빠르고 정확하고 알기 쉬운 몰농도 희석도구

학번 : 2018016

이름 : 류경용

Github address:
https://github.com/2018016-Kyungyong-Ryu/easy_dilution_cal.git

1. 계산기의 목적

a. 사업장에서 화학물질을 물이나 용매에 녹인 수용액을 자주 사용하곤 한다. 이때 물에 녹인 수용액을 인체에 유해하지 않을 수준까지 낮추거나, 제조업에서 필요한 농도까지 희석을 해야한다. 이러한 용액을 희석할때 사용하는 계산식을 이용해 계산기를 만들어서 인체에 노출되거나 외부로 누출되었을 때 신속하게 계산을 하여 위험하지 않은 농도까지 낮추어 2차 피해가 커지는것을 막기위해 해당 계산기를 제작하게 되었다.

b. 계산기 활용 대상 : 화학물질을 용매에 녹여 용액을 만들어 사용하거나, 인체에 유독한 액체상의 위험물질을 다루는 사업장에서 응급조치를 하거나, 특정 농도의 용액을 만들 때 사용할 수 있다.

2. 계산기의 네이밍의 의미

a. 이 계산기는 간단한 변수만 입력한다면, 계산에 필요한 모든 조건들과 내용들을 알기 쉽게 한번에 출력하여 보여주는 계산기 이므로 빠르다는점, 정확하다는점, 그리고 보기 쉽다는점을 강조한 제목으로 하였다.

3. 계산기 개발 계획

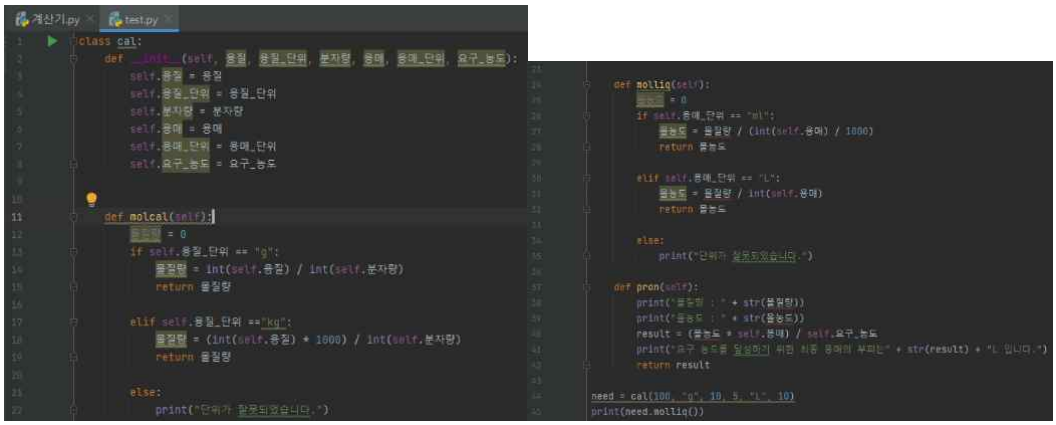
a. input을 통해 1. 용질의 양, 2. 용질의 단위, 3. 용질의 분자량, 4. 용매의 부피, 5. 용매의 부피의 단위, 6. 희석을 통해 얻고자 하는 목표 용액의 농도까지 총 6개의 변수를 입력해야하고, 계산을 위해 내부적으로 사용하는 변수는 1. 용질의 몰질량, 2. 용액의 몰농도 로 총 2가지가 있다.

b. 개발한 함수는 우선 용질의 단위를 확인해 용질의 질량을 g으로 통일한 뒤, 분자량으로 나눠 몰질량을 구하고, 용매의 부피의 단위를 확인해 L로 통일한 뒤 몰질량에 부피를 나눠 몰농도를 구한 뒤 (몰농도 * 용매의 부피 = 희석한

