

# C.M.S 계산기

학번: 2018005

이름: 김민성

Github address: [minsungkim333333 \(github.com\)](https://github.com/minsungkim333333)

## 1. 계산기의 목적

- a. **산업안전보건관리비(안전관리비)**란 건설사업장과 본사 안전전담부서에서 산업재해의 예방을 위하여 법령에 규정된 사항의 이행에 필요한 비용을 말한다.
- b. 안전관리비는 현재 진행중인 진척도에 따른 필수 사용 금액이 정해져 있다. 이 계산기를 통해서 공사에 맞는 안전관리비를 쉽게 계산할 수 있도록 하는 것이 목적이다.

## 2. 계산기의 네이밍의 의미

- a. C.M.S 계산기는 Cost of Money for Safety 의 약자로 안전을 위한 비용을 뜻한다.  
안전관리비는 안전을 위해 지출되는 비용으로 이것을 계산한다는 의미를 담았다.

## 3. 계산기 개발 계획

- a. 산업안전보건관리비를 계산하는 함수를 생성한다. Def cms()
  - i. 함수변수로는 계산에 필요한 변수들을 지정해준다(재료비, 직접노무비, 공사 종류) def cms(Material\_cost, labor\_costs, kind)
  - ii. 계산에 필요한 대상액(재료비+직접노무비)을 지정해준다.  
Dsy= Material\_cost + labor\_costs
  - iii. If 조건문을 활용하여 대상액의 금액, 공사 종류에 따른 계산을 입력한다.
  - iv. Return 을 통해 값을 출력한다.
- b. 공사 진척도에 따른 필수 사용 금액을 계산하는 함수를 생성한다.
  - i. Def cms()에서 구한 값과 진척도를 변수로 지정해준다.
  - ii. If 조건문을 사용하여 진척도에 따른 사용 금액을 계산한다.
  - iii. Return 을 통해 값을 출력한다.
- c. 함수를 2 개 이상 사용하기에 Class 로 함수를 묶어 사용한다.

- i. Class cms\_calculator: 로 설정한다.
- ii. Init 함수를 이용하여 위의 함수들에 필요한 변수들을 지정한다.
- d. While True 반복문을 활용하여 계산을 마친 후 다시 반복되게 설정해준다.
  - i. 위의 계산들에 필요한 변수 4 가지를 a, b, c, d 에 input()을 이용하여 받는다. 이때 정수형이 필요한 것들은 int()를 이용해 변환해준다.
  - ii. 각 함수에 필요한 변수들을 집어넣고 print()를 통해 값을 출력한다.

#### 4. 계산기 개발 과정

##### a. 개발 과정

- i. Class 를 생성하고 init 함수를 통해 변수들을 지정해주었다.

```
class cms_calculator:
    def __init__(self, Material_cost, labor_costs, progress_chart, kind):
        self.Material_cost=Material_cost
        self.labor_costs=labor_costs
        self.progress_chart=progress_chart
        self.kind=kind
        self.money=0
```

- ii. def cms(self) 함수를 생성하고 dsy(대상액)을 계산 한 후 dsy 의 금액별, 공사종류의 조건에 따른 비율을 입력하고 알맞은 계산을 하도록 설정하였다. - (수식이 길어 일부 생략하였습니다.)

```
def cms(self):
    dsy = self.Material_cost+self.labor_costs
    if dsy < 5000000000 and self.kind == '일반건설공사_갑':
        rate=0.0293
        result= dsy*rate
    elif 5000000000 > dsy >= 5000000000 and self.kind == '일반건설공사_갑':
        rate=0.0186
        result= dsy*rate+5349000
    elif 5000000000 >= dsy and self.kind == '일반건설공사_갑':
        rate=0.0197
        result= dsy*rate
    self.money=self.money + result
    return print('안전관리비는' + str(result) + '원 입니다.')
```

앞서 지정해둔 self.money 에 결과값을 더해주고 값을 출력시키도록 하였다.

