

도라에몽 계산기

학번: 2318095

이름: 조상준

Github address: https://github.com/chosangjun1/2318095_quiz.git

1. 계산기의 목적

- a. 실제 안전관리자가 업무 중 작성하거나, 사업장의 규모에 따라서 알고 있어야 하는 계산들이 있다. 이러한 계산들의 종류가 매우 다양하기 때문에 이를 하나의 집약적인 안전계산기로 모두 계산 가능 하다면 업무의 효율이 상당히 증가 할 수 있다고 판단되어, 종합적인 계산이 가능한 안전계산기를 구상해 보았다. 또한 사업장의 안전경영 KOSHA-ms 에서 가장 흔하고 가장 반영하기 쉬운 무재해 운동을 동시에 계산기에 나타내거나, 장애 등급에 따른 근로 손실일수를 즉각적으로 구할 수 있도록 하여, 단순 계산기를 넘어서 안전관리를 효율적이고, 보다 정확하게 시행 가능한 도구를 목적으로 하였다.
- b. 계산기 활용 대상: 사업장 규모에 상관없이 사무업을 동반한 (현장+사무)관리자 모두를 대상으로 하였다. 특히 건설과, 생산관련 안전 관리자의 업무와 직접적으로 관련 있는 기능들이 많이 있다.

2. 계산기의 네이밍의 의미

- a. 도라에몽 계산기는 만화 도라에몽에 나오는 도라에몽이 친구의 부탁을 들어주고 생활을 좋게 만들어 주려는 것처럼, 이 계산기 또한 안전관리자에게 있어서 도라에몽 같이 여러 도움을 줄 수 있다, 라는 의미에서 “도라에몽” 이라고 작명했다.+(내가 도라에몽을 좋아한다.)

3. 계산기 개발 계획

- a. 입력 변수는 무엇
도라에몽의 입력 변수는 크게 3 가지로 나눌 수 있다.
 1. 첫 입력변수는 어떤 계산을 할 것인지이다.
 2. 두 번째 입력변수는 해당 계산에서 요구하는 다수의 정보 값이다.

사망자수, 재해자수, 사업장크기 등등 계산에 필요한 기초적인 정보를 요구한다.

3. 무재해 운동 n 일차에서 n 의 값을 입력 받는다.

b. 개발한 함수는 무엇을 연산하는 함수인 지

도라에몽에는, 실제 업무에서 유용하게 쓰일 수 있도록 산업안전기사에서 안전관리 기사의 업무 중 필요한 계산식들을 대부분 넣어 두었다.

1. 연천인율
2. 강도율
3. 도수율
4. FIS(종합재해지수)
5. 재해율
6. 사망인율
7. 재해손실일수
8. 환산도수율
9. 환산강도율
10. 평균강도율
11. 안전관리비
12. 안전하중
13. 종합재해위험지수
14. 무재해운동 n 일차

위 14 가지 결과들을 계산하는 함수들이다.

c. 연산 과정은 어떻게?, 조건문은 왜 필요하며, 왜 이렇게 설계했는지 등

모든 연산들은 현장에서 값들을 구한 뒤 넣기만 하면 결과가 나오도록 설계 하였다. 실제 안전 관리자들의 업무는 현장과 사무를 둘 다 왕래하는 경우가 보통이므로, 기존에 큰 수 들을 적어두고 나중에 한번에 계산하는 방식이 아닌, 그 자리에서 값을 바로 입력 후 원하는 값을 바로 받고, 그것을 기록하는 효율적인 방식을 목표로 모든 코드를 설계 하였다. 코드들은 첫번째 입력 함수에서 자신의 코드를 호출 받으면 바로 그 코드로 넘어갈 수 있도록 if , elif, else 부분을 사용 하였다. 7,12,14 부분을 제외하면 나머지

