

## 컴퓨터 프로그래밍

- 안전공학용 계산기 HazMatCalc 개발 명세서



소 속 : 국립한국교통대학교

학 부 : 안전공학전공

성 명 : 임재호

학 번 : 2318041

제 출 일 : 2024.05.15.

## 1. 계산기의 목적

본 계산기는 안전관리 컨설팅 기관에서 사용할 수 있도록 화학물질의 위험성을 평가하고, 안전 사고 가능성을 계산하는 기능을 제공하는 것이다. 이를 통해 엔지니어들이 더 효율적이고 정확하게 작업할 수 있도록 돕는다. 주된 대상은 화학물질을 다루는 사업장부터 학교, 공공기관 등 폭 넓은 분야를 포함한다. 특히, 사고 발생 시 큰 피해를 초래할 수 있는 화학사고 예방에 중점을 두었다.

## 2. 계산기 개발 계획

### 2.1 입력 변수

Gaussian Plume Model: 방출율( $Q$ ), 풍속( $u$ ), 확산 계수  $y(\sigma_y)$ ,  $z(\sigma_z)$ , 거리( $y$ ), 높이( $z$ ), 방출 높이( $H$ )

TNT 등가량 계산: 물질의 양( $W_{\text{substance}}$ ), 물질의 에너지( $H_{\text{substance}}$ ), TNT의 에너지( $H_{\text{TNT}}$ )

BLEVE 에너지 계산: 효율( $\eta$ ), 액체 질량( $m$ ), 증발 잠열( $\Delta H$ )

LFL 계산: 하한 폭발 한계(LEL), 부피 백분율(Vol%)

제트 화염 길이 계산: 상수( $C$ ), 질량 유속( $m$ ), 중력 가속도( $g$ ), 배출 구경( $d$ ), 압력( $p$ ), 연소 열( $\Delta H$ )

### 2.2 연산 과정

- ① 초기 입력 받기
- ② 조건문을 통해 입력 값의 유효성 검사
- ③ 연산 수행
- ④ 반복문을 통해 잘못된 입력 값 재입력 요청
- ⑤ 결과 반환 및 출력

## 3. 개발 과정 및 후기

### 3.1 에러 발생 지점 및 해결 방법

에러 발생 지점: 입력 값이 음수이거나, 0인 경우 (코드에서 다루는 5가지 함수는 모두 음수

값으로 입력되어서는 안된다. 각 함수에서는 물질의 양, 풍속, 거리, 높이 등과 같은 물리적인 값들을 다루는데, 이러한 값들은 음수가 될 수 없다.)

해결 방법: 조건문을 통해 잘못된 입력 값이 들어왔을 때 경고 메시지를 출력하고 재입력 요청 (각 함수의 매개변수에 대해 입력 값이 음수인지 여부를 검사하고, 음수일 경우 예외 처리를 한다. 이를 통해 사용자가 음수 값을 입력할 경우 프로그램이 중단되고 예외 메시지가 출력되어, 올바른 양수 값을 입력하도록 유도한다. 이를 위해 각 함수의 시작 부분에 입력 값이 음수인지 확인하는 조건문을 추가했다.)

변화한 내용: 입력 값 유효성 검사를 추가하여 사용자에게 정확한 값을 입력하도록 유도

#### 4. 개발 과정 캡처

##### ① 원추형 확산 모델 (Gaussian Plume Model)

```
1 import math
2
3 1개의 사용 위치 신규 *
4 def gaussian_plume(Q, u, sigma_y, sigma_z, y, z, H):
5     if Q <= 0 or u <= 0 or sigma_y <= 0 or sigma_z <= 0 or y <= 0 or z <= 0 or H <= 0:
6         raise ValueError("입력 값은 양수여야 합니다.")
7
8     C = Q / (2 * math.pi * u * sigma_y * sigma_z) * math.exp(-y ** 2 / (2 * sigma_y ** 2)) * math.exp(
9         -(z - H) ** 2 / (2 * sigma_z ** 2))
10    return C
```

##### ② TNT 등가량 계산

```
11 1개의 사용 위치 신규 *
12 def tnt_equivalent(W_substance, H_substance, H_TNT):
13     if W_substance <= 0 or H_substance <= 0 or H_TNT <= 0:
14         raise ValueError("입력 값은 양수여야 합니다.")
15
16     W_TNT = W_substance * (H_substance / H_TNT)
17     return W_TNT
```

### ③ BLEVE 에너지 계산

```
1개의 사용 위치 신규 *
18 def bleve_energy(eta, m, delta_H):
19     if eta <= 0 or m <= 0 or delta_H <= 0:
20         raise ValueError("입력 값은 양수여야 합니다.")
21
22     E = eta * m * delta_H
23     return E
```

### ④ 하한 폭발 한계 (LFL) 계산

```
1개의 사용 위치 신규 *
25 def lfl_calculation(lel, vol_percent):
26     if lel <= 0 or vol_percent <= 0:
27         raise ValueError("입력 값은 양수여야 합니다.")
28
29     LFL = (lel / 100) * vol_percent
30     return LFL
```

