n') {

result = int\_operation(op, operand1, operand2);

printf("= %d", result);

current\_stage = STAGE3;

}

}

else if (current\_stage == STAGE3) {

if (c == '\n') {

operand1 = 0;

current\_stage = STAGE1;

}

else if (is\_operator(c)) {

op = c;

current\_stage = STAGE2;

operand1 = result;

operand2 = 0;

}

}

}

}

int int\_operation(char op, int op1, int op2)

{

int result;

switch (op) {

case '+':

result = op1 + op2;

break;

case '-':

result = op1 - op2;

break;

case '\*':

result = op1 \* op2;

break;

case '/':

result = op1 / op2;

break;

}

return result;

}

int get\_int(int current\_operand, char c)

{

int new\_value;

new\_value = c - '0';

current\_operand \*= 10;

current\_operand += new\_value;

return current\_operand;

}

int is\_operator(char c)

{

if (c == '+' || c == '-' || c == '\*' || c == '/')

return 1;

else

return 0;

}

int is\_digit(char c)

{

if (c >= '0' && c <= '9')

return 1;

else

return 0;

}

[참고2] 실수 단수 계산기 프로그램(real\_cal.c)

#include <stdio.h>

#include <math.h>

#define STAGE1 0

#define STAGE2 1

#define STAGE3 2

void real\_calculator(void);

double real\_operation(char op, double operand1, double operand2);

double get\_real(double current\_operand, char c, int status);

int is\_operator(char c);

int is\_digit(char c);

main()

{

printf("계산기 프로그램을 시작합니다.\n");

printf("실수형의 피연산자와 사칙연산자 중 하나를 입력하시오.\n");

real\_calculator();

printf("\n계산기 프로그램을 종료합니다.\n");

}

void real\_calculator(void)

{

double operand1 = 0, operand2 = 0, result;

int op, c = '0';

int current\_stage = STAGE1, status = 0;

while (c != 'x') {

c = getchar();

if (current\_stage == STAGE1) {

if (is\_digit(c)) {

operand1 = get\_real(operand1, c, status);

if (status != 0) status++;

}

else if (is\_operator(c)) {

op = c;

current\_stage = STAGE2;

status = 0;

operand2 = 0;

}

else if (c == '.') status = 1;

} // STAGE1

else if (current\_stage == STAGE2) {

if (is\_digit(c)) {

operand2 = get\_real(operand2, c, status);

if (status != 0) status++;

}

else if (c == '.') status = 1;

else if (c == '\n') {

result = real\_operation(op, operand1, operand2);

printf("= %.3f", result);

current\_stage = STAGE3;

status = 0;

}

} // STAGE2

else if (current\_stage == STAGE3) {

if (c == '\n') {

operand1 = 0;

current\_stage = STAGE1;

}

else if (is\_operator(c)) {

op = c;

current\_stage = STAGE2;

operand1 = result;

operand2 = 0;

}

} // STAGE3

} // while

}

double get\_real(double current\_operand, char c, int status)

{

int part1;

double part2;

if (status == 0) {

part1 = c - '0';

current\_operand \*= 10;

current\_operand += part1;

}

else {

part2 = c - '0';

part2 = part2 \* pow(10, -status);

current\_operand += part2;

}

return current\_operand;

}

int is\_operator(char c)

{

if (c == '+' || c == '-' || c == '\*' || c == '/')

return 1;

else

return 0;

}

int is\_digit(char c)

{

if (c >= '0' && c <= '9')

return 1;

else

return 0;

}

double real\_operation(char op, double operand1, double operand2)

{

double result;

switch (op) {

case '+':

result = operand1 + operand2;

break;

case '-':

result = operand1 - operand2;

break;

case '\*':

result = operand1 \* operand2;

break;

case '/':

result = operand1 / operand2;

break;

}

return result;

}