코드들은 일정부분 정형화된 코드로 실행 가능 했기에 가장 최근에 배운 def 함수 사용법을 살짝 응용해서 기본 틀로 만들어 코드를 만들었다. 7 번의 경우, 산업안전보건법에서 해당 사항을 규정으로써 다루고 있기 때문에, 해당 규정에 적합한 코드를 만들기 위해 수업 시간중 “성적산출 시스템” 부분을 응용하여 만들어 보았다. 이는 장애 등급에 따른 근로 손실일수를 바로 확인 가능하기 때문에 실제 업무에서도 매우 유용하게 사용 될 것이라 판단된다. 마지막 14 번은 무재해운동 n 일차 계산인데, 처음에는 사용자가 무재해운동 n 일차를 입력하면 for 반복문을 사용하여 0 일부터 해당 일수만큼 나타내는 코드를 생각 해였지만, 이런 식이면 무재해운동의 날짜가 조작가능 한 부분에서 기존 무재해운동과 성질이 다르다고 판단되어, 자동으로 일수가 넘어 갈 수 있도록 while 반복문 함수와 time.sleep 일시정지함수를 사용하여 일정 시간마다 1 일씩 증가하는 코드를 만들어 보았다.

4. 계산기 개발 과정

a. 계획 후 실제 개발 과정을 기록

첫번째로 짠 코드는 기본적인 반복적으로 사용할 구조와, 기본적인 틀을 구상해 보았다. 결과적으로 연천인율,강도율로 def 의 기본적인 틀을 만들고 for 문을 사용하여 무재해운동 n 일차 초기 코드를 만들었다.

그 뒤 기본 def 코드를 가지고 나머지 강도율, 환산도수율, 사망인율, FIS 등 나머지 코드를 비슷하게 코드를 만들었다. 이쯤 까지 만들고서 도라에몽이라는 이름에 걸맞게 더 다양한 도구를 넣기로 결정했다. 장애등급에 따른 근로손실일수를 표시 할 수 있도록 어떤 코드를 사용할지 고민 하던 중 수업 중에 실습 문제였던, “학점산출기” 코드가 적합하다고 생각하여, 응용해서 만들었다. 마지막으로 현재 안전에서 각광받는 안전보건 경영(kosha-ms)시스템의 내용을 넣고 싶어서 time.sleep 을 이용하여 무재해 운동을 나타내었다.

b. 각 함수는 어떻게 동작하는 지 구체적으로 설명

1. 연천인율, 도수율, 강도율, FIS, 재해율, 환산강도율, 환산도수율,평균강도율,안전관리비, 사망인율, 종합재해위험지수는

첫 input()에서 자신의 계산이름이 호출되면 if 함수로 해당 계산 식으로 이동된다. 그 후 함수(def)로 해당 연산을 구축하고, 해당 식이 진행되는 것을 print 로 보여준다. 마지막으로 결과를 얻기 위해 해당 식에 필요한 현장정보를 input()을 통하여 입력 받으면 해당 식의 결과를 도출하며, 해당 식이 의미하는 부분을 같이 출력한다.
위에 언급한 계산들은 이러한 방식으로 간단히 작동시킬 수 있다.

