

계산기 이름

학번: 2318084

이름: 이시현

Github address: <https://github.com/Leesihyeon9/4-.git>

1. 계산기의 목적

- a. 계산기 활용 대상: 근로 손실 일수를 계산해야 할 필요가 있는 사람들 모두

2. 계산기의 네이밍의 의미

- a. 근로손실 계산기는 운동하는 사람들이 제일 무서워하는 근로손실에서 영향을 받았으며 근로 손실 일수를 줄이면 근로손실로 만들 수 있고 근로 손실이 생기면 잃는 것이 생기는 점이 비슷하다고 생각하여 이와 같이 이름을 지었다

3. 계산기 개발 계획

A. 입력 변수는 무엇?

- a. days_lost: 사용자가 손실된 일수를 입력할 때 사용된다
- b. hourly_wage: 사용자가 시간당 임금을 입력할 때 사용된다
- c. productivity_loss_rate: 사용자가 생산성 감소율을 입력할 때 사용된다

B. 개발한 함수는 무엇을 연산하는 함수인 지

- a. total_wage_loss: 하루 일하는 시간을 8 시간으로 가정하여 손실된 일수에 따른 총 임금 손실을 계산합니다.
- b. productivity_loss: 생산성 감소율을 곱하여 생산성에 따른 추가 임금 손실을 계산합니다.
- c. total_loss: 총 임금 손실과 추가 임금 손실을 합하여 총 근로 손실액을 계산합니다.

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

연산과정

- a. total_wage_loss 계산: 사용자가 입력한 손실된 일수를 하루 일하는 시간(여기서는 8 시간으로 가정)과 시간당 임금을 곱하여 하루의 임금

손실을 계산합니다. 즉, 하루 일하는 시간 * 손실된 일수 * 시간당 임금입니다.

b. productivity_loss 계산: 사용자가 입력한 생산성 감소율을 총 임금 손실에 곱하여 추가적인 임금 손실을 계산합니다. 이 값은 총 임금 손실의 일정 비율을 더 잃게 되는 것을 나타냅니다.

c. total_loss 계산: 총 임금 손실과 추가적인 임금 손실을 합하여 전체 근로 손실액을 계산합니다. 이 값은 손실된 일수, 시간당 임금 및 생산성 감소율에 따라 결정됩니다.

설계 이유

이렇게 설계하면 복잡한 과정 없이 빠르게 사용 가능하여 이렇게 만들게 됨

4. 계산기 개발 과정

a. 계획 후 실제 개발 과정을 기록

우선 위와 같이 함수들과 입력 변수들을 어떤 이름으로 작성할지 지정 후에 조건문, 반복문, 사용자 입력 함수를 필수적으로 집어넣어야 했기에 어떤 방식으로 하면 좋을지 생각해보고 근로손실일수에 대한 자료를 찾아보고 어떻게 계산하는게 좋을것인지 구체화 하여 작성하였다.

b. 각 함수는 어떻게 동작하는 지 구체적으로 설명

get_positive_integer 함수:이 함수는 사용자로부터 양의 정수를 입력받도록 요청합니다.사용자가 입력한 값이 숫자가 아니거나 음수일 경우 오류 메시지를 출력하고 다시 입력을 요청합니다.사용자가 올바른 입력을 제공할 때까지 이 과정을 반복하며, 올바른 입력이 제공되면 해당 값을 반환합니다.

get_nonnegative_float 함수:이 함수는 사용자로부터 양의 실수를 입력받도록 요청합니다.사용자가 입력한 값이 숫자가 아니거나 음수일 경우, 또는 입력값이 부동소수점 수가 아닐 경우 오류 메시지를 출력하고 다시 입력을 요청합니다.사용자가 올바른 입력을 제공할 때까지 이 과정을 반복하며, 올바른 입력이 제공되면 해당 값을 반환합니다.

calculate_lost_workdays 함수:이 함수는 사용자로부터 손실된 일수, 시간당 임금, 생산성 감소율을 입력받습니다.손실된 일수, 시간당 임금, 생산성 감소율이 올바른지 확인하기 위해 각각의 입력에 대해 적절한 검증을 수행합니다.손실된 일수, 시간당 임금, 생산성 감소율이 올바르게 입력되면 각각의 손실을

계산하여 총 근로 손실액을 계산합니다.마지막으로 계산된 총 근로 손실액을 출력합니다.

c. 에러 발생 지점

조건문 사용시 클론을 찍지 않아 문제가 발생하였습니다.

