

안전계수 계산기

학번: 2118031

이름: 이웅섭

Github address: https://github.com/2118031Leewoongsub/Safety_factor_calculator

1. 계산기의 목적

안전계수 계산기는 극한강도, 허용응력을 입력하면 자동으로 안전계수의 값이 계산되는 계산기이다.

2. 계산기의 네이밍의 의미

안전계수 계산기의 네이밍의 의미는 말그대로 극한강도와 허용응력의 비를 구하기 위한 계산기의 이름이다.

3. 계산기 개발 계획

`safety_factor`(안전계수)함수가 두 개의 입력변수 `ultimate_strength`(극한강도), `allowable_stress`(허용응력)의 입력값을 받아서 나눈 뒤 출력하는 식이다.

이후 `safety_factor_check` 함수는 사용자로부터 `float`(실수)의 값을 `input` 함수를 통해 입력 받고 앞에 `safety_factor` 함수를 사용해서 안전계수를 계산하고 `if` 조건문을 사용해서 계산된 값이 1 이상이면 “안전계수가 1 이상입니다. 안전합니다.”를 출력하고 1 미만이면 “안전계수가 1 미만입니다. 위험합니다.”로 출력한다. 이후 `choice` 변수를 만들어서 입력된 `Y, N` 의 값에 따라 사용자가 계속 계산기를 사용할 것인지 여부를 결정하게 한다. 마지막으로 `break` 를 입력해서 `while` 반복문을 종료시킨다.
`if __name__ == "__main__":` 코드는 코드블록으로 `safety_factor_check` 함수를 호출하여 프로그램을 실행하는 역할을 한다.

4. 계산기 개발 과정

The screenshot shows a Python IDE with a project named 'Safety_factor_calculator'. The main file, 'Safety_factor_calculator.py', contains the following code:

```
def safety_factor(ultimate_strength, allowable_stress):  
    safety_record = allowable_stress / ultimate_strength if ultimate_strength != 0  
    return safety_record  
  
def safety_factor_check():  
    while True:  
        ultimate_strength = float(input("극한강도를 입력하세요: "))  
        allowable_stress = float(input("허용응력을 입력하세요: "))  
  
        safety = safety_factor(ultimate_strength, allowable_stress)  
  
        print("안전계수:", safety)  
  
        if safety >= 1:  
            print("안전계수가 1 이상입니다. 안전합니다.")  
        else:  
            print("안전계수가 1 미만입니다. 위험합니다.")  
  
        choice = input("이 계산하시겠습니까? (Y/N): ")  
        if choice.upper() != 'Y':  
            break  
  
if __name__ == "__main__":  
    safety_factor_check()
```

The Run console at the bottom shows an error message:

```
File "C:/Users/lempangshu/Documents/GitHub/Safety_factor_calculator/Safety_factor_calculator.py", line 26  
    safety_factor_check()  
    *  
IndentationError: expected an indented block  
  
Process finished with exit code 1
```

첫 번째로 마지막 if `__name__ == "__main__":` `safety_factor_check()` 코드의 들여쓰기 위치를 입력하는 과정에서 함수 `safety_factor_check()`에 포함되는 줄 알고 에러가 발생하였습니다. 이후 함수가 `safety_factor_check()`가 아닌 함수 `safety_factor` 인 것을 확인하고 들여쓰기를 수정하였더니 오류가 해결되었습니다.

The screenshot shows a Python IDE with a project named 'Safety_factor_calculator'. The main file, 'Safety_factor_calculator.py', contains the following code:

```
def safety_factor(ultimate_strength, allowable_stress):
    safety_record = allowable_stress / ultimate_strength if ultimate_strength != 0
    return safety_record

def safety_factor_check():
    while True:
        ultimate_strength = int(input("극한강도를 입력하세요: "))
        allowable_stress = int(input("허용응력을 입력하세요: "))

        safety = safety_factor(ultimate_strength, allowable_stress)

        print("안전계수:", safety)

        if safety >= 1:
            print("안전계수가 1 이상입니다. 안전합니다.")
        else:
            print("안전계수가 1 미만입니다. 위험합니다.")

        choice = input("이 계산하시겠습니까? (Y/N): ")
        if choice.upper() != 'Y':
            break

if __name__ == "__main__":
    safety_factor_check()
```

The bottom panel shows the output of the program. It prompts for '극한강도를 입력하세요:' and '허용응력을 입력하세요:'. The user enters '33.1' for the first value. A traceback error occurs at line 7, column 25, in the 'safety_factor' function, stating: 'ValueError: invalid literal for int() with base 10: '33.1''. The process finished with exit code 1.

두 번째로 Input 함수를 적용하는 과정에서 처음에 int(정수)만 입력해도 문제가 없었지만 소수점이 붙은 계산을 할 때 오류가 발생하여서 float(실수)로 수정하여 오류를 해결하였습니다.

5. 계산기 개발 후기

지금까지 올렸던 github 의 Quiz 들을 모두 다시 보고 실행시켜보면서 이번 과제 코드를 만들었다. 아직까지도 들여쓰기의 위치나 정수, 실수 타입의 작성 등에서 실수하는 모습을 보고 나자신에게 실망했고 기말고사가 매우 걱정되었다. 기말고사 전까지 배웠던 내용들을 다시한번 복습하고 수업시간에 열심히 참여해야겠다는 생각을 하게 되었다. 또한 이러한 코드가 시간을 단축시켜주고 생활을 편리하게 해준다는 생각을 다시한번 하게 되었다.