- iii. Def work\_progress\_chart(self) 함수를 생성하고 진척도를 조건으로 한 계산을 실행하고 결과값을 프린트하게 설정하였다.

```
def work_progress_chart(self):
    if 70 > self.progress_chart >= 50:
        cost = self.money * 0.5
        print('안전관리비를 ' + str(cost) + '원 이상 사용하여야 합니다.')
    elif 90 > self.progress_chart >= 70:
        cost = self.money * 0.7
        print('안전관리비를 ' + str(cost) + '원 이상 사용하여야 합니다.')
    elif self.progress_chart >= 90:
        cost = self.money * 0.9
        print('안전관리비를 ' + str(cost) + '원 이상 사용하여야 합니다.')
    else:
        print('안전관리비 사용 기준금액이 정해져있지 않습니다. 자유롭게 사용 가능합니다.')
    return cost
```

- iv. 함수들을 작성한 후 테스트를 해 보았다. 이 때 오류가 발생하였다.  
 자세한 내용은 아래의 c. 에러 발생 지점에서 자세히 설명하겠습니다.
- v. 에러를 해결하기 위해 함수를 수정해주었다.

```
def cms(Material_cost, labor_costs, kind):
    dsy = Material_cost+labor_costs
    result=0
    if dsy < 5000000000 and kind == '일반건설공사_갑':
        rate=0.0293
        result=dsy*rate
    elif 5000000000 > dsy >= 5000000000 and kind == '일반건설공사_갑':
        rate=0.0186
        result=dsy*rate+5349000
    elif 5000000000 <= dsy and kind == '일반건설공사_갑':
        rate=0.0197
        result=dsy*rate
```

(생략)

```
    else:
        return 0
    return result

def work_progress_chart(progress_chart, money):
    cost=0
    if 70 > progress_chart >= 50:
        cost = money * 0.5
    elif 90 > progress_chart >= 70:
        cost = money * 0.7
    elif progress_chart >= 90:
        cost = money * 0.9
    else:
        return 0
    return cost
```

- vi. 제작 중 cms 에서 앞서 사용한 조건을 제외한 특수조건이 있었기에 cms\_2 함수를 따로 만들어주었다.

```
def cms_2(Material_cost, labor_costs, kind):  
    dsy = Material_cost+labor_costs  
    result=0  
    if kind == '일반건설공사_갑':  
        rate=0.0215  
        result=dsy*rate  
    elif kind == '일반건설공사_을':  
        rate=0.0229  
        result=dsy*rate
```

(생략)

```
else:  
    return 0  
return result
```

- vii. 이후 while true 함수를 통해 값을 입력하도록 설정해주었다.

```
while True:  
    a=int(input('1. 자료비를 입력해주세요. (숫자만 입력해주세요. 단위 : 원)'))  
    b=int(input('2. 직접 노무비를 입력해주세요. (숫자만 입력해주세요. 단위 : 원)'))  
    c=input('3. 다음의 보기 중 공사의 종류를 선택하여 입력해주세요. (일반건설공사_갑, 일반건설공사_을, 중건설공사, 철도개도건설공사, 특수정기타건설공사)  
    d=int(input('4. 공사의 진척도를 입력해주세요. (단위 : %)'))  
    e=input('5. 보전관리자 선임대상 공사인가요? 예/아니요')  
    if e == '예':  
        cost_2=int(cms_2(a,b,c))  
        use_money=int(work_progress_chart(d,cost_2))  
        if cost_2 == 0:  
            print('정보가 알맞지 않습니다. 정보를 다시 입력해주세요.')  
        elif use_money == 0:  
            print('안전관리비는 '+str(cost_2)+'원 입니다.')  
            print('안전관리비 최저 사용금액이 없습니다.')  
        else:  
            print('안전관리비는 '+str(cost_2)+'원 입니다.')  
            print('안전관리비를 ' +str(use_money) +'원 이상 사용하여야 합니다.')  
    elif e == '아니요':  
        cost=int(cms(a,b,c))  
        use_money=int(work_progress_chart(d,cost))  
        if cost == 0:  
            print('정보가 알맞지 않습니다. 정보를 다시 입력해주세요.')  
        elif use_money == 0:  
            print('안전관리비는 '+str(cost)+'원 입니다.')  
            print('안전관리비 최저 사용금액이 없습니다.')  
        else:  
            print('안전관리비는 '+str(cost)+'원 입니다.')  
            print('안전관리비를 ' +str(use_money) +'원 이상 사용하여야 합니다.')  
    else:  
        print('정보가 알맞지 않습니다. 다시 입력해주세요.')
```