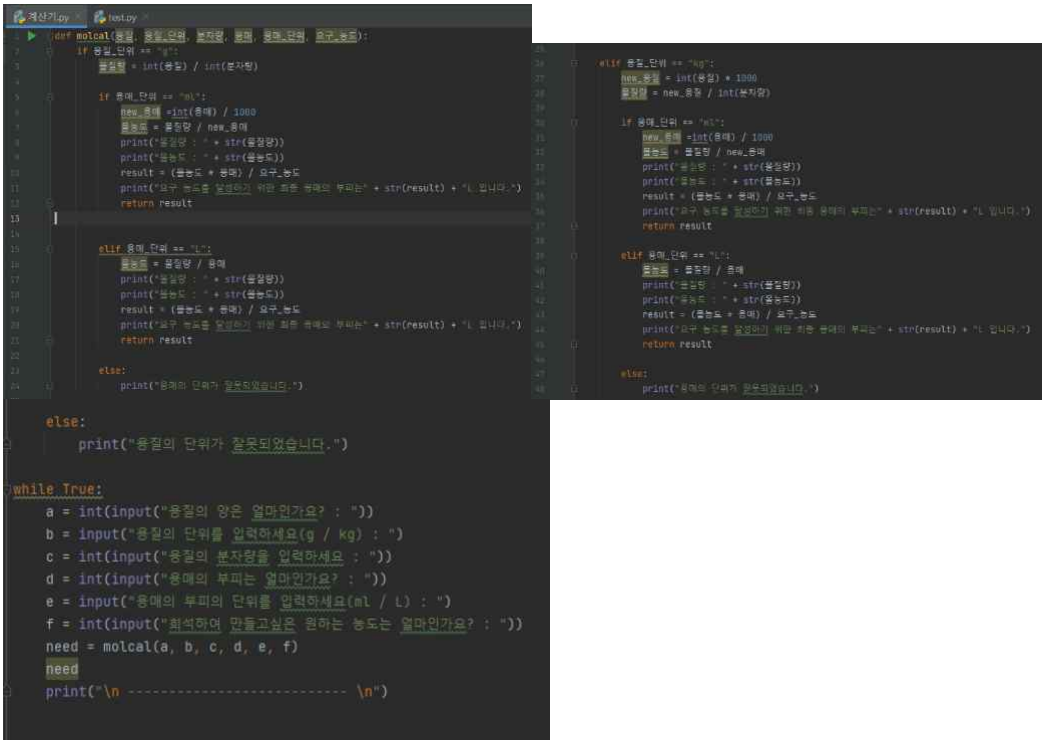
몰농도 * 최종 용매의 부피) 공식을 사용해 최종 용매의 부피를 구하는 함수이다

c. 우선 if문을 사용해 용질의 질량단위가 “g”인지를 판단하고, 만약 “g”이 아닌 “kg”이라면 1000을 곱해 단위를 “g”으로 맞춰준다. 만약 용질의 단위가 “g”,”kg” 둘 다 아니라면 print()를 통해 잘못된 단위가 입력되었음을 알린다. 용질의 단위를 판단했다면, 이중조건문을 통해 이번엔 용매의 부피의 단위를 판단한다. 만약 단위가 “ml”라면, 1000을 나눠 “L”로 바꿔서 계산식을 작동하게 된다. 이를 더욱 자세히 표현하게 된다면.
만약 질량 단위가 “g”이 맞다면, 이중조건문을 통해 용매의 부피가 “L”인지 “ml”인지를 판단한다. 만약 질량단위가 “g”이고, 용매의 부피가 “L”라면 바로 계산식을 통해 몰질량과 최종 용매의 부피를 구하는 공식을 사용해 결과값을 출력한다. 만약 질량단위가 다르다면, 조건문을 통해 “g” 과 “L”로 단위를 맞춘 뒤 계산식을 사용해 결과값을 출력한다.

4. 계산기 개발 과정

a. 

처음에는 몰질량을 계산하는 함수, 몰농도를 계산하는 함수, 희석 용액을 계산하는 함수를 각각 따로따로 만들어, 클래스를 사용해 여러 함수들을 묶었다. 하지만 클래스 안의 함수는 __init__을 통해 지정한 변수에 계산 결과값을 저장하고, 이를 다른 함수를 작동시킬때 받아 사용할 수 없다는것을 깨닫고, 모든 계산들이 하나의 함수에서 작동하도록 합쳤다.



```
계산기.py × test.py ×
def molcal(용질, 용질_단위, 분자량, 용매, 용매_단위, 요구_농도):
    if 용질_단위 == "g":
        몰질량 = int(용질) / int(분자량)

        if 용매_단위 == "ml":
            new_용매 =int(용매) / 1000
            몰농도 = 몰질량 / new_용매
            print("몰질량 : " + str(몰질량))
            print("몰농도 : " + str(몰농도))
            result = (몰농도 * 용매) / 요구_농도
            print("요구 농도를 달성하기 위한 최종 용액의 부피는" + str(result) + "L 입니다.")
            return result

        elif 용매_단위 == "L":
            몰농도 = 몰질량 / 용매
            print("몰질량 : " + str(몰질량))
            print("몰농도 : " + str(몰농도))
            result = (몰농도 * 용매) / 요구_농도
            print("요구 농도를 달성하기 위한 최종 용액의 부피는" + str(result) + "L 입니다.")
            return result

        else:
            print("용매의 단위가 잘못되었습니다.")

    elif 용질_단위 == "kg":
        new_용질 = int(용질) * 1000
        몰질량 = new_용질 / int(분자량)

        if 용매_단위 == "ml":
            new_용매 =int(용매) / 1000
            몰농도 = 몰질량 / new_용매
            print("몰질량 : " + str(몰질량))
            print("몰농도 : " + str(몰농도))
            result = (몰농도 * 용매) / 요구_농도
            print("요구 농도를 달성하기 위한 최종 용액의 부피는" + str(result) + "L 입니다.")
            return result

        elif 용매_단위 == "L":
            몰농도 = 몰질량 / 용매
            print("몰질량 : " + str(몰질량))
            print("몰농도 : " + str(몰농도))
            result = (몰농도 * 용매) / 요구_농도
            print("요구 농도를 달성하기 위한 최종 용액의 부피는" + str(result) + "L 입니다.")
            return result

        else:
            print("용매의 단위가 잘못되었습니다.")

    else:
        print("용질의 단위가 잘못되었습니다.")

while True:
    a = int(input("용질의 양은 얼마인가요? : "))
    b = input("용질의 단위를 입력하세요(g / kg) : ")
    c = int(input("용질의 분자량을 입력하세요 : "))
    d = int(input("용매의 부피는 얼마인가요? : "))
    e = input("용매의 부피의 단위를 입력하세요(ml / L) : ")
    f = int(input("희석하여 만들고싶은 원하는 농도는 얼마인가요? : "))
    need = molcal(a, b, c, d, e, f)
    need
    print("\n -------------------------- \n")
```