2. 재해 손실 일수의 경우 시작은 다른 코드 들과 똑같이 처음 input()에서 자신의 이름을 호출 받으면 if 부분이 호출 받으며 해당 코드로 이동한다. 그 후 사용자에게 장애등급(1~14)을 입력 받아, 해당 장애 등급에 따른 근로 손실 일수를 출력한다. 만약 범위 내에 해당 장애 등급이 없을 경우 None 을 반환한다. 모든 등급에는 근로 손실일수를 몇일 로 return 할지 def 와 if 부분을 사용하여 지정해 두었고, 마지막으로 해당 calculate_loss_days 함수를 호출하여 입력 받은 장애 등급에 따른 재해 손실 일수를 계산한다. None 이 반환되면 잘못된 등급이 입력되었음을 알리는 메시지를 else 를 활용하여 출력되도록 하였다
3. 안전하중계산의 경우 입력한 재료의 항복강도, 안전계수, 단면적을 기반으로 안전 하중을 계산하도록 만든 프로그램이다. 다른 코드와 마찬가지로 처음 입력 값에서 안전하중의 이름을 호출 받으면 코드를 실행한다. 안전하중의 매개변수는 항복강도, 안전계수, 단면적 3 가지이다. 이 3 가지로 허용응력과 안전 하중을 계산하여 결과를 도출한다. 이 3 가지 정보를 input()을 사용하여 사용자에게 입력을 받고, 변수에 저장한다. 이를 float()를 사용하여 실수로 변환한 뒤 함수를 호출하여 입력 값을 바탕으로 안전하중을 계산한다. 계산된 하중을 safe_load 에 저장하고 안전 하중 특성상 유한소수가 나올 가능성이 크기 때문에 현재 현장에서도 표기하는 방식대로 소수점 둘째 자리까지 f"{safe_load:.2f}" 로 출력한다.
4. 마지막으로 무재해운동 코드는 다른 함수들과 다르게 호출 받고 실행하는 코드가 아닌 매번 실행되는 코드이다. 무재해운동을 추적하는 프로그램으로, 프로그램이 시작되면 변수를 0 부터 24 시간(코드에는

임의로 설정한 시간)까지 기다리면서 해당 시간이 지나면 일수를 출력하는 방식이다. Def track_days(): 로 시작하여 day_count 를 사용하여 변수를 0 으로 초기화 한다. 이 변수가 무재해운동의 일수를 저장하는 역할을 한다. 그 다음은 while 을 사용하여 무한 루프를 만드는데 이 루프를 계속 기억시키고 진행하게 만들고 싶지만 아직 모르는 부분이 너무 많고, 어려워서 한번에 계산에서 다음 계산으로 넘어가면 끊기는 부분이 한계이다. 마지막으로 track_days 를 호출하여 프로그램을 시작한다.

c. 에러 발생 지점

def 에 변수를 저장하고 간단한 계산식을 만들고 호출하는 대부분의 경우는 어려울 예러나 어려울 부분이 없었다. 또한 재해손실 일수와, 안전 하중의 경우에도 수업시간에 배운 내용을 약간 응용한 문제라 어려울 부분이 없었다. 하지만 무재해 운동 프로그램을 짜는 중 어떤 코드를 써야 일정 시간마다 무한 반복인 코드를 만들 수 있을지 몰랐다. For, while 반복문을 사용하여 무재해 일수를 나타내는 부분을 쉽게 캐치 했지만, 일정 시간마다 일수를 늘리는 방법을 모색하던 중 times.sleep 이라는 기능을 인터넷에서 발견하여 공부하고 사용해 보았다. 생각보다 할만한 난이도의 간단한 코드여서 어려움 없이 해결하였다.

d. 에러 발생에 대한 해결책

에러가 발생하여 힘든 부분은 없었지만, 프로그램이 끝나도 무재해일수를 계속 진행시키고 저장한 뒤 다시 프로그램이 시작될 때 저장된 값을 이어받아 계속 실행하는 방법을 계속 모색하는 중이다. 인터넷 등에 찾아본 결과 파일을 사용하는 방법이 가장 잘 어울린다고 판단되지만, 파일을 사용하여 만드는 코드는 아직 내 수준에서는 너무 어렵고 구상한대로 만들지도 못하는 수준이라 생각되어, 쉽게 만들지 못하고 있다. 파일을 만들고 open(파일이름, "일수") as file: 로 파일에 일수를 저장하고 file.write() 을 사용하는 것 까지는 알 것 같았지만 좀처럼 생각한대로 결과와 코드가 나오지 못했다.