## b. 에러 발생 지점

i. 위의 클래스함수를 테스트 한 결과

```
test=cms_calculator(4000000000,500000000,82,'일반건설공사_갑')
test.cms()
test.work_progress_chart()
안전관리비는89049000.0원 입니다.
안전관리비를 0.0원 이상 사용하여야 합니다.
```

ii. 원인 분석

1. Cms 리턴 값은 정상적으로 출력되기에 work\_progress\_chart 함수에서 오류가 발생한 것으로 생각하였다.
2. 코드 자체의 오류는 없었으며 cms 의 리턴 값을 변수로 가져오는 과정에서 오류가 발생한 것으로 판단하였다.

c. 에러 발생에 대한 해결책

- i. \_\_init\_\_이 초기화함수이기 때문에 cms()를 통해 값을 self.money 에 저장된다 하더라도 work\_progress\_chart ()로 다시 호출하는 과정에서 값이 0 으로 초기화된다는 것을 알게 되었다.
- ii. Init 으로 변수를 지정하지 않고, 함수에 직접적으로 변수를 지정하는 방법을 사용하여 에러를 해결하였다. 이 과정에서 class 가 불필요 해졌고 삭제하였다.

d. 함수 설명

```
def cms(Material_cost, labor_costs, kind):
    dsy = Material_cost+labor_costs
    if dsy < 5000000000 and kind == '일반건설공사_갑':
        rate=0.0293
        result=dsy*rate
    elif 5000000000 > dsy >= 5000000000 and kind == '일반건설공사_갑':
        rate=0.0186
        result=dsy*rate+5349000
    elif 5000000000 <= dsy and kind == '일반건설공사_갑':
        rate=0.0197
        result=dsy*rate
```

(생략)

```
else:
    return 0
return result
```

Cms 함수에서 재료비, 직접노무비, 공사종류의 변수를 받고 재료비와, 노무비를 이용하여 dsy 를 계산한다.

두가지 조건이 모두 겹칠 경우에 곱해지는 비율이 다르기 때문에 조건문을 사용하였다. And 를 통해 두가지 조건이 모두 성립할 경우 연산이 되게 하였다. 같은 방법으로 조건들을 생성해주었다.

이후 result 를 리턴 값으로 출력되게 하였다.

- Else: return 0 값을 만들어준 이유는 앞의 조건들에 성립하지 않는 다른 값을 입력했을 경우 리턴 값이 0 이 나온 경우에 재입력 안내 문구를 출력하게 하기 위하여 추가하였다.

```
def work_progress_chart(progress_chart, money):  
    if 70 > progress_chart >= 50:  
        cost = money * 0.5  
    elif 90 > progress_chart >= 70:  
        cost = money * 0.7  
    elif progress_chart >= 90:  
        cost = money * 0.9  
    else:  
        return 0  
    return cost
```

위 함수도 2 가지 조건, 진척도, 안전관리비를 변수로 받고 조건에 맞는 값을 계산하는 함수이다.