### ⑤ 제트 화염 길이 계산

```
1개의 사용 위치 신규 *
32 def jet_flame_length(C, m, g, d, p, delta_H):
33     if C <= 0 or m <= 0 or g <= 0 or d <= 0 or p <= 0 or delta_H <= 0:
34         raise ValueError("입력 값은 양수여야 합니다.")
35
36     L = C * ((m ** 2 * g * d ** 2) / (p * delta_H)) ** (1 / 3)
37     return L
```

### ⑥ 계산기의 옵션 표시

```
1개의 사용 위치 신규 *
39 def main():
40     print("안전공학용 계산기")
41     print("1. 원추형 확산 모델 (Gaussian Plume Model)")
42     print("2. TNT 등가량 계산")
43     print("3. BLEVE 에너지 계산")
44     print("4. 하한 폭발 한계 (LFL) 계산")
45     print("5. 제트 화염 길이 계산")
46
47     choice = int(input("원하는 계산을 선택하세요 (1-5): "))
48
```

⑦ 선택한 옵션에 따라 해당하는 함수를 호출하기 위해 입력 값을 받고, 입력 값의 유효성을  
검사하고 예외를 처리

```

49     if choice == 1:
50         Q = float(input("방출율 (Q) [kg/s]: "))
51         u = float(input("풍속 (u) [m/s]: "))
52         sigma_y = float(input("확산 계수 y ( $\sigma_y$ ) [m]: "))
53         sigma_z = float(input("확산 계수 z ( $\sigma_z$ ) [m]: "))
54         y = float(input("거리 y [m]: "))
55         z = float(input("높이 z [m]: "))
56         H = float(input("방출 높이 (H) [m]: "))
57
58         try:
59             print("원추형 확산 모델 결과 (농도) [kg/m^3]:", gaussian_plume(Q, u, sigma_y, sigma_z, y, z, H))
60         except ValueError as e:
61             print(e)
62
63     elif choice == 2:
64         W_substance = float(input("물질의 양 (W) [kg]: "))
65         H_substance = float(input("물질의 에너지 (H) [J/kg]: "))
66         H_TNT = float(input("TNT의 에너지 (H_TNT) [J/kg]: "))
67

```

```

68         try:
69             print("TNT 등가량 계산 결과 (TNT 등가량) [kg]:", tnt_equivalent(W_substance, H_substance, H_TNT))
70         except ValueError as e:
71             print(e)
72
73     elif choice == 3:
74         eta = float(input("효율 (n) [비율, 예: 0.7]: "))
75         m = float(input("액체 질량 (m) [kg]: "))
76         delta_H = float(input("증발 잠열 ( $\Delta H$ ) [J/kg]: "))
77
78         try:
79             print("BLEVE 에너지 계산 결과 (에너지) [J]:", bleve_energy(eta, m, delta_H))
80         except ValueError as e:
81             print(e)
82
83     elif choice == 4:
84         lel = float(input("하한 폭발 한계 (LEL) [%]: "))
85         vol_percent = float(input("부피 백분율 (Vol%) [%]: "))
86
87         # 입력 값 유효성 검사 및 에러 처리
88         try:
89             print("LFL 계산 결과 (LFL) [%]:", lfl_calculation(lel, vol_percent))
90         except ValueError as e:
91             print(e)

```

```

92
93     elif choice == 5:
94         C = float(input("상수 (C): "))
95         m = float(input("질량 유속 (m) [kg/s]: "))
96         g = float(input("중력 가속도 (g) [m/s^2]: "))
97         d = float(input("배출 구경 (d) [m]: "))
98         p = float(input("압력 (p) [Pa]: "))
99         delta_H = float(input("연소 열 (ΔH) [J/kg]: "))
100
101         try:
102             print("제트 화염 길이 계산 결과 (길이) [m]:", jet_flame_length(C, m, g, d, p, delta_H))
103         except ValueError as e:
104             print(e)
105
106     else:
107         print("잘못된 선택입니다. 프로그램을 종료합니다.")
108
109 > if __name__ == "__main__":
110     main()

```

## 5. 개발 후 느낀 점

이번 과제를 통해 안전관리 컨설팅 기관에서 실제로 사용할 수 있는 실용적인 계산 프로그램을 개발할 수 있었으며, 실제 현장에서 어떤 계산식을 주로 사용할지 고민하는 계기가 되었다. 또한 코드를 개발하면서 정확하고 신뢰할 수 있는 계산 결과를 제공하는 것은 필수적이라는 것을 깨닫게 되었다. 이론적 수업과 다르게 안전관리 컨설팅 기관에서 실제 안전관리자의 입장에서 계산기를 개발할 때 많은 흥미를 느끼고 프로그래밍이 어떤 도움이 되는지 알게 되었다.

## 6. 계산기의 효과

사고 예방 및 위험 관리: 화학물질 유출, 화재, 폭발과 같은 잠재적인 위험 요소를 사전에 예측하고 평가함으로써 사고를 예방할 수 있다.

규정 준수: 산업 및 환경 안전 규정 및 규제 준수를 보다 효율적으로 준수할 수 있다. 계산기를 사용하여 산업 시설의 화학물질 사용, 저장 및 처리에 대한 안전성을 평가하고 적절한 대응을 취할 수 있다.

교육 및 훈련: 안전공학용 계산기는 안전 교육 및 훈련에 활용될 수 있다. 계산기를 사용하여 다양한 시나리오를 시뮬레이션하고 안전한 작업 방법을 학습할 수 있다.