**Area of Rectangule and Triangle**

**사각형 그리고 삼각형의 넓이를 구하는 코드입니다. Polygon 안에 Figure를 넣은 것이 중요한 포인트 같습니다.**

#include <stdio.h>

typedef enum Figure {

Rectangle, Triangle

}Figure;

typedef struct Polygon{

float width;

float height;

Figure figure;

}Polygon;

void Set(Polygon \*p, float w, float h, Figure f);

float Area (Polygon p);

int main(){

Polygon r,t;

Set(&r,4,5,Rectangle);

Set(&r,4,5,Triangle);

printf("Area of Rectangle is %.0f\n", Area(r));

printf("Area of Triangle is %.0f\n", Area(t));

return 0;

}

void Set(Polygon \*p, float w, float h, Figure f){

\*p=(Polygon){w,h,f};

}

float Area(Polygon p){

switch (p.figure){

case Rectangle: return p.width\*p.height;

case Triangle: return p.width\*p.height/2;

default: printf("Illegal figure\n"); return 0;

}

}

**student information-1**

**c언어에서 class는 사용하지 못하지만 비슷한 기능을 이런식으로 사용할 수 있음을 알 수 있습니다.**

#include <stdio.h>

#include <stdlib.h>

#define SIZE 32

typedef struct Student{

char name[SIZE];

int id;

char nation[SIZE]

}Student;

void Store (Student \*s, char \*name, int id, char \*nation);

void Print(Student s);

int main()

{

Student s;

Store(&s,"Kim",1234,"Korea");

Print(s);

return 0;

}

void Store(Student \*s, char \*name, int id, char \*nation){

strcpy(s->name,name);

s->id=id;

strcpy(s->nation,nation);

}

void Print(Student s){

printf("Student: name = %s; id = %d; nation = %s\n",s.name,s.id,s.nation);

}

**Stack**

**stack을 새로운 방법의 코드로 구현을 할 수 있음을 알게되었습니다.**

#include <stdio.h>

#include <stdlib.h>

#define ERROR(s){fprintf(stderr,s); exit(1);}

#define SIZE 5

typedef struct Stack{

int data[SIZE], top;

}Stack;

void Initialize(Stack\*s);

void Push(Stack \*s, int n;);

int Pop(Stack \*s);

int main() {

Stack s;

Initialize(&s);

Push(&s,10);Push(&s,20);Push(&s,30);

printf("%d ",Pop(&s));

printf("%d ",Pop(&s));

return 0;

}

void Initialize(Stack \*s){

s->top=-1;

}

void Push(Stack \*s, int n){

if (s->top==SIZE-1) ERROR("Stack is full");

s->data[++(s->top)]=n;

}

int Pop(Stack \*s){

if (s->top==-1 ) ERROR("Stac is empty");

return s-> data[s->top--];

}

**Stack(c++)**

**C++로 Stack 코드를 이렇게 구현할 수 있다는 것을 알게 되었습니다. C++의 Stack 코드가 C보다 이해하기 쉬워 보이는 것 같습니다.**

#include <iostream>

#include <cstdlib>

#define ERROR(s){std::cerr<<s;exit(-1);}

#define SIZE 5

class Stack

{

int data[SIZE], top;

public:

Stack();

void Push(int n);

int Pop();

};

int main()

{

Stack s;

s.Push(10);s.Push(20);s.Push(30);

std::cout<<s.Pop()<<" ";

std::cout<<s.Pop();

return 0;

}

Stack::Stack()

{

top = -1;

}

void Stack::Push(int n)

{

if (top==SIZE-1) ERROR("Stack is full");

data[++top]=n;

}

int Stack::Pop()

{

if (top==-1) ERROR("stack is empty");

return data[top--];

}