b. 하나의 함수를 통해 입력받은 단위 변수를 판단하고, 이러한 단위들을 계산을 통해 “g” 과 “L”로 통일해 계산을 해 몰질량과 몰 농도를 구하고, return값을 통해 함수를 출력시 몰질량과 몰농도를 표시하도록 했다. 또한 계산한 몰질량과 몰농도를 통해 원하는 농도로 희석하기 위해서 필요한 용액의 최종 부피를 계산하고, 이를 프린트 하여 최종 결과를 제시하였다. 그런 뒤 변수를 받는 input 함수를 while 반복문을 통해 지속적으로 물어보게 하고, 계산이 끝난 뒤 한줄씩 띄어 위의 계산 결과와 아래 계산 결과가 섞이지 않도록 해 가독성을 높였다.

c. class로 함수를 제작할 당시, molcal()함수에서 계산하여 저장한 몰질량 변수가, 다른 함수인 molliq()를 계산할때 받아서 사용하지 못하는 오류가 지속적으로 발생하였다. 또한 molcal()함수에서 지정한 변수를 molliq() 함수에서 사용하려고 하면, 변수가 지정되지 않음 오류가 발생하였다.

d. 이러한 에러는 class의 변수의 특징에 나타난 오류로 __init__을 통해 변수를 지정했기때문에, 다른 함수에서 변수를 받아쓸땐 __init__에서 지정한 초기값으로 변수가 계속 초기화 되는것이 오류의 원인으로 생각하였다.

따라서 변수에 저장된 점을 지속적으로 유지하고, 계산에 사용하기 위해서는 하나의 함수로 만드는것이 답일거라고 생각하고, 이중 조건문을 통해 함수들을 하나로 합쳤다.

e. 해결책을 적용하였더니 원하는 계산 결과를 얻기 위해서 각 함수를 순차적으로 실행해야 했던 전의 코딩에 비해, 한번의 함수를 입력하면 계산을 통해 나온 결과값을 한번에 보여주어 가독성이 더욱 좋아졌다고 생각한다.

f.

```
용질의 양은 얼마인가요? : 100
용질의 단위를 입력하세요(g / kg) : g
용질의 분자량을 입력하세요 : 10
용매의 부피는 얼마인가요? : 5
용매의 부피의 단위를 입력하세요(ml / L) : L
희석하여 만들고싶은 원하는 농도는 얼마인가요? : 1
몰질량 : 10.0
몰농도 : 2.0
요구 농도를 달성하기 위한 최종 용매의 부피는10.0L 입니다.

 ---------------------------
```

5. 계산기 개발 후기

a. 계산기 개발을 시작할때까지만 해도 class의 특징에 대해 정확히 알지 못했다. 하지만 개발을 하며 class의 사용 목적을 더욱 정확히 알게 되었고, 함수와 class의 정확한 기능을 더욱 잘 알 수 있었다. 또한 가독성을 높이기 위한 고민을 하며 다른 코드여도 같은 결과가 나올 수 있다는점을 직접 만들어보며 느껴보니 파이썬의 특징들이 더 잘 와닫는 느낌이 들었다.