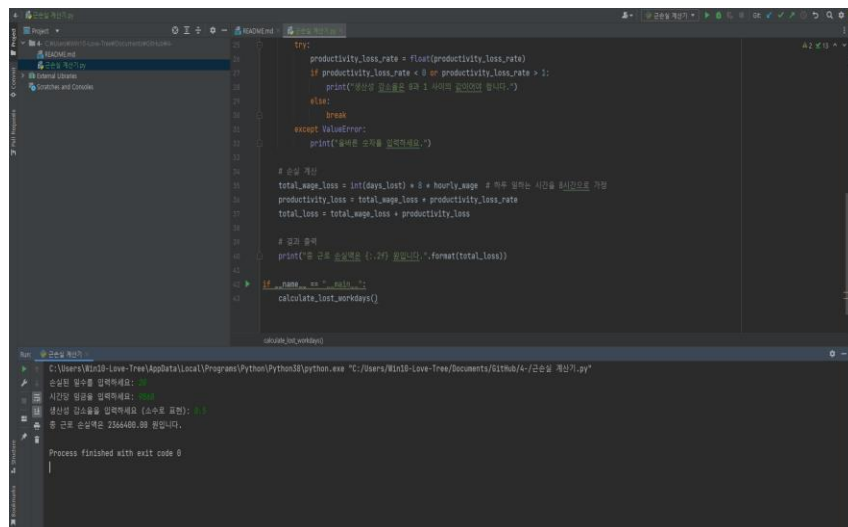
d. 에러 발생에 대한 해결책

조건문 뒤에는 항상 클론을 붙여야 한다는 점을 다시 깨닫고 다시 실행하여봤다.

e. 해결책 적용 시 어떻게 변화

Chat GPT 를 활용하여 잘못 된 부분을 찾고 해결 할 수 있었습니다

f. 동작 결과 캡처



```
def calculate_total_loss():
    productivity_loss_rate = float(productivity_loss_rate)
    if productivity_loss_rate < 0 or productivity_loss_rate > 1:
        print("생산성 손실률은 0과 1.0000 사이여야 합니다.")
        return

    # 손실 계산
    total_wage_loss = (int(days_lost) + 0.5 * hours_wage) * productivity_loss_rate
    productivity_loss = total_wage_loss * productivity_loss_rate
    total_loss = total_wage_loss + productivity_loss

    # 결과 출력
    print(f"총 근로 손실액은 {int(days_lost)} 일과 {0.5 * hours_wage} 시간의 손실로 인해 총 손실액은 {total_loss}입니다.")

if __name__ == "__main__":
    calculate_total_loss()

calculate_total_loss()
```

Run

C:\Users\World-Tree\AppData\Local\Programs\Python\Python38\python.exe "C:/Users/World-Tree/Documents/GitHub/A-근로손실계산기.py"

손실된 일수를 입력하세요: 30
시간당 임금을 입력하세요: 3000
생산성 손실률을 입력하세요 (소수로 입력): 0.1
총 근로 손실액은 3364000.00 원입니다.

Process finished with exit code 0

5. 계산기 개발 후기

a. 계산기 개발 후 느낀 점 설명

계산기를 제작하며 우리가 안전과 관련된 수업을 들었을 때 계산하기에 막막했던 수식들이 내가 만든 계산기를 가지고 숫자 몇가지만 입력하면 값이 나온다는 것이 새로우면서도 신기하다고 느껴졌고 앞으로도 이러한 안전과 관련된 계산기들이 널리 보급되면 힘들게 계산하지 않으며 어떠한 일이나 프로젝트를 진행할 때 조금 더 효율적이고 빠르게 진행될 수 있을 거라고 생각했다.