

만능 안전율 계산기

2018020

박경민

https://github.com/2018020Park/almighty_safety_calculator.git

1. 계산기의 목적

안전율의 공식은 '안전율 = $\frac{\text{기준강도}}{\text{허용응력}}$ ' 이다. 단순히 안전율만 계산하는 것이 아닌 허용응력과 기준강도를 계산하는 계산기도 만들어보고자 이 계산기를 만들게 되었다.

이 계산기는 안전율을 계산하고자 하는 안전관리자라면 유용하게 사용할 수 있을 것이고, 안전관리자를 지망하는 학생들이 안전율 관련 문제를 풀 때, 유용하게 사용할 수 있을 것이다.

2. 계산기의 네이밍 의미

단순히 안전율만 계산하는 것이 아닌 허용응력과 기준강도도 손쉽게 구하는 계산기라는 의미에서 '만능 안전율 계산기'이다.

3. 계산기 개발 계획

class, if, while, int, input 함수들을 사용할 것이다.

class함수 :

안전율 계산기, 허용응력 계산기, 기준강도 계산기를 각자 만들어 class함수로 묶어 둘 것이다.

if함수 :

사용자에게 어떠한 계산기를 사용할 것인지 물어보고 사용자가 답했을 때, 원하는 계산기 블록으로 갈 수 있게 할 것이다.

while함수 :

안전율은 무조건 1보다 커야 한다. 그렇기에 사용자가 입력한 값이 1보다 작으면 허용응력과 기준강도를 다시 입력하게 해 1보다 큰 숫자가 나올 때까지 반복할 것이다.

int함수 :

사용자가 계산을 위해 입력한 값을 정수타입으로 바꾸어 계산할 수 있게 할 것이다.

input함수 :

사용자가 원하는 계산기, 값을 입력할 수 있게 할 것이다.

4. 계산기 개발

4.1 계산기 개발 과정

<p>안전율 계산기라는 의미에서</p> <p>Safety_Calculator라고 작명했다. x와 y는</p> <p>계산하기 위한 변수를 입력받는 장소로 설정</p> <p>했다.</p>	<pre>class Safety_Calculator: def __init__(self, x, y): self.x = x self.y = y</pre>
<p>첫 번째로 안전율 계산기이다. if함수를 넣어</p> <p>계산한 값이 1보다 크면 결과값이 바로 출력</p> <p>되도록 했고, 1보다 작다면 일단 결과값을</p> <p>출력한 다음 계산을 다시 하게 했다. 또한</p> <p>while반복문으로 1보다 큰 수가 나올 때까지</p> <p>반복하게 만들었다.</p>	<pre>def safety(self): z = self.x / self.y if z > 1: print(f"안전율이 {z}이므로 1보다 높습니다. 허용 가능합니다.") elif z < 1: print(f"안전율이 {z}이므로 1보다 낮습니다. 허용 할 수 없습니다. 재입력하십시오.") while True: a = int(input("기준강도를 입력하십시오. ")) b = int(input("허용응력을 입력하십시오. ")) c = a / b if c < 1: print(f"안전율이 {c}이므로 1보다 낮습니다. 허용 할 수 없습니다. 재입력 하십시오.") elif c > 1: print(f"안전율이 {c}이므로 1보다 높습니다. 허용 가능합니다.") break</pre>
<p>허용응력 계산기이다. 응력이 영어로 stress</p> <p>이기에 stress로 작명했다. 입력받은 값을</p> <p>바로 출력할 수 있게 했다.</p>	<pre>def stress(self): print(f"허용응력은 {self.y / self.x}입니다.")</pre>
<p>기준강도 계산기이다. 강도가 영어로</p> <p>strength이기에 strength로 작명했다. 입력</p> <p>받은 값을 바로 출력할 수 있게 했다.</p>	<pre>def strength(self): print(f"기준강도는 {self.x * self.y}입니다.")</pre>
<p>input을 써서 사용자가 원하는 계산기를 입</p> <p>력할 수 있게 했다.</p>	<pre>n = input("계산기 입력. ('안전율 계산기', '허용응력 계산기', '기준강도 계산기' 중 택1)")</pre>

<p>사용자가 입력한 계산기에 맞추어서 class로 묶어 둔 계산기로 보내는 과정이다. or을 사용한 이유는, 예를 들어 사용자가 '안전율 계산기'에 띄어쓰기를 하지 않을 경우, 이를 고려하여 or을 써서 띄어쓰기를 했을 경우와 하지 않았을 경우 둘 다 대비했다.</p> <p>if를 사용하여 안전율 계산기, 허용응력 계산기, 기준강도 계산기에 맞추어서 class를 호출할 수 있도록 했다.</p>	<pre> if m == "안전율 계산기" or m == "안전율계산기": k = int(input("기준강도 입력")) l = int(input("허용응력 입력")) safety_factor = Safety_Calculator(k, l) safety_factor.safety() elif m == "허용응력 계산기" or m == "허용응력계산기": o = int(input("안전율 입력")) p = int(input("기준강도 입력")) allowable_stress = Safety_Calculator(o, p) allowable_stress.stress() elif m == "기준강도 계산기" or m == "기준강도계산기": u = int(input("안전율 입력")) e = int(input("허용응력 입력")) basic_strength = Safety_Calculator(u, e) basic_strength.strength() </pre>
--	---

4.2 에러 발생 지점

<pre> 계산기 입력. ('안전율 계산기', '허용응력 계산기', '기준강도 계산기' 중 택1) >> 허용응력 계산기 기준강도 입력 >> 50 허용응력 입력 >> 10 안전율이 5.0이므로 1보다 높습니다. 허용 가능합니다. </pre> <p>사진과 같이 계산기 입력란에 '허용응력 계산기'라고 입력했음에도 불구하고 '안전율 계산기'가 호출되었다.</p>

4.3 에러 발생에 대한 해결책

```
if m == "안전율 계산기" or "안전율계산기":  
    k = int(input("기준강도 입력"))  
    l = int(input("허용응력 입력"))  
    safety_factor = Safety_Calculator(k, l)  
    safety_factor.safety()
```

에러지점을 검토해 본 결과 or을 사용하고 or 뒤에 'm =='을 붙이지 않아 에러가 발생했다는 것을 알 수 있었다.

```
if m == "안전율 계산기" or m == "안전율계산기":  
    k = int(input("기준강도 입력"))  
    l = int(input("허용응력 입력"))  
    safety_factor = Safety_Calculator(k, l)  
    safety_factor.safety()
```

해결책을 적용한 후엔 제대로 작동하는 것을 확인했다.

4.4 동작 결과

```
계산기 입력. ('안전율 계산기', '허용응력 계산기', '기준강도 계산기' 중 택1) >? 안전율계산기
기준강도 입력 >? 50
허용응력 입력 >? 10
안전율이 5.0이므로 1보다 높습니다. 허용 가능합니다.
```

1. 안전율 계산기 사용 (1)

안전율과 계산기 사이에 띄어쓰기를 하지 않아도 잘 작동했다.

계산 결과가 1보다 높았을 때의 결과이다.

```
계산기 입력. ('안전율 계산기', '허용응력 계산기', '기준강도 계산기' 중 택1) >? 안전율 계산기
기준강도 입력 >? 10
허용응력 입력 >? 50
안전율이 0.2이므로 1보다 낮습니다. 허용 할 수 없습니다. 재입력 하십시오.
기준강도를 입력하십시오. >? 20
허용응력을 입력하십시오. >? 50
안전율이 0.4이므로 1보다 낮습니다. 허용 할 수 없습니다. 재입력 하십시오.
기준강도를 입력하십시오. >? 50
허용응력을 입력하십시오. >? 10
안전율이 5.0이므로 1보다 높습니다. 허용 가능합니다.
```

2. 안전율 계산기 사용 (2)

계산 결과가 1보다 작았을 때, 계산기가 다시 작동하는 것을 확인했다.

결국엔 계산 결과가 1보다 커졌을 때, break로 인해 작동을 멈추는 것을 확인했다.

```
계산기 입력. ('안전율 계산기', '허용응력 계산기', '기준강도 계산기' 중 택1) >? 허용응력 계산기
안전율 입력 >? 5
기준강도 입력 >? 10
허용응력은 2.0입니다.
```

3. 허용응력 계산기 사용

10 / 5 가 잘 작동하는 것을 확인했다.

<div> <div>계산기 입력. ('안전율 계산기', '허용응력 계산기', '기준강도 계산기' 중 택1)</div> <div> <div>기준강도 계산기</div> <div> <div>안전율 입력</div> <div>2</div> </div> <div> <div>허용응력 입력</div> <div>5</div> </div> <div>기준강도는 10입니다.</div> </div> </div>
<div>4. 기준강도 계산기 사용</div> <div>2 * 5 가 잘 작동하는 것을 확인했다.</div>

5. 계산기 개발 후기

일단 가장 먼저 든 생각을 말해보자면, '머리에 쥐가 난다'는 표현을 사용할 수 있을 것 같다. 나 자신이 사용할 수 있는 함수들이 머릿속에서 뒤죽박죽이 되어 떠다니는데, 머리에 쥐가 나질 않고는 버틸 수 없었다. 머릿속에 있는 함수를 정리하는 데만 오랜 시간이 걸렸고, 또 그것을 활용하기까지 오랜 시간이 걸렸다.

컴퓨터프로그래밍 실습II 중간고사를 본 경험이 큰 도움이 됐다. 중간고사 맨 마지막 문제를 해결해서 프로그래밍에 자신감을 얻을 수 있었고, 계산기를 완성할 수 있었다. 또한 이 계산기를 만들면서 프로그래밍 실력에 큰 발전이 있었다는 것을 체감할 수 있었다.

개발 도중에 챗gpt의 도움을 받을까 수백번 고민이 되었지만, 나 자신에게 오기가 생겨 결국엔 성공할 수 있었다. 만들고 나니 참 별것 없는 코딩이 짜였는데, 이 간단한 것을 생각하는 게 왜 이렇게 어려웠는지 모르겠다. 전 세계 모든 프로그래머분들에게 존경심이 들며 내가 갈 길은 아니라는 생각이 들었다.

전 세계 모든 프로그래머분들 파이팅입니다..!