Else 함수는 위와 같은 의미로 추가하였다.

```
def cms_2(Material_cost, labor_costs, kind):  
    dsy = Material_cost+labor_costs  
    result=0  
    if kind == '일반건설공사_갑':  
        rate=0.0215  
        result=dsy*rate  
    elif kind == '일반건설공사_을':  
        rate=0.0229  
        result=dsy*rate
```

(생략)

```
else:  
    return 0  
return result
```

Cms\_2 함수는 특정조건에서 발생하도록 따로 분류하여 생성해준 함수이다.

```

while True:
    a=int(input('1. 재료비를 입력해주세요.(숫자만 입력해주세요. 단위 : 원)'))
    b=int(input('2. 직접 노무비를 입력해주세요.(숫자만 입력해주세요. 단위 : 원)'))
    c=input('3. 다음의 보기 중 공사의 종류를 선택하여 입력해주세요.(일반건설공사, 일반건설공사-음, 중건설공사, 철도궤도건설공사, 특수및기타건설공사)')
    d=int(input('4. 공사의 진척도를 입력해주세요. (단위 : %)'))
    e=input('5. 보검관리자 선임대상 공사인가요? 예/아니요')
    if e == '예':
        cost_2=int(cms_2(a,b,c))
        use_money=int(work_progress_chart(d_cost_2))
        if cost_2 == 0:
            print('정보가 알맞지 않습니다. 정보를 다시 입력해주세요.')
        elif use_money == 0:
            print('안전관리비는 '+str(cost_2)+'원 입니다.')
            print('안전관리비 최저 사용금액이 없습니다.')
        else:
            print('안전관리비는 '+str(cost_2)+'원 입니다.')
            print('안전관리비를 ' +str(use_money) + '원 이상 사용하여야 합니다.')
    elif e == '아니요':
        cost=int(cms(a,b,c))
        use_money=int(work_progress_chart(d_cost))
        if cost == 0:
            print('정보가 알맞지 않습니다. 정보를 다시 입력해주세요.')
        elif use_money == 0:
            print('안전관리비는 '+str(cost)+'원 입니다.')
            print('안전관리비 최저 사용금액이 없습니다.')
        else:
            print('안전관리비는 '+str(cost)+'원 입니다.')
            print('안전관리비를 ' +str(use_money) + '원 이상 사용하여야 합니다.')
    else:
        print('정보가 알맞지 않습니다. 다시 입력해주세요.')

```

While true 로 계속해서 반복되는 반복문을 생성해주었다.

A,b,c,d,e 값에 필요한 변수들을 지정 받을 수 있도록 문구를 작성하였다.

이때 변수가 정수형이 필요한 변수들에는 int()를 이용하여 전환해주었다.

E 의 변수를 할당 받을 때 앞서서 특정조건에 성립할 때 다른 함수를 사용할 수 있도록 (예/아니요)의 값을 받고,

E 의 값에 따라 알맞은 cms, cms\_2 를 실행하도록 조건문을 입력해주었다.

이후 앞서 잘못된 값을 입력 받았을 경우 0 을 출력하게 하였는데 이 점을 조건문으로 하여 값이 0 인 경우 값을 다시 입력하도록 요구하는 문구가 출력되도록 하였고 값이 제대로 입력되었다면 안전관리비와 최저사용금액을 출력되도록 함수를 작성하였다.

## 5. 계산기 개발 후기

계산기를 제작하는 과정에서 사전 준비가 중요하다는 점을 느끼게 되었습니다. Init 을 사용했을 때 오류가 발생하였고 이를 수정하는 과정이 굉장히 힘들었기에 사전에 미리 준비하는 것이 중요하다고 느꼈습니다.

간단한 연산임에도 제작이 굉장히 힘들다고 느꼈고, 프로그래밍을 제작할 때 끈기의 중요성을 알게 되었습니다.

코드가 길어질수록 오류를 식별, 수정하기 힘들다는 것을 느꼈습니다. 그래서 중간중간 테스트를 하며 차근차근 하는 것이 중요하다고 생각하였고, 기록의 중요성을 느끼게 되었습니다.