e. 해결책 적용 시 어떻게 변화

f. 동작 결과 캡처

```
print("원재료비/안전보유비 = 300,000원/1000원 = 300")
GFr = input("원재료비율 입력하시오:")
GFr = input("안전보유비율 입력하시오:")
result = wt_calculate(GFr,GFr)
print("원재료비율 : ", result, ", 안전보유비 : ")

#안전관리비

def wt_calculate == "안전관리비"
    language = "python"
    def wt_calculate == "안전관리비"
        return total_cost * (safety_percent/100)
    tot = float(input("원재료비율 총비율 입력하시오:"))
    app = float(input("안전보유비율 입력하시오:"))
    result = calculate_safety_management_cost(tot,app)
    print("안전관리비 : ",result,"원 입니다.")

원재료비
calculate == "원재료비" else
안전관리비계산기
1
```

```
print("GFR =", GFR, "ml/min/1.73m2", sep="")
GFR = input("환자 GFR 값을 입력하십시오 : ")
GFR = input("GFR 값을 입력하십시오 : ")
result = wt.calculate_GFR(GFR, GFR)
print("환자 GFR은", result, "입니다.")

# 안전관리 계산

elif wt.calculate == "안전관리계산":
    input("계산 : ")
    def calculate_safety_management_cost(total_cost, safety_percent):
        return total_cost * (safety_percent/100)
    tot = float(input("안전관리계산 비용 값을 입력하십시오 : "))
    spp = float(input("안전관리계산 비용 안전비율 값을 입력하십시오 : "))
    result = calculate_safety_management_cost(tot, spp)
    print("안전관리계산은", result, "를 합니다.")

# 안전관리 계산

wt.calculate == "안전관리계산" : else
    print("안전관리계산이 없습니다.")

# 안전관리계산기

환자 GFR 계산 결과에 따라, 환자를 관리할 수
있을 정도로 입력하세요 (1~14) : 6
환자 몸무게 6kg 미만 환자 및 몸무게 3000g 이하,
환자에 관한 항목을 입력하십시오.
환자 몸무게 300g 이하입니다.
환자 몸무게 300g 이하입니다.
환자 몸무게 300g 이하입니다.
환자 몸무게 300g 이하입니다.
환자 몸무게 300g 이하입니다.
환자 몸무게 300g 이하입니다.
```

6

```

71     print("GPR:강제감소/교체비용 x 100,000\n" + "도수율 x 1/10")
72     GPr = input("강제감소를 입력하십시오:")
73     GPr = input("교체감소율을 입력하십시오:")
74     result = wt.calculate_GPr(GPr, GPr)
75     print("강산 도수율은", result, "입니다.")
76
77 #안전관리비
78
79 elif wt.calculate == "안전관리비":
80     usage = None
81     def calculate_safety_management_cost(total_cost, safety_percent):
82         return total_cost * (safety_percent/100)
83     print("프로그램 중 비용은 입력하십시오:")
84     toC = float(input("프로그램 중 비용을 입력하십시오:"))
85     syp = float(input("안전관리비 비율을 입력하십시오:(%)"))
86     result = calculate_safety_management_cost(toC, syp)
87     print("안전관리비는", result, "을 합니다.")
88
89 #안전관리도율
90 def wt.calculate == "안전관리도율":
91     pass

```

5.계산기 개발 후기

g. 계산기 개발 후 느낀 점 설명

처음엔 여러가지 계산기를 구상해 보다가, 내가 안전관리자가 되었을 때 어떤 계산기가 있으면 편하고, 효율적으로 업무를 처리할 수 있을까 하는 생각을 하다가 해당 계산기를 만들게 되었는데, 부족한 지식은 찾아보고 채워가고, 다양한 방면에서 조언을 구하는 활동을 해보니 조금은 안전과 프로그래밍 두 분야에 자신감이 붙은 느낌이다. 코드를 만들고 있으면 생각보다 재미있어서 시간이 금방 지나갔고, 안전에 관한 지식이 부족할 때는 스트레스도 받았지만 결과적으로 미래에 사용할 수 있는 나의 지식의 일부가 되었다고 생각하니 그 또한 좋은 기억으로 남았다. 이번 과제는 나에게 매우 인상깊고 즐거운 시간이었다.

Ps. 실습과제 쉽게 내주세요